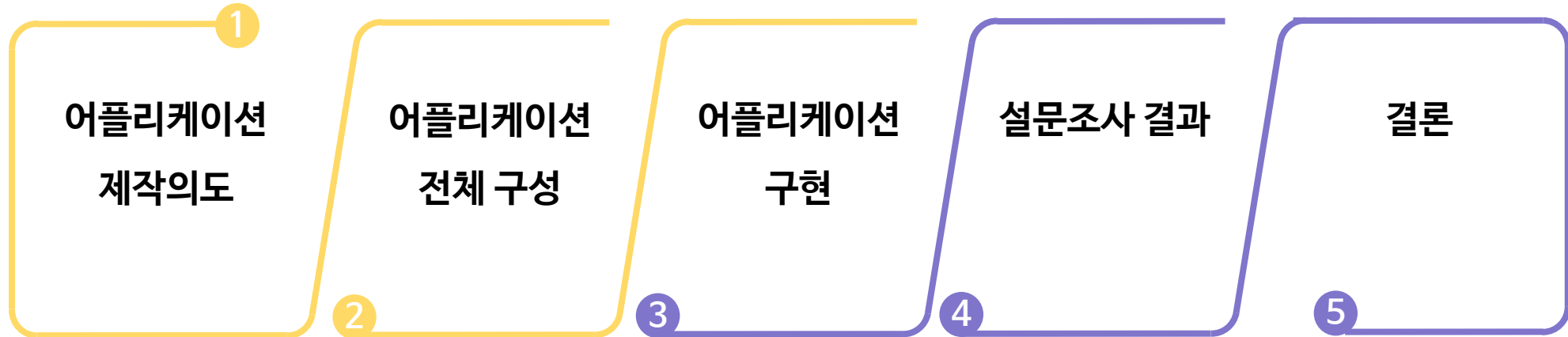


**다양한 여행지의
필요한 정보들을 담은
여행기 앱 개발**

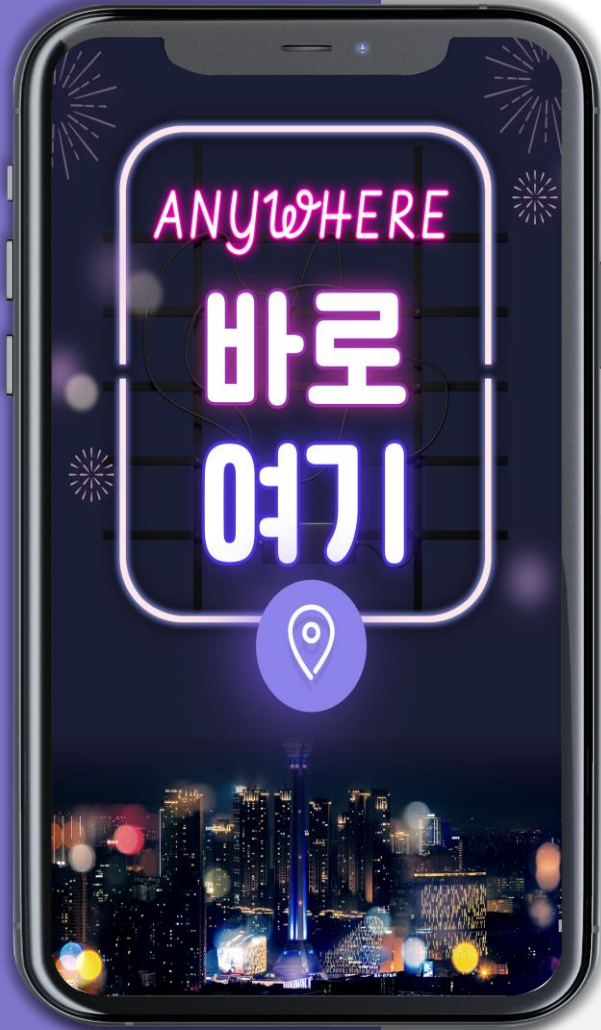
여행지 어플리케이션 소개



어플리케이션 구현 및 결과

Project 1

어플 제작 의도



어플리케이션 제작의도

흔히 여행을 다니다 보면, 다양한 정보가 필요하고
그 정보를 수집 하기 위해 다양한 플랫폼들을 이용하고는 한다.

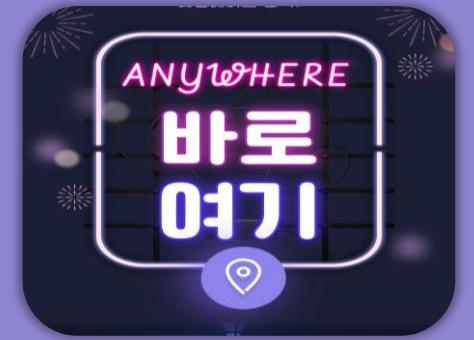
하지만 그에 대한 정보를 찾기에는 많은 시간이 필요하고,
일일이 다 찾아보아야 하는 점이 불편하게 느껴진다.
또한 여행지에 대한 흡연장, 주차장, 화장실에 대한 정보를 찾기 쉽지 않다.

이러한 정보들을 한번에 모두 담은 앱이 있다면
보편적으로 편리할 것이라고 생각하여 앱을 제작하게 되었다.

Project 1

어플 제작 의도

선행 연구 관련 플랫폼 및 비교



데이트를 유용하게 할 수 있는 데이트 플랫폼

- ✓ 특정 지역에 해당하는 정보들만 가지고 있음
- ✓ 다양한 지역에 대한 정보가 부족함.
- ✓ 광고성에 많이 의존하여 정확도가 조금 떨어짐

지역에 대한 숙박, 레저, 액티비티, 항공 등 구매 서비스 및 여행 종합 정보를 담은 플랫폼

- ✓ 티켓과 숙박 업소를 추천하는 앱
- ✓ 여행에 대한 정보는 부족
- ✓ 따로 지역에 대한 추천은 없음

지역에 대한 숙박, 레저, 액티비티, 항공 등 구매 서비스 및 여행 종합 정보를 담은 플랫폼

- ✓ 티켓과 숙박 업소를 추천하는 앱
- ✓ 여행에 대한 정보는 부족
- ✓ 따로 지역에 대한 추천은 없음

여행지의 모든 정보를 담은 플랫폼

- ✓ 사용자에게 여행지 핫플레이스를 추천
- ✓ 여행지의 맛집, 카페 추천
- ✓ 내 위치 기준 흡연장 찾기
- ✓ 내 위치 기준 화장실 찾기
- ✓ 내 위치 기준 주차장 찾기
- ✓ 일정 스케줄러

Project 2

어플 구성



어플리케이션 구성

- ✓ 핫플레이스
- ✓ 맛집 , 카페
- ✓ 주차장
- ✓ 화장실
- ✓ 흡연장 찾기
- ✓ 일정메모

Step.1

```

Button b = (Button)findViewById(R.id.Button1);
Button c = (Button)findViewById(R.id.Button2);
Button d = (Button)findViewById(R.id.Button3);
Button e = (Button)findViewById(R.id.Button4);
Button f = (Button)findViewById(R.id.Button5);
Button g = (Button)findViewById(R.id.Button6);
Button h = (Button)findViewById(R.id.Button7);
Button i = (Button)findViewById(R.id.Button8);
Button j = (Button)findViewById(R.id.Button9);
Button k = (Button)findViewById(R.id.Button10);
Button l = (Button)findViewById(R.id.Button11);
Button m = (Button)findViewById(R.id.Button12);
Button n = (Button)findViewById(R.id.Button13);
Button o = (Button)findViewById(R.id.Button14);
Button p = (Button)findViewById(R.id.Button15);
Button q = (Button)findViewById(R.id.Button16);
Button r = (Button)findViewById(R.id.Button17);
Button s = (Button)findViewById(R.id.Button18);
Button t = (Button)findViewById(R.id.Button19);
Button u = (Button)findViewById(R.id.Button20);
b.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent( packageContext: f_seoul_north.this,
            f_seoul_north_eunpyung.class);
        startActivity(intent);
    }
});
c.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent( packageContext: f_seoul_north.this,
            f_seoul_north_eunpyung.class);
    }
});

```

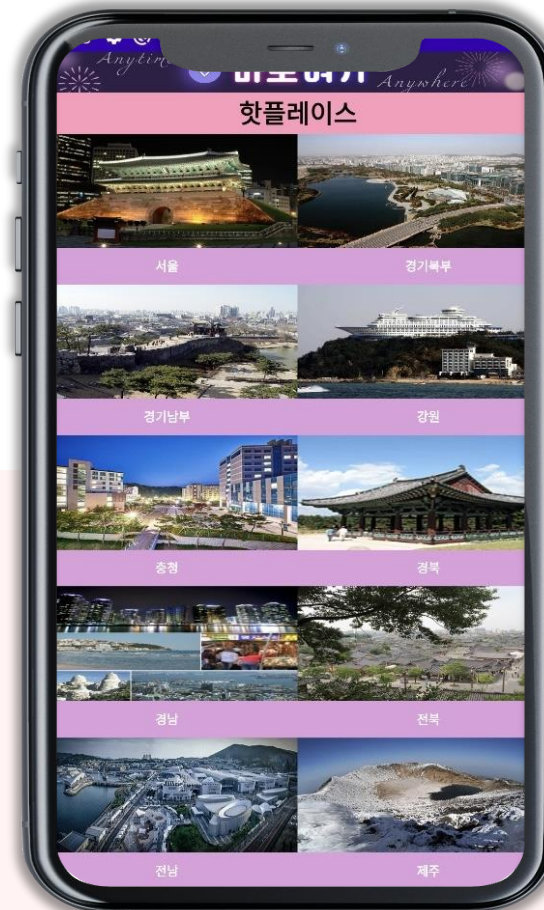
Step.2

```

c h_seoul_north_etaewon
c h_seoul_north_eunpyung
c h_seoul_north_gundae
c h_seoul_north_hongdae
c h_seoul_north_insadong
c h_seoul_north_jongro
c h_seoul_north_joongkoo
c h_seoul_north_kangbook
c h_seoul_north_kwanghwamoon
c h_seoul_north_kwangjingoo
c h_seoul_north_mapo
c h_seoul_north_myungdong
c h_seoul_north_nowon
c h_seoul_north_sangsoo
c h_seoul_north_sinchon
c h_seoul_north_sungbook
c h_seoul_north_yongsan
c h_seoul_north_yunnam
c h_seoul_south
c h_seoul_south_dongzak
c h_seoul_south_kangdong
c h_seoul_south_kangnam
c h_seoul_south_kangseo
c h_seoul_south_karsoogil
c h_seoul_south_kumcheon
c h_seoul_south_kwanak
c h_seoul_south_seocho
c h_seoul_south_songpa
c h_seoul_south_yungdungpo
c hotplace
c main

```

Step.3



Step.4



핫플레이스와, 맛집 카페는 레이아웃이 동일하게 이루어져 있으며, 데이터들은 향후 개발을 위해 각각 생성됨.

Step.1

```
<!-- 카카오 지도 API -->
<meta-data
    android:name="com.kakao.sdk.AppKey"
    android:value="f8cbc7d437b5949e3569c8f8b489c5af" />
```

Step.3

```
String apiUrl = "http://api.odcloud.kr/api/15069274/v1/uddi:6e9ed9bd-b5ad-4188-9bbe-f9516113ddf4?page=1&perPage=196";
String apiKey = "u6QSft9MEaI605p0q6rPPmV01bh%2F9j2J8uFurG3h5%2BPU1pNA%2FKmp35jNC2Ne1d0N19z6c6Zu%2B42HrMf28tknqg%3D%3D";

public ArrayList<Data> getData() {
    //return과 관련된 부분
    ArrayList<Data> dataArr = new ArrayList<Data>();

    //네트워킹 작업은 메인스레드에서 처리하면 안된다. 따로 스레드를 만들어 처리하자
    Thread guro = new Thread() {
        @Override
        public void run() {
            try {
                //url과 관련된 부분
                String fullurl = apiUrl + "&returnType=XML" + "&serviceKey=" + apiKey;
                URL url = new URL(fullurl);
                InputStream is = url.openStream();
```

Step.2

```
libs
├── libDaumMapAndroid.jar
├── release
├── src
│   ├── androidTest
│   └── main
│       └── java
│           └── jniLibs
│               ├── arm64-v8a
│               │   ├── libDaumMapEngineApi.so
│               ├── armeabi
│               │   ├── libDaumMapEngineApi.so
│               ├── armeabi-v7a
│               │   ├── libDaumMapEngineApi.so
```

흡연장, 주차장, 화장실의 위치를 지도에 표시하기 위해 카카오 지도 연동

Step.4

```
class Data {
    String name;
    Double latitude;
    Double longitude;

    public void setName(String name) { this.name = name; }

    public String getName() { return name; }

    public void setLatitude(Double latitude) { this.latitude = latitude; }

    public Double getLatitude() { return latitude; }

    public void setLongitude(Double longitude) { this.longitude = longitude; }

    public Double getLongitude() { return longitude; }

    @Override
    public String toString() {
        return "Data{" +
            "name='" + name + '\'' +
            ", latitude=" + latitude +
            ", longitude=" + longitude +
            '}';
    }
}
```

Step.5

```
if(type == XmlPullParser.START_TAG) {
    if (parser.getName().equals("col")) {
        if (parser.getAttributeValue(0).equals("지번주소"))
            bName = true;
        else if (parser.getAttributeValue(0).equals("위도"))
            bLat = true;
        else if (parser.getAttributeValue(0).equals("경도"))
            bLong = true;
    }
}

//내용 확인
else if(type == XmlPullParser.TEXT) {
    if (bName) {
        name = parser.getText();
        bName = false;
    } else if (bLat) {
        latitude = parser.getText();
        bLat = false;
    } else if (bLong) {
        longitude = parser.getText();
        bLong = false;
    }
}

//내용 다 읽었으면 데이터 추가
else if (type == XmlPullParser.END_TAG && parser.getName().equals("item")) {
    data.setName(name);
    data.setLatitude(Double.valueOf(latitude));
    data.setLongitude(Double.valueOf(longitude));

    dataArr.add(data);
}
```


Step.6

쉽게 볼 수 있는
광고와 위치



Step.1

```
public void checkDay(int cYear,int cMonth,int cDay,String userID){  
    fname="" +userID+cYear+"-"+(cMonth+1)+"-"+cDay+".txt";//저장할 파일 이름설정  
    FileInputStream fis=null;//FileStream fis 변수  
  
    try{  
        fis=openFileInput(fname);  
  
        byte[] fileData=new byte[fis.available()];  
        fis.read(fileData);  
        fis.close();  
    }  
}
```

Step.2



Step.3



FileInputStream을 통하여, 해당 날짜에 대한 경로로 파일이 생성되고, 해당 파일을 통해 일정이 저장.

어플리케이션 설문조사



어플리케이션 피드백 설문 조사

이 설문 조사에 참여해 주셔서 감사합니다.
귀하의 피드백은 향후 어플리케이션을 개선하는 데 많은 도움이 될 것입니다.

*는 필수항목입니다.

1. [화트레이스]에 대해 평가해주세요.*

	부족함	평균이하	보통	평균이상	탁월함
UI 설계 정도	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
기능의 품질	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
지역구에 맞는 정보	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
기능의 만족도	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. [카페, 맛집]에 대해 평가해주세요.*

	부족함	평균이하	보통	평균이상	탁월함
UI 설계 정도	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
기능의 품질	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
지역구에 맞는 정보	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

안드로이드 Travel APP 설문조사

작성 시간: 2022. 11. 12. 21:49 수정 삭제

링티로 리뷰팅 공식물

간소한 요건 확실한 리뷰팅은 정말 중요합니다. 최대 5만원 할인혜택까지

안녕하세요, 저는 개발자의 꿈을 꾸고 있는 사람으로 현재 졸업생입니다.
이번에 졸업 작품으로 앱을 만들게 되었고, 이 앱에 대한 여러분들의 만족도를 조사하고 싶습니다.
참여해주신 모든분들 소중한 시간 내주셔서 정말 감사합니다.
앱을 설치하실 시간이 없으시다면, 아래 사진들과 영상을 참고하여 평가해주시면 감사하겠습니다

설문조사 링크
<https://form.office.naver.com/form/responseView.cmd?formkey=N2FmZDc5Y2E1MmY4NS00YVVO5LTpSMGU1ZjVhNmYyYVYj1&sourceid=urlshare>

N 오피스

어플리케이션 피드백 설문 조사

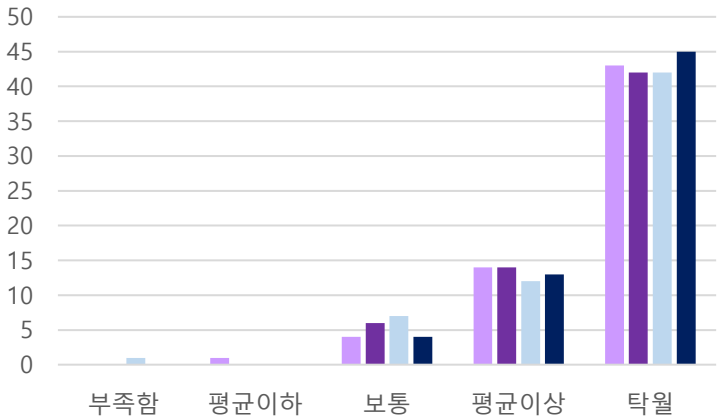
이 설문 조사에 참여해 주셔서 감사합니다. 귀하의 피드백은 향후 어플리케이션을 개선하는 데 많은 도움이 될 것입니다.

form.office.naver.com

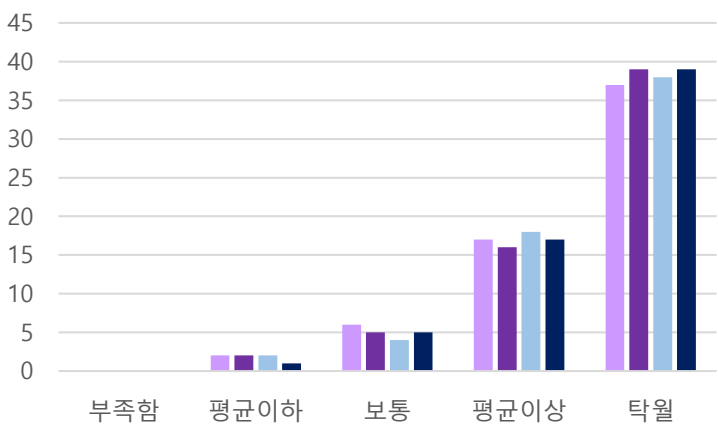
배포 방식

- ✓ 인스타그램 활용
- ✓ 네이버 설문조사
- ✓ 티스토리 활용

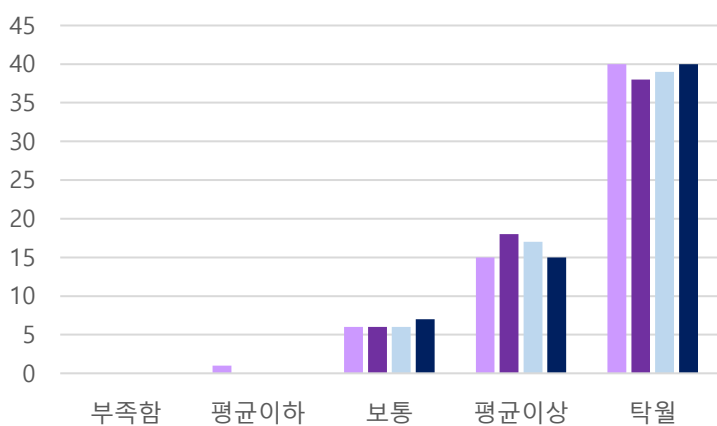
확률레이스



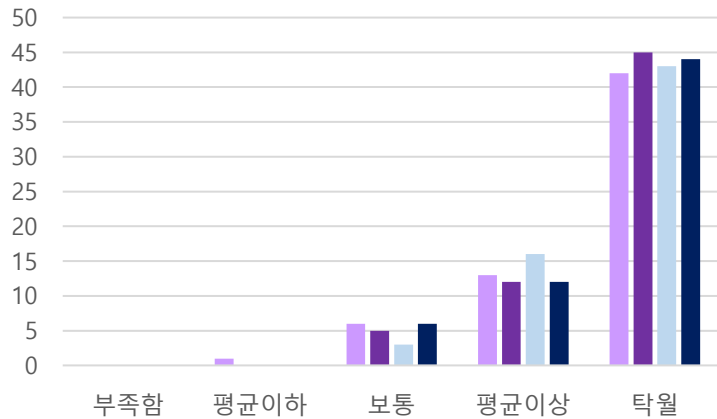
주차장



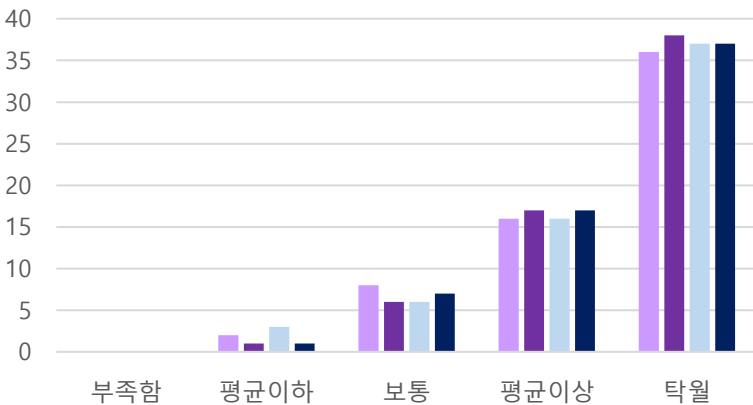
화장실



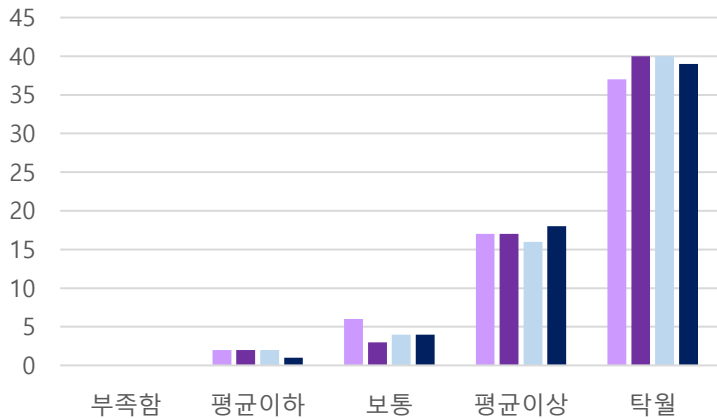
맛집 / 카페



흡연장



스케줄러



어플리케이션 별점 - 4.68 점



어플리케이션 유저 의견



- ✓ 다양한 기능들을 갖춰서 **향후 개발 가능성이 매우 뛰어나다고** 생각해요!
- ✓ 원하는 정보를 쉽게 찾을 수 있어서 **편리**하네요!
- ✓ 일일이 찾아보지 않아도 되서 여행 갈 때 굉장히 편리하겠어요
- ✓ 위치도 뜨는 것이 너무 만족스럽네요! 핫플레이스와 카페도 위치가 뜨면 좋겠네요
- ✓ **평소에 이런 앱이 있었으면 좋겠다고** 생각 했었는데, 유용할 것 같아요!
- ✓ 핫플레이스와 카페는 디자인 적 요소들이 좋았지만,
흡연장과 주차장 등은 평범한 디자인이라 아쉬운 부분이 있었습니다.
- ✓ UI 설계도, 디자인 적 요소들도 취향저격이었습니다.
- ✓ 보완해서 **정식 출시 되었으면 좋겠어요!**



- ✓ **여행 다닐 때 매우 유용**할 것 같아요! 다만 정보가 조금 더 많았으면 좋겠네요!!
- ✓ 다양하고 유용해서 좋음. 하지만, 정보가 조금 부족하고 **보완할 부분이 조금 보임.**
- ✓ 안 그래도 이런 앱이 있었으면 좋겠다고 생각했는데, 너무 유용하네요!!
- ✓ 디자인도 깔끔하고 다양하게 구성되어 있어서 좋네요!
- ✓ Ui, ux는 더 다듬어야 할 것 같고 **공식 홈페이지랑 연결되는 url이 있었으면 좋겠다고** 생각함. 확실히 메리트가 있는 어플인 것 같음
- ✓ 아이디어는 정말 좋은 것 같음.
- 인스타를 자주 이용하는 사람으로 써, **다양한 정보가 담겨있어서 상당히 유용함.**
- ✓ 여러가지 어플 들의 아쉬운 점들을 **한 곳에 담아서 더 좋은 것 같아요!!**

“바로여기” 앱의 기능

여행지의 필요한
모든 정보가 담겨 있는 만능앱

유용한 피드백

설문조사를 통하여,
부족한 점들을 보완 후 상용화



보편적인 사용성

내가 가고자 했던 여행지에 대한 정보들을 찾으려,
다양한 플랫폼들을 이용하여
정보를 일일이 찾는 수고를 덜어줄 앱

다양한 요소

- ✓ 핫플레이스: 가고자 하는 지역의 핫플레이스
- ✓ 맛집, 카페: 가고자 하는 지역의 핫한 맛집, 카페
- ✓ 흡연장: 가고자 하는 지역의 흡연장의 위치
- ✓ 주차장: 가고자 하는 지역의 주차장의 위치
- ✓ 화장실: 가고자 하는 지역의 화장실의 위치
- ✓ 스케줄러: 5가지 기능을 갖춘 스케줄러



향후 과제

- ✓ 시간 부족으로, 많은 정보가 담기지 못한 부분에 더욱 많은 정보를 추가
- ✓ 흡연장, 화장실, 주차장에 지도에 마커에 주소만 표시할 뿐만 아니라, 해당 주소지의 이미지 및 정보 제공
- ✓ 핫플레이스, 맛집, 카페 부분에도 카카오맵 API를 적용하여, 현재 내 위치 기반에 있는 주변 맛집 카페들을 탐색
- ✓ 앱 전체적인 구성 및 디자인을 보완
- ✓ 스케줄러의 더욱 다양한 기능 추가