OneDate

프로젝트

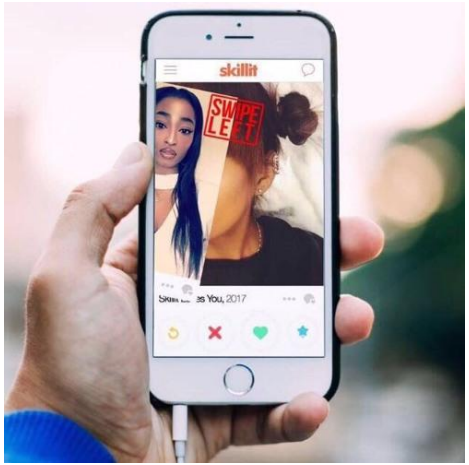
개요

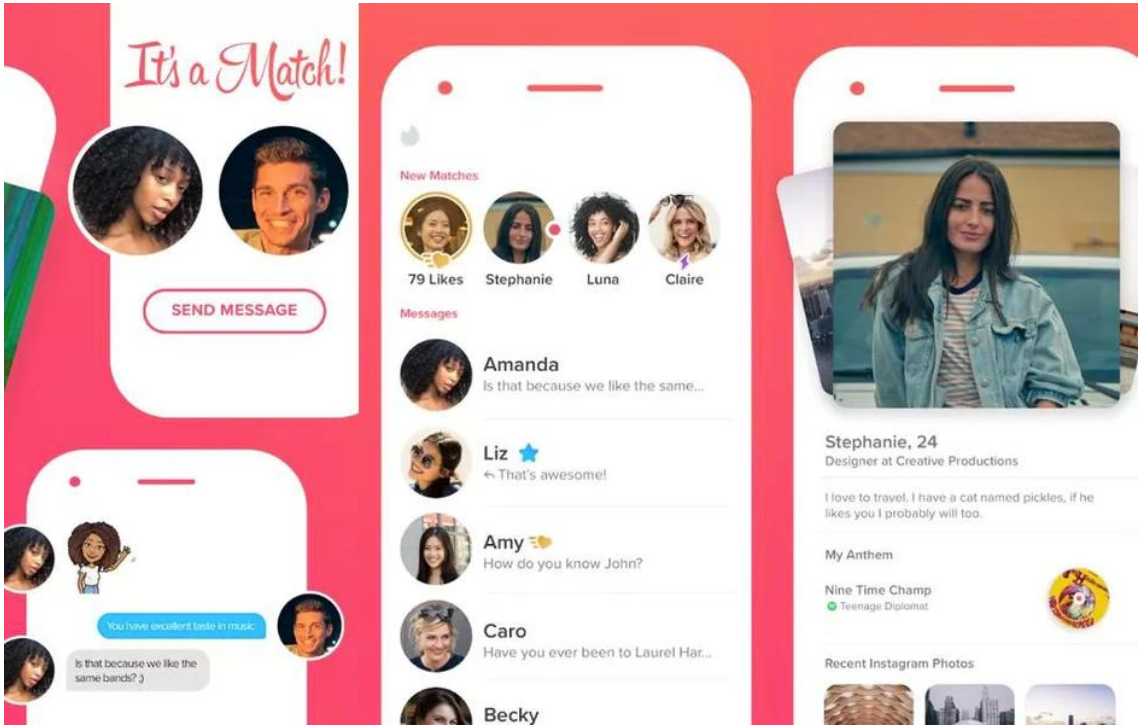
1. 개발 목적 / 개발 일정 / 개발 환경
2. DB 구조 / 페이지 흐름도 / 설계 주안점
3. 사용기술
4. 기술 상세

데이트 어플리케이션

“소셜 디스커버리”

새로운 이성 또는 친구를 매칭





개발 목적

- 데이트 어플리케이션 자체 구현
- 이성 뿐만 아닌 '새로운 친구'를 사귈 수 있는 매개체 역할

개발 일정

-1주차

DB 설계 회의

데이터 베이스 구현

-2주차

서비스 구현

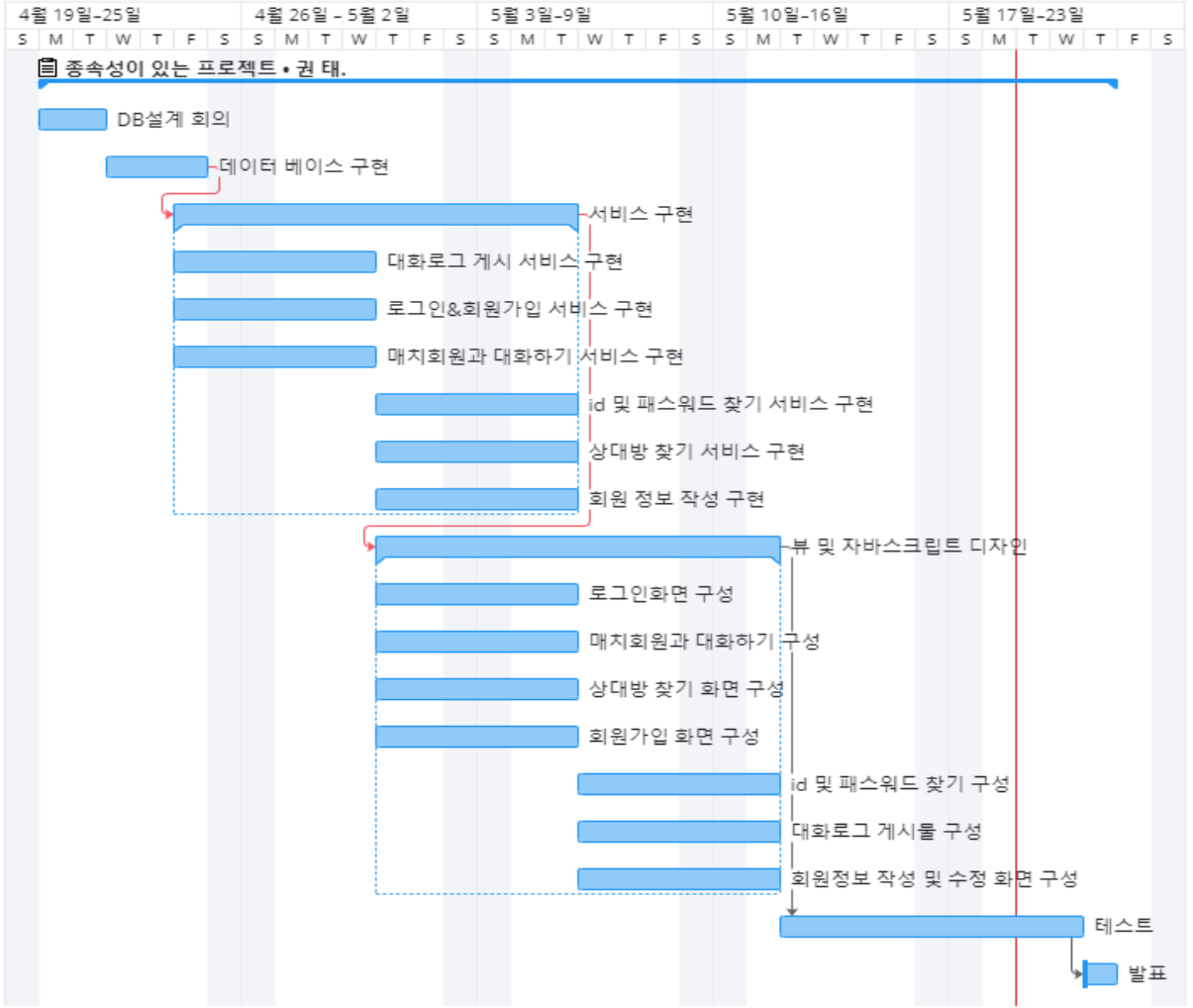
뷰 구현

-3주차

자바 스크립트 / css 구현

-4주차

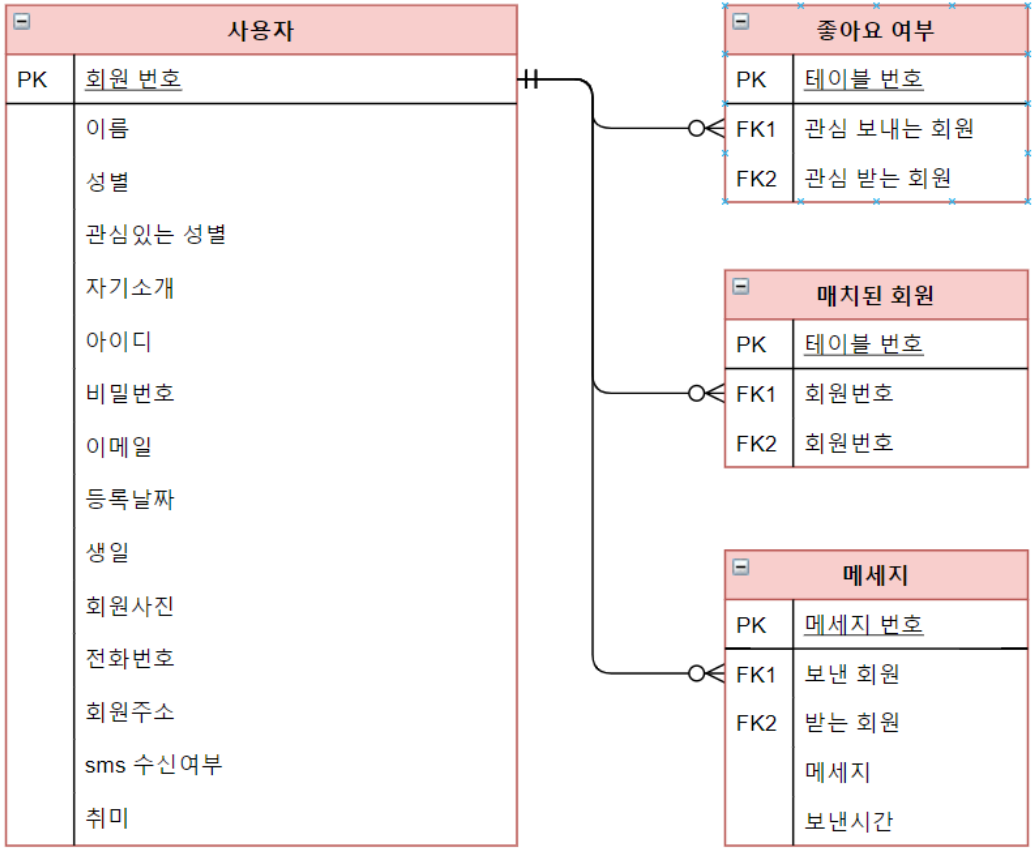
테스트



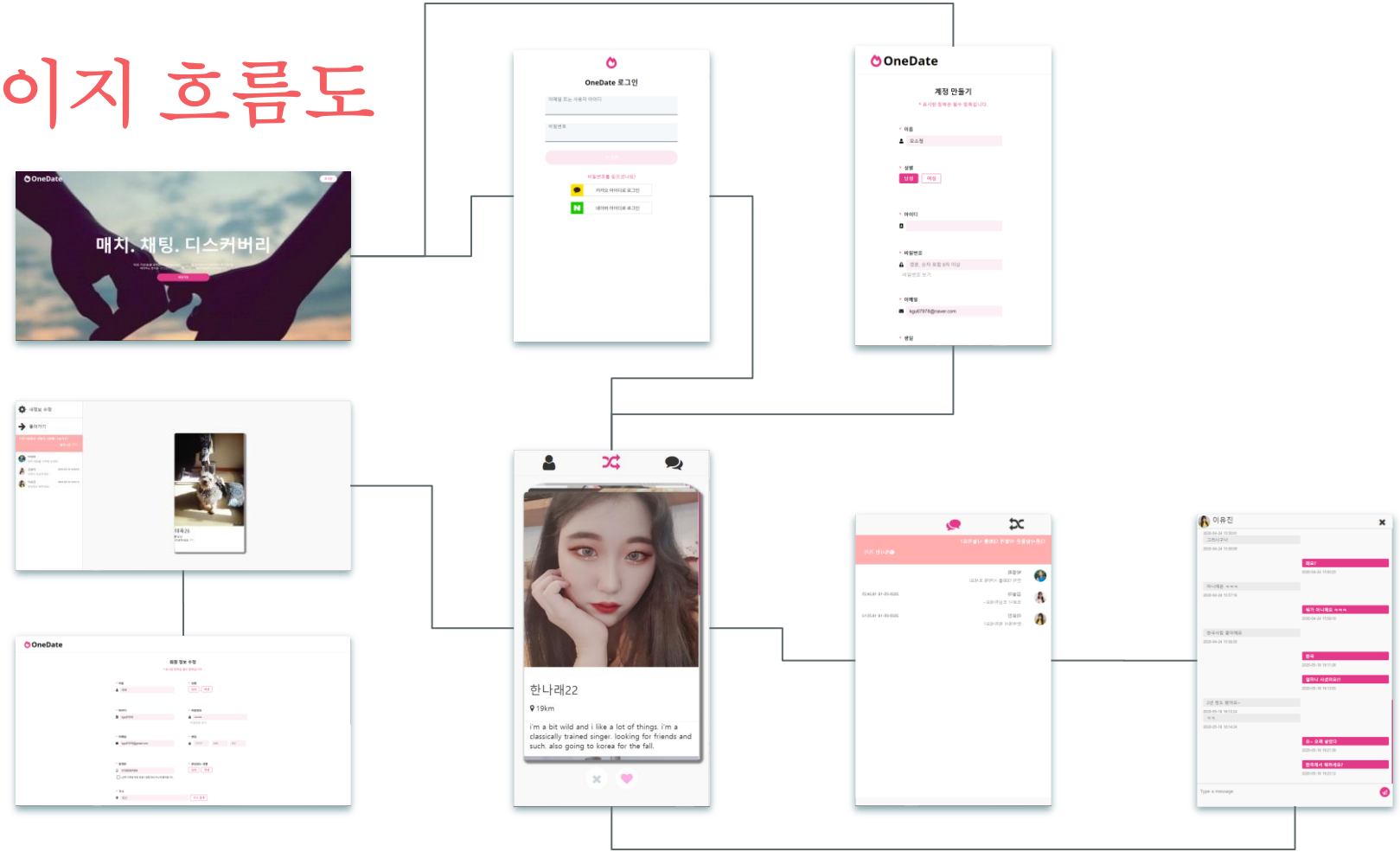
개발 환경

- 언어
  - java
  - Javascript
- 데이터 베이스
  - Mysql
- IDE
  - spring tool
- 빌드도구
  - maven

DB 구조


사용자
PK
회원 번호
이름
성별
관심있는 성별
자기소개
아이디
비밀번호
이메일
등록날짜
생일
회원사진
전화번호
회원주소
sms 수신여부
취미
좋아요 여부
PK
테이블 번호
FK1
관심 보내는 회원
FK2
관심 받는 회원
매치된 회원
PK
테이블 번호
FK1
회원번호
FK2
회원번호
메세지
PK
메세지 번호
FK1
보낸 회원
FK2
받는 회원
메세지
보낸시간

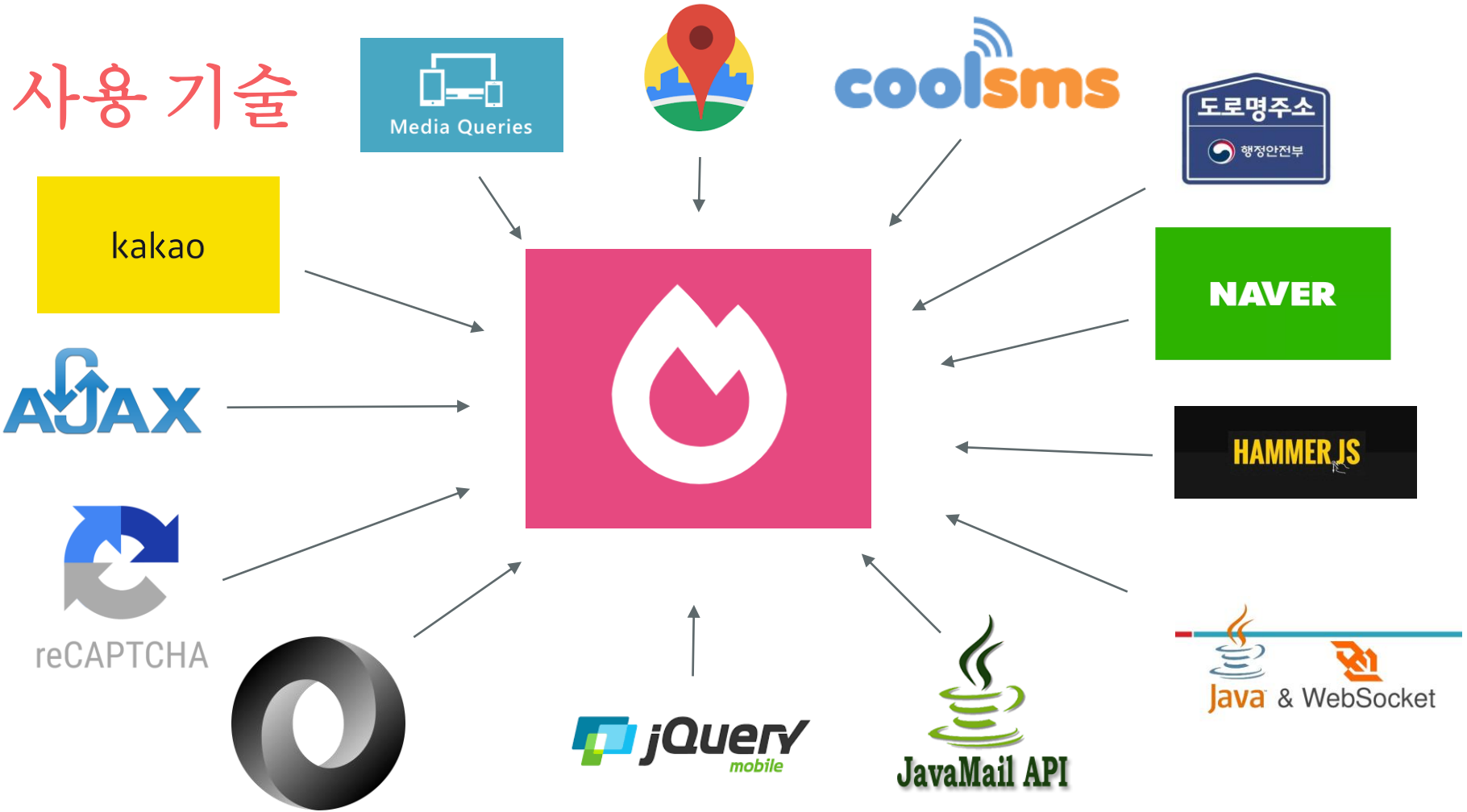
페이지 흐름도


OneDate
매치. 채팅. 디스커버리
OneDate 로그인
OneDate
계정 만들기
OneDate
한나래22
19km
I'm a bit wild and i like a lot of things. i'm a classically trained singer. looking for friends and such. also going to korea for the fall.
이유진

설계 주안점

- WebSocket과 표준 데이터 포멧인 json을 사용하여 실시간 양방향 채팅 구현
- 구글 GeoCoder를 이용하여 사용자간의 주소 거리를 계산
- ajax 비동기 통신을 활용하여 페이지 이동을 최소화
- 카카오로그인 / 네이버로그인 api 활용, 회원가입 및 로그인 편의성 제공
- 이메일 인증 및 구글 리캡챠 활용, 인증 절차 강화

사용 기술


Media Queries
coolsms
도로명주소
행정안전부
kakao
NAVER
AJAX
HAMMER.JS
reCAPTCHA
jQuery mobile
JavaMail API
Java & WebSocket

자바 mail API

회원 가입시 사용자 인증을 위해 메일 발송

네이버 smtp 프로토콜 이용

ajax 비동기 통신으로 페이지 이동없이 인증코드를 발송

자바 mail API

```
@ResponseBody
@RequestMapping(value="/sendCode", method=RequestMethod.POST)
public String sendCode(@RequestParam("email")String email, Model model) throws Exception {

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

    String key = "";
    int n = (int)(Math.random() * 1000000);
    if(n < 100000) {
        n += 100000;
    }
    key = String.valueOf(n);

    MailHandler sendMail = new MailHandler(mailSender);
    sendMail.setSubject("[OneDate 인증 코드입니다]");
    sendMail.setText(new StringBuffer().append("<h1>이메일 주소 인증하기</h1>")
            .append("<br><br>OneDate를 시작하려면 다음 인증 코드를 입력하세요.<br><br>")
            .append("<h1>" + key + "</h1>")
            .append("<br>감사합니다.")
            .append("<br>OneDate").toString());
    sendMail.setFrom("noraboca@naver.com", "OneDate");
    sendMail.setTo(email);
    sendMail.send();

    session.setAttribute("key", key);

    return key;
}
```

자바 mail API

```java
public class MailHandler {

    private JavaMailSender mailSender;
    private MimeMessage msg;
    private MimeMessageHelper msgHelper;

    public MailHandler(JavaMailSender mailSender) throws MessagingException {
        this.mailSender = mailSender;
        msg = this.mailSender.createMimeMessage();
        msgHelper = new MimeMessageHelper(msg, true, "UTF-8");
    }

    // Email Title
    public void setSubject(String subject) throws MessagingException {
        msgHelper.setSubject(subject);
    }

    // Email Content(Text)
    public void setText(String htmlContent) throws MessagingException {
        msgHelper.setText(htmlContent, true);
    }

    // Sender Info
    public void setFrom(String email, String name) throws UnsupportedEncodingException, MessagingException {
        msgHelper.setFrom(email, name);
    }

    // Recipient Info
    public void setTo(String email) throws MessagingException {
```

카카오 로그인 API

카카오톡에서 제공하는 openAPI 활용

회원가입 및 로그인 시 닉네임, 이메일, 성별, 생일 등

카카오톡의 사용자 프로필 정보를 가져옴

카카오 로그인 API

```
private final static String REDIRECT_URI = "http://localhost:8181/project/kakaologin";

public String getAuthorizationUrl(HttpSession session) {
    String kakaoUrl = "https://kauth.kakao.com/oauth/authorize?client_id=" + CLIENT_ID + "&redirect_uri="
            + REDIRECT_URI + "&response_type=code";
    return kakaoUrl;
}
```

kakao

카카오 로그인 API

```
public JsonNode getAccessToken(String authorize_code) {
    final String RequestUrl = "https://kauth.kakao.com/oauth/token";
    final List<NameValuePair> postParams = new ArrayList<NameValuePair>();
    postParams.add(new BasicNameValuePair("grant_type", "authorization_code"));
    postParams.add(new BasicNameValuePair("client_id", CLIENT_ID)); // REST API
    postParams.add(new BasicNameValuePair("redirect_uri", REDIRECT_URI)); //
    postParams.add(new BasicNameValuePair("code", authorize_code)); //
    final HttpClient client = HttpClientBuilder.create().build();
    final HttpPost post = new HttpPost(RequestUrl);
    JsonNode returnNode = null;
    try {
        post.setEntity(new UrlEncodedFormEntity(postParams));
        final HttpResponse response = client.execute(post);
        final int responseCode = response.getStatusLine().getStatusCode();

        System.out.println("\nSending 'POST' request to URL : " + RequestUrl);
        System.out.println("Post parameters : " + postParams);
        System.out.println("Response Code : " + responseCode);

        // JSON
        ObjectMapper mapper = new ObjectMapper();
        returnNode = mapper.readTree(response.getEntity().getContent());
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        // clear resources
    }
    return returnNode;
}
```

kakao

카카오 로그인 API

```
public JsonNode getKakaoUserInfo(JsonNode accessToken) {
    final String RequestUrl = "https://kapi.kakao.com/v2/user/me";
    final HttpClient client = HttpClientBuilder.create().build();
    final HttpPost post = new HttpPost(RequestUrl);
    // add header
    post.addHeader("Authorization", "Bearer " + accessToken);
    JsonNode returnNode = null;
    try {
        final HttpResponse response = client.execute(post);
        final int responseCode = response.getStatusLine().getStatusCode();

        System.out.println("\nSending 'POST' request to URL : " + RequestUrl);
        System.out.println("Response Code : " + responseCode);

        // JSON □□□□ □□ó□□ ó□□
        ObjectMapper mapper = new ObjectMapper();
        returnNode = mapper.readTree(response.getEntity().getContent());
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        // clear resources
    }
    return returnNode;
}
```

kakao

네이버 로그인 API

네이버에서 제공하는 openAPI 활용

회원가입 및 로그인 시 닉네임, 이메일, 성별, 생일 등

네이버의 사용자 프로필 정보를 가져옴

네이버 로그인 API

```
public String getAuthorizationUrl(HttpSession session) {
    // 세션 유효성 검증을 위하여 java.util.UUID를 사용하여 난수를 생성
    String state = generateRandomString();
    // 세션에 저장
    setSession(session, state);
    //  Scribe에서 제공하는 인증 URL  생성 기능을 이용하여 인증 URL  생성하여 리턴
    OAuth20Service oauthService = new ServiceBuilder()
        .apiKey(CLIENT_ID)
        .apiSecret(CLIENT_SECRET)
        .callback(REDIRECT_URI)
        .state(state)
        .build(NaverLoginApi.instance());
    return oauthService.getAuthorizationUrl();
}
```

NAVER

네이버 로그인 API

```
// getAccessToken()메서드를 통해 로그인 시 Callback으로 전달
public OAuth2AccessToken getAccessToken(HttpSession session,
        String code, String state) throws IOException {
    // Callback
    String sessionState = getSession(session);
    if (StringUtils.pathEquals(sessionState, state)) {
        OAuth20Service oauthService = new ServiceBuilder()
                .apiKey(CLIENT_ID)
                .apiSecret(CLIENT_SECRET)
                .callback(REDIRECT_URI)
                .state(state)
                .build(NaverLoginApi.instance());
        // Scribe
        OAuth2AccessToken accessToken = oauthService.getAccessToken(code);
        return accessToken;
    }
    return null;
}
```

NAVER

네이버 로그인 API

```
public String getUserProfile(OAuth2AccessToken oauthToken) throws IOException {
    OAuth20Service oauthService = new ServiceBuilder()
            .apiKey(CLIENT_ID)
            .apiSecret(CLIENT_SECRET)
            .callback(REDIRECT_URI)
            .build(NaverLoginApi.instance());
    OAuthRequest request = new OAuthRequest(Verb.GET, PROFILE_API_URL, oauthService)
    oauthService.signRequest(oauthToken, request);
    Response response = request.send();
    return response.getBody();
}
```

NAVER

hammer.js

모바일 웹 터치 제스쳐

- PointerEvent, Touch, Mouse 이벤트들로 부터 제스쳐를 식별(recognize)하는 오픈소스 라이브러리
- ajax와 연동, 매칭시 이벤트 발생
- SDK를 다운받아서 사용

```
<script type="text/javascript" src="resources/js/hammer.js"></script>
```

hammer.js

```
'use strict';

var tinderContainer = document.querySelector('.tinder');
var allCards = document.querySelectorAll('.tinder--card');
var nope = document.getElementById('nope');
var love = document.getElementById('love');
function getUrlParams() {
    var params = {};
    window.location.search.replace(/[?&]+([^=&]+)=([^&]*)/gi, function(str, key, value) { params[key] = value; });
    return params;
}

function initCards(card, index) {
    var newCards = document.querySelectorAll('.tinder--card:not(.removed)');

    newCards.forEach(function (card, index) {
        card.style.zIndex = allCards.length - index;
        card.style.transform = 'scale(' + (20 - index) / 20 + ') translateY(-' + 30 * index + 'px)';
        card.style.opacity = (10 - index) / 10;
    });

    tinderContainer.classList.add('loaded');

}

initCards();
```

hammer.js


```
allCards.forEach(function (el) {

    var hammertime = new Hammer(el);

    hammertime.on('pan', function (event) {
        el.classList.add('moving');
    });

    hammertime.on('pan', function (event) {
        if (event.deltaX === 0) return;
        if (event.center.x === 0 && event.center.y === 0) return;

        tinderContainer.classList.toggle('tinder_love', event.deltaX > 0);
        tinderContainer.classList.toggle('tinder_nope', event.deltaX < 0);

        var xMulti = event.deltaX * 0.03;
        var yMulti = event.deltaY / 80;
        var rotate = xMulti * yMulti;

        event.target.style.transform = 'translate(' + event.deltaX + 'px, ' + event.deltaY + 'px) rotate(' + rotate + 'deg)';
    });
```

hammer.js

```
hammertime.on('panend', function (event) {

    var cards = document.querySelectorAll('.tinder--card:not(.removed)');
    var card = cards[0];

    el.classList.remove('moving');
    tinderContainer.classList.remove('tinder_love');
    tinderContainer.classList.remove('tinder_nope');

    var moveOutWidth = document.body.clientWidth;
    var keep = Math.abs(event.deltaX) < 80 || Math.abs(event.velocityX) < 0.5;

    event.target.classList.toggle('removed', !keep);

    if (keep) {
        event.target.style.transform = '';

    } else {
```

hammer.js

```
if (keep) {
    event.target.style.transform = '';

} else {
    var endX = Math.max(Math.abs(event.velocityX) * moveOutWidth, moveOutWidth);
    var toX = event.deltaX > 0 ? endX : -endX;
    var endY = Math.abs(event.velocityY) * moveOutWidth;
    var toY = event.deltaY > 0 ? endY : -endY;
    var xMulti = event.deltaX * 0.03;
    var yMulti = event.deltaY / 80;
    var rotate = xMulti * yMulti;
    event.target.style.transform = 'translate(' + toX + 'px, ' + (toY + event.deltaY) + 'px) rotate(' + rotate + 'deg)';

    if(toX>0){

        var oParams = getUrlParams();
        //서버에 보냄
        $.ajax({
            url:"main",
            dataType:"json",
            data: {user : oParams.userId,
                interested_idx : card.id
            },
            success:function(data){
                if(data.ismatched){
                    Swal.fire({
                        title: '축하드립니다!',
                        text: data.newMatch+'님 과 매치되셨습니다',
                        imageUrl: 'resources/client_img/'+data.img_src,
                        imageHeight: 400,
                        imageAlt: 'Custom image',
                    }).then((result) => {
                        location.reload();
                    })

                }
            }
        });

    }
    initCards();
}
```

구글 GeoCoding API

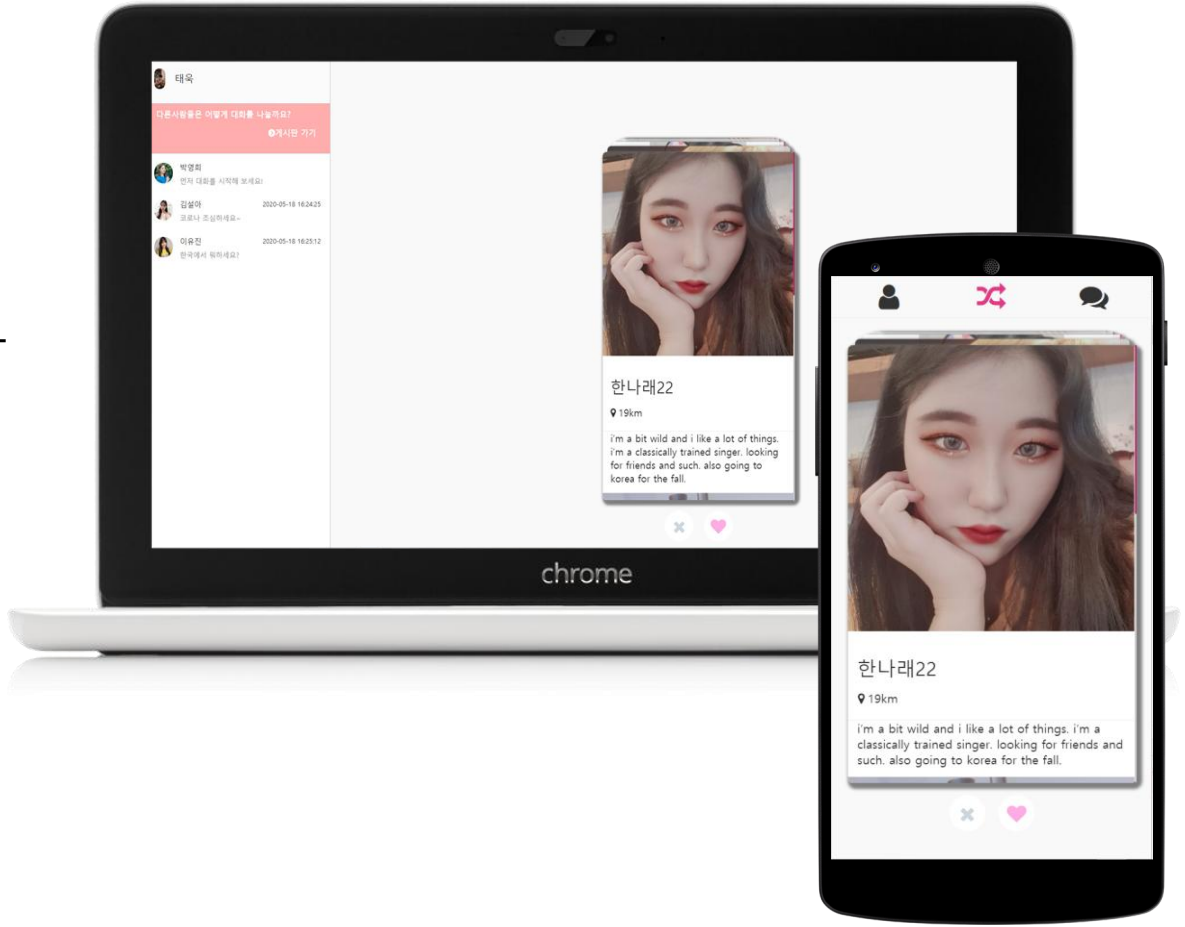
- 고유명칭(주소나 산, 호수의 이름 등)을 가지고 위도와 경도의 좌표값을 리턴
- 사용자의 도로명 주소와 각 회원들 주소간 거리 계산

반응형 웹페이지

미디어 쿼리

JQuery mobile 사용

모바일 UI 최적화



THANK YOU!