



애완동물 데이팅 어플 PET WORLD

20185280 강다윤

20171601 김정엽

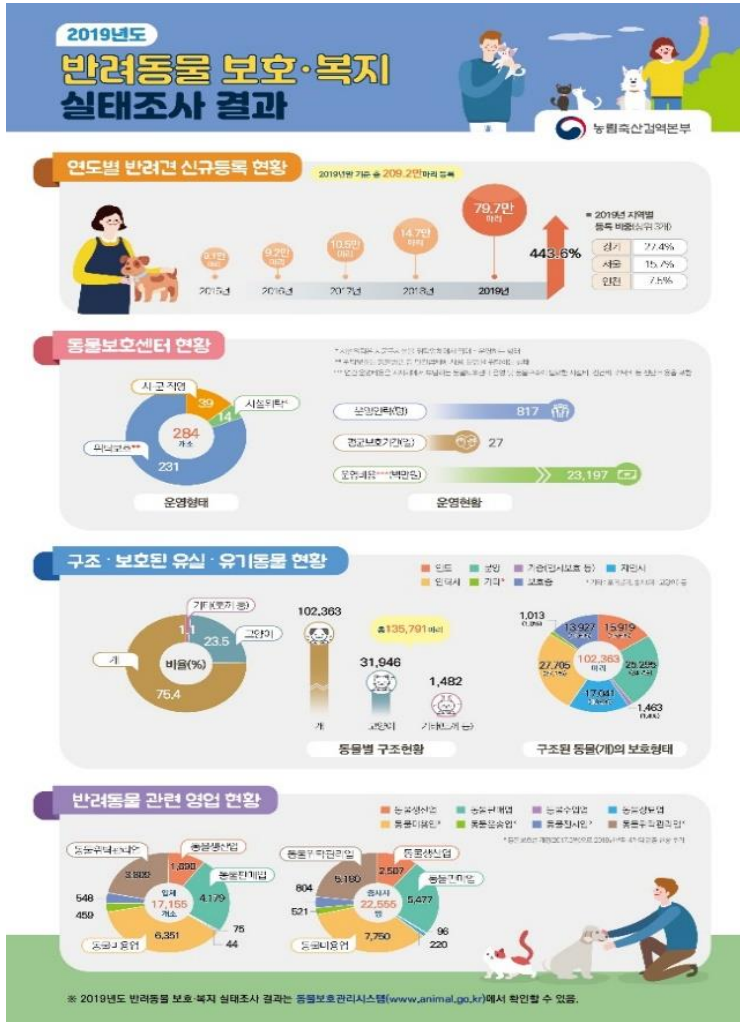
20171617 노지민

착안

반려동물들도 친구를 원해요!

다른 동물들을 만나는 것을 좋아하는 반려동물에게는 친구를 만들어주고,
사회성이 부족한 반려동물에게는 친구를 만들어줌으로써 사회성을 길러줄 수 있다. (기대효과)

시장 조사 : 전망



⇒ 연도별 반려견 신규등록 현황, 반려동물 관련 영업 현황 증가 상태

2019년 신규 등록 반려견 79만 7,081마리로 전년대비 443.6% 증가,
등록된 반려견 누계로 209만 2,163마리

반려동물 관련 영업장 총 1만 7,155개소, 종사자 약 2만 2,555명으로
전년 대비 각각 27.2%, 35.8% 증가

시장 조사 : 어플



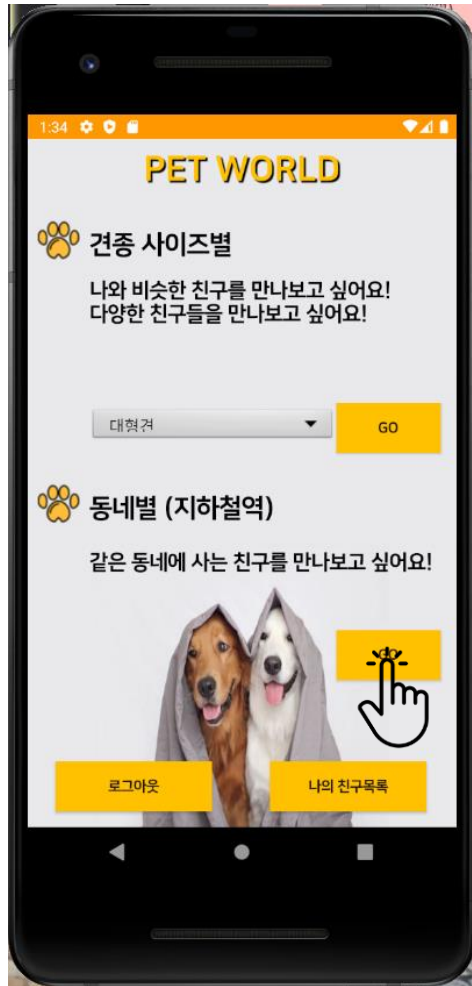
오픈 베타 단계. (미국)

반려동물들의 사진을 업로드하고 호감의 표시인
'하트'를 받을 때까지 기다림.

반려동물 할로윈 분장 코스튬 콘테스트 등
이벤트 진행.

⇒ 이 어플은 반려동물 소개팅이 목적이라면,
우리는 원하는 지역에서 반려동물의 친구를
만들어주는 것이 목적

1. 지하철역 searchview & listview



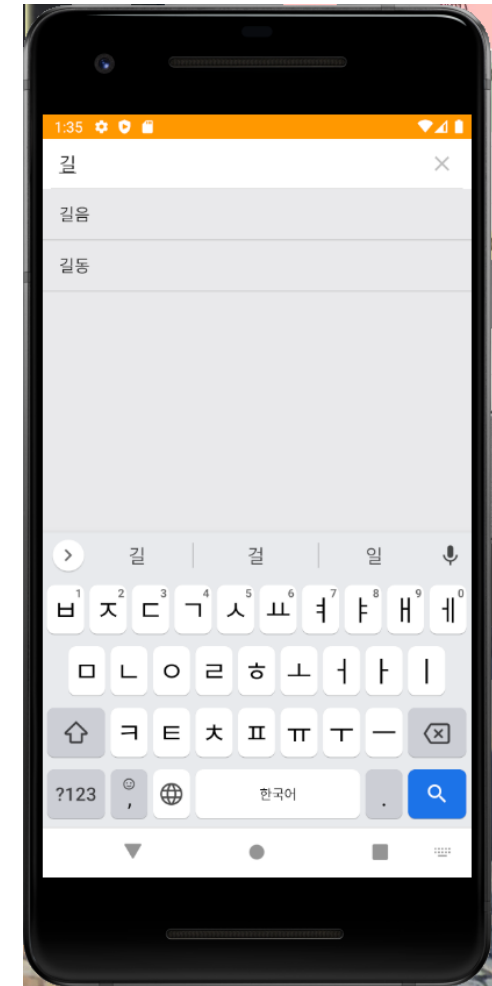
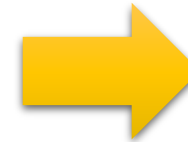
ThirdActivity View

동네별
선택



FifthActivity View

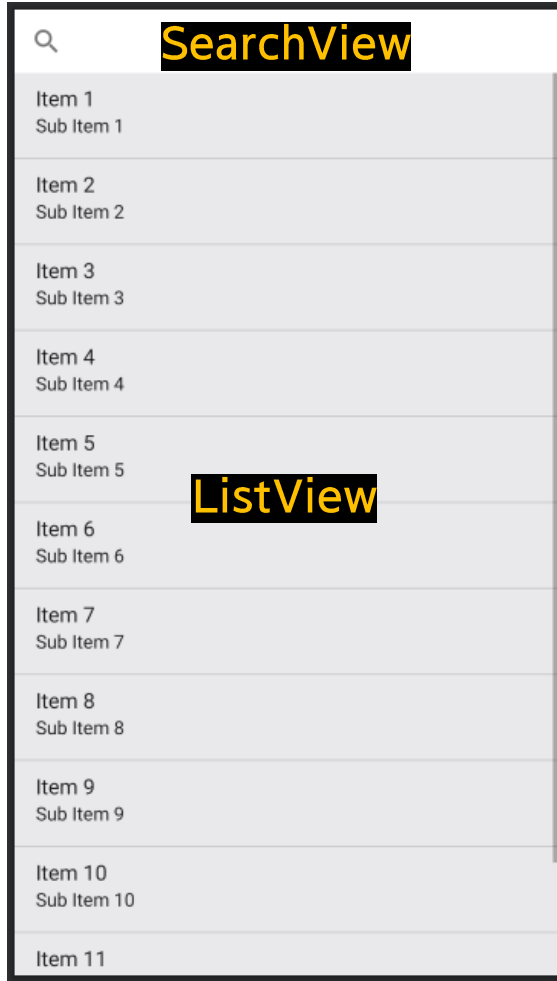
역 입력



FifthActivity View

1. 지하철역 searchview & listview

해당 XML에 SearchView와 ListView를 배치



activity_fifth.xml(design)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/listLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity"
    android:background="@drawable/background_color">

    <SearchView
        android:id="@+id/search_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#ffffff"/>

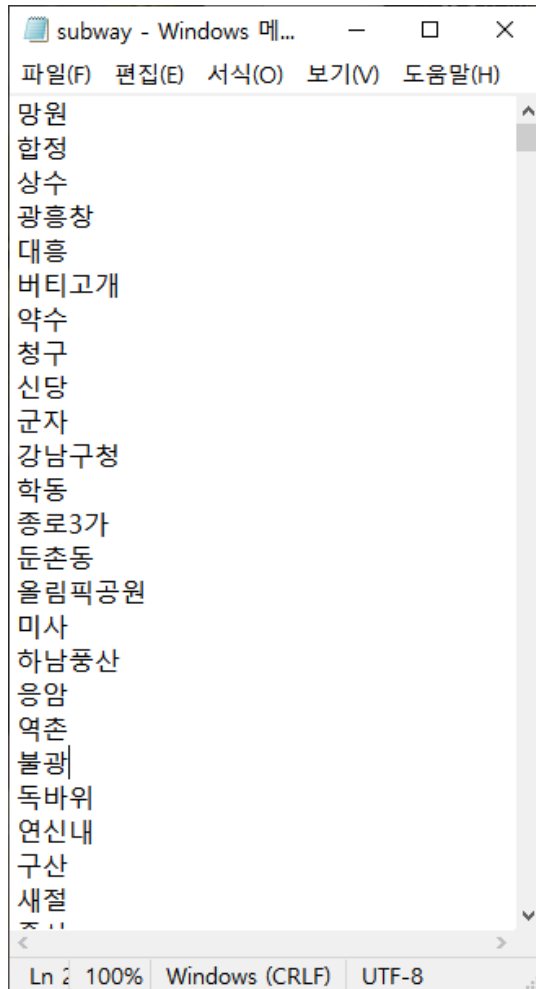
    <ListView
        android:id="@+id/listview"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

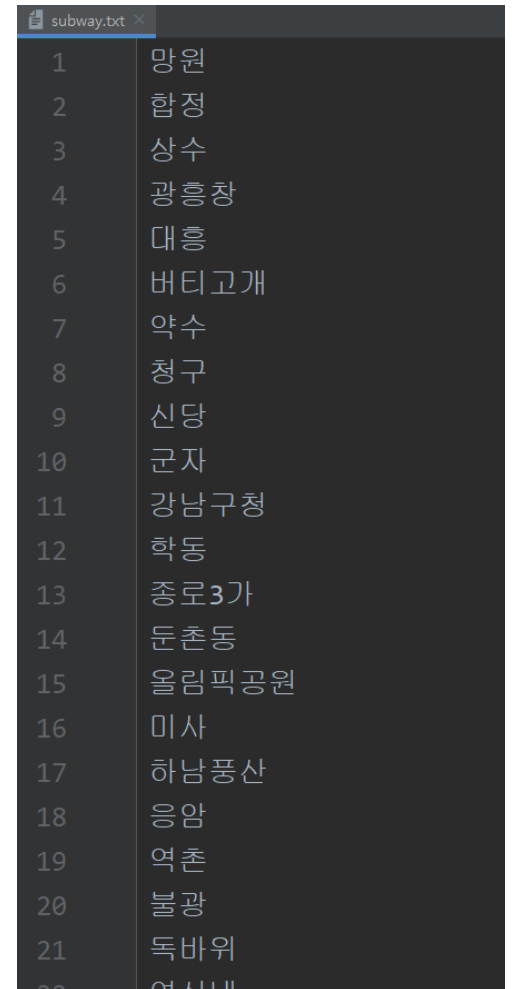
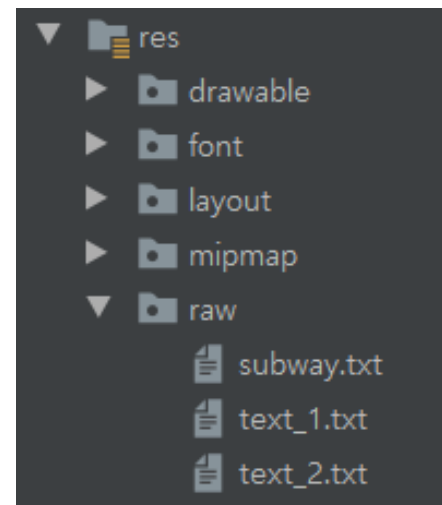
activity_fifth.xml(code)

1. 지하철역 searchview & listview

지하철역 이름이 담긴 text file을 res/raw/ 에 넣어줌



subway.txt



1. 지하철역 searchview & listview

FifthActivity.java

```
public class FifthActivity extends AppCompatActivity{

    ListView listview;
    String[] subway;
    int intCount = 0;

    // text file의 줄 카운팅함.
    InputStream inputStreamCounter;
    BufferedReader bufferedReaderCounter;

    // text file의 values를 String array에 로딩.
    InputStream inputStreamLoader;
    BufferedReader bufferedReaderLoader;
```

지하철역 정보를 담을 String array 선언

지하철역 개수를 카운팅할 변수 선언

Text file을 읽고 String array에 저장해줄
로더와 카운터를 만들어줌

1. 지하철역 searchview & listview

FifthActivity.java

```
// is & br를 text file과 연결.
// counters
InputStreamCounter = this.getResources().openRawResource(R.raw.subway);
BufferedReaderCounter = new BufferedReader(new InputStreamReader(inputStreamCounter));

// loaders
InputStreamLoader = this.getResources().openRawResource(R.raw.subway);
BufferedReaderLoader = new BufferedReader(new InputStreamReader(inputStreamLoader));

// text file의 줄/라인 카운팅.
try {
    while(BufferedReaderCounter.readLine() != null){
        intCount++;
    }
} catch (Exception e){
    e.printStackTrace();
}

subway = new String[intCount];

// string array에 text file의 줄/라인 저장.
try{
    for(int i=0 ; i<intCount ; i++){
        subway[i] = BufferedReaderLoader.readLine();
    }
} catch (Exception e){
    e.printStackTrace();
}
```

카운터와 로더에게 text file을 지정해줌

text file의 line을 카운팅해준 후,

String array에 text file에 있는 라인을 한 줄 씩 넣어줌

1. 지하철역 searchview & listview

FifthActivity.java

```
// Adapter 적용.
ArrayAdapter < String > adapter = new ArrayAdapter<>(context: this, android.R.layout.simple_list_item_1, android.R.id.text1, subway);

listview.setAdapter(adapter);

// 접속중인 ID 값을 String으로 받음.
Intent intent = getIntent();
final String preId = intent.getStringExtra( name: "preId");

// listview의 item 클릭 시 기능 구현.
listview.setOnItemClickListener((parent, view, position, id) → {
    // 선택된 지하철역 값 intent 선언.
    Intent intent = new Intent(view.getContext(), SixthActivity.class);
    String subwayselection = (String) parent.getItemAtPosition(position);

    // 선택된 지하철역 값과 접속중인 ID 값을 넘겨줌.
    intent.putExtra( name: "subwayselection", subwayselection);
    intent.putExtra( name: "preId", preId);

    // 액티비티 실행.
    startActivity(intent);
});
}
```

String array 를
ArrayAdapter에 적용

지하철 역 선택 시,
지하철 역의 값과 접속중인 ID
값을 **intent**를 이용해 다음
액티비티로 넘김

1. 지하철역 **searchview** & listview

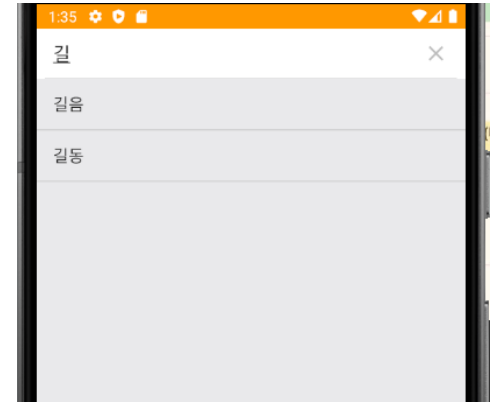
FifthActivity.java

```
// SearchView 기능 구현.
searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {

    // 검색 버튼이 눌러졌을 때 이벤트 처리.
    @Override
    public boolean onQueryTextSubmit(String query) { return false; }

    // 검색어가 변경되었을 때 이벤트 처리.
    @Override
    public boolean onQueryTextChange(String newText) {
        ArrayList<String> filterList = new ArrayList<>();

        // 검색어와 일치하는 값을 보여줌.
        for (String tempSubwayList : subway){
            if(tempSubwayList.toLowerCase().startsWith(newText.toLowerCase())){
                filterList.add(tempSubwayList);
            }
        }
        // Adapter 적용.
        ArrayAdapter<String> adapterWithFilteredList = new ArrayAdapter<>(getApplicationContext(), android.R.layout.simple_list_item_1, filterList);
        listview.setAdapter(adapterWithFilteredList);
        return true;
    }
});
```

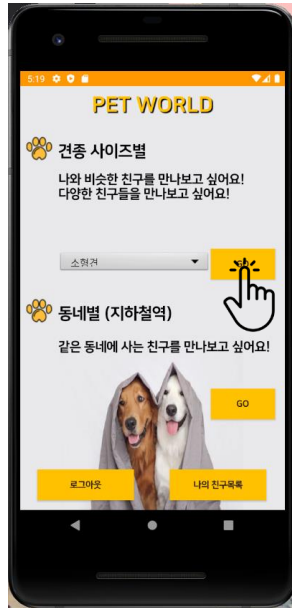


onQueryTextChange로
검색어가 변경되었을 때
이벤트 처리

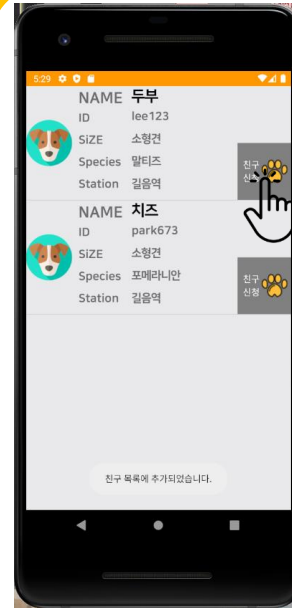
검색어와 일치하는 값들을
String ArrayList
filterList에 add하여
보여줌

ArrayAdapter에 적용시
킴

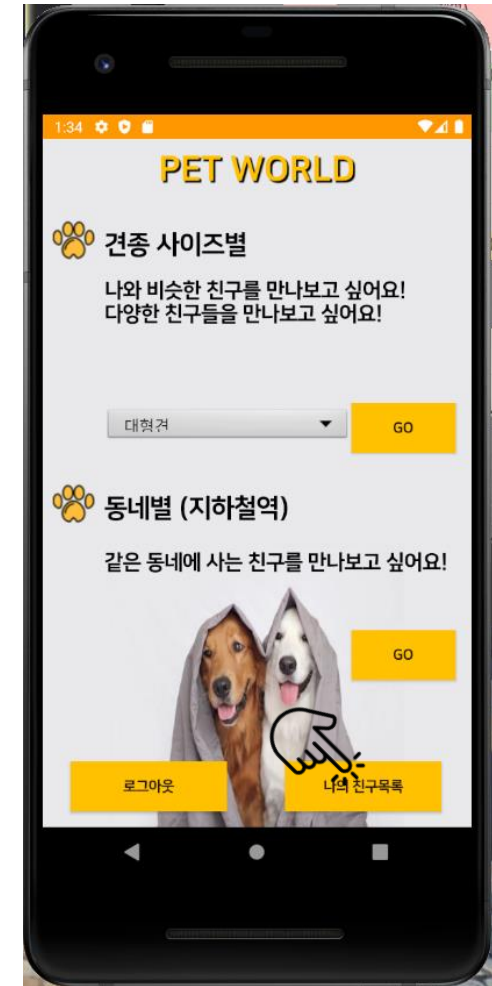
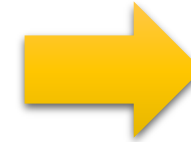
2. 선택별 친구 목록



‘소형견’ 선택



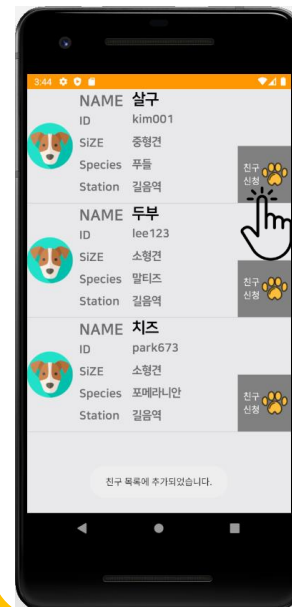
친구 신청 후,
나의 친구목록



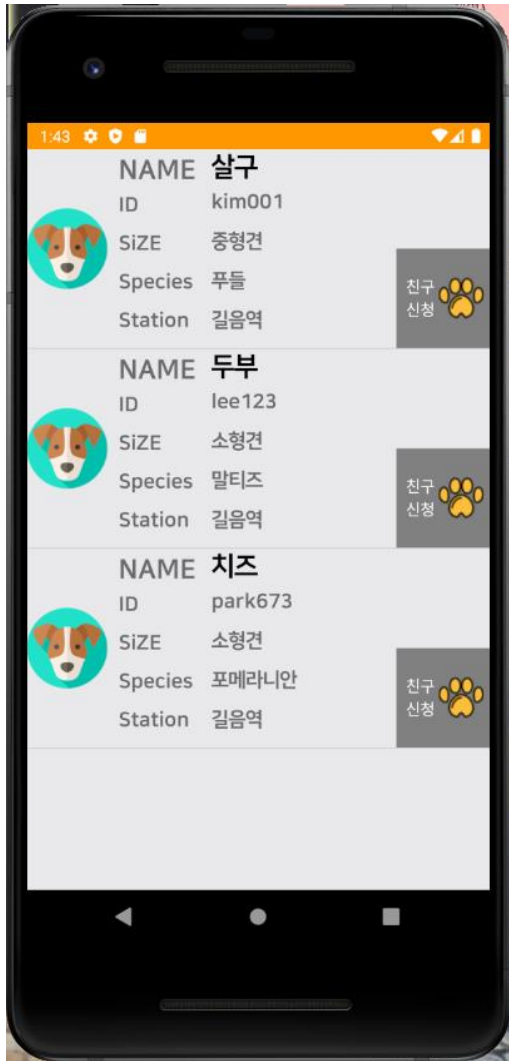
ThirdActivity View



‘길음역’ 선택

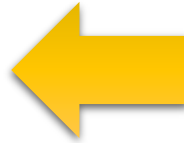


2. 선택별 친구 목록



SixthActivity View

Firestore의 database에서
각각의 데이터를 받아와야 함



Firestore의 Realtime Database

2. 선택별 친구 목록

Information.java

```
// firebase에서 받아온 reference를 하나씩 가져오기 위한 class
public class Information {

    private String id;
    private String pw;
    private String name;
    private String size;
    private String kinds;
    private String station;

    public Information() {}

    // 생성자
    public Information(String id, String pw, String name, String size, String kinds, String station) {...}

    // getmethod(), setmethod() 생성
    public String getId() { return id; }

    public void setId(String id) { this.id = id; }

    public String getName() { return name; }

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

    public String getSize() { return size; }

    public void setSize(String size) { this.size = size; }

    public String getKinds() { return kinds; }

    public void setKinds(String kinds) { this.kinds = kinds; }

    public String getStation() { return station; }

    public void setStation(String station) { this.station = station; }

    public String getPw() { return pw; }
    public void setPw(String pw) { this.pw = pw; }
}
```

id, pw, name, size, kinds, station 변수명을 갖는
String 선언

선언된 모든 변수의 getter, setter 메소드 생성

2. 선택별 친구 목록

FourthActivity.java

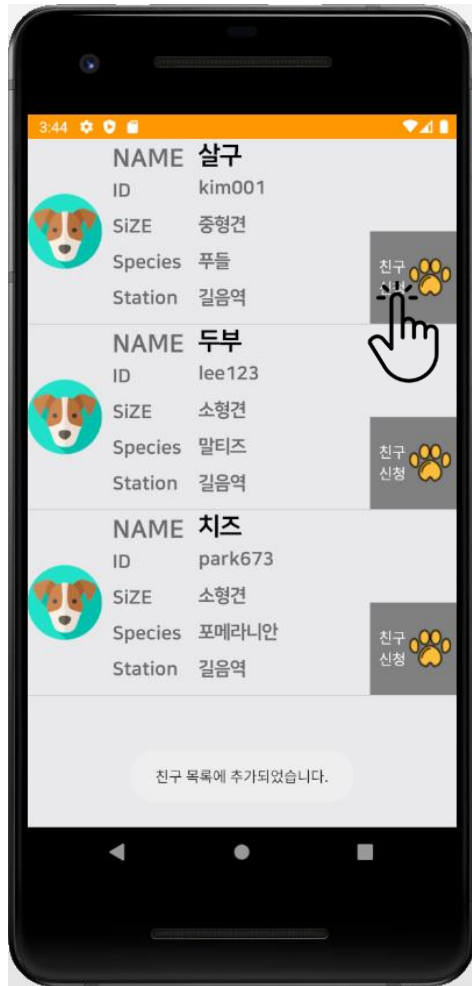
```
// Adapter에 추가 기능
// child에 있는 값들을 for문을 돌며 순차적으로 messageData에 저장
for (DataSnapshot messageData : dataSnapshot.getChildren()) {
    final Information info = messageData.getValue(Information.class); // infomation를 이용해 값을 전달
    // 종류(대형견,중형견...)가 같고 가져온 아이디가 현재 접속중인 아이디와 다르면 실행
    if(sizeofpet.equals(info.getSize()) && !preId.equals(info.getId())){
        adapter.addItem(ContextCompat.getDrawable(getApplicationContext(), R.drawable.dogimage), // Adapter에 아이템 추가
            info.getId(),info.getName(), info.getSize(), info.getKinds(), info.getStation());
    }
}
```

SixthActivity.java

```
// 데이터 읽어오기.
for (DataSnapshot messageData : dataSnapshot.getChildren()) {
    Information info = messageData.getValue(Information.class);
    // 선택된 지하철역 값과 지하철 리스트의 값이 동일하고 접속중인 ID 값과 가져온 User ID 값 다를 시,
    // Adapter에 해당 User 프로필 add.
    if((subwayselection+"역").equals(info.getStation()) && !preId.equals(info.getId())){
        adapter.addItem(ContextCompat.getDrawable(getApplicationContext(), R.drawable.dogimage),
            info.getId(),info.getName(), info.getSize(), info.getKinds(), info.getStation());
    }
}
```

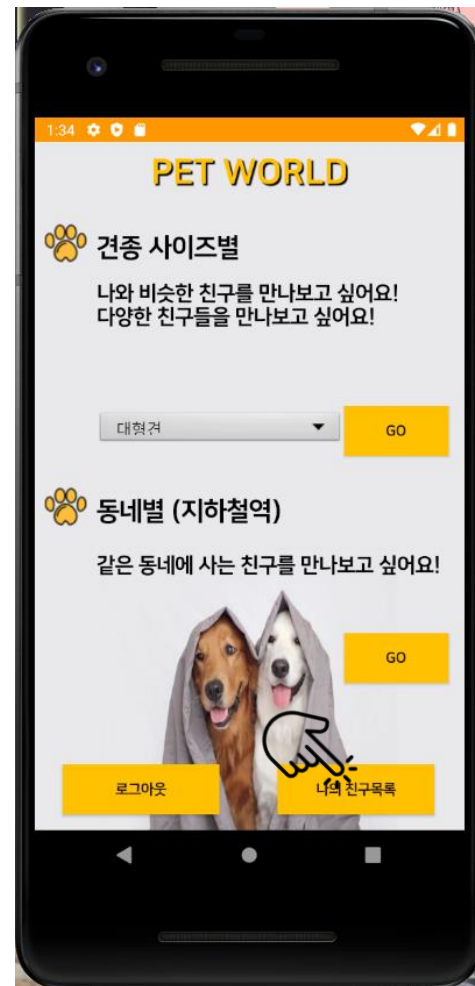
If문의 조건 충족 시,
Information Class의 get 메소드들을 통해
Id, name, size, kinds, station 값들을
Adapter에 추가함

3. 나의 친구 목록



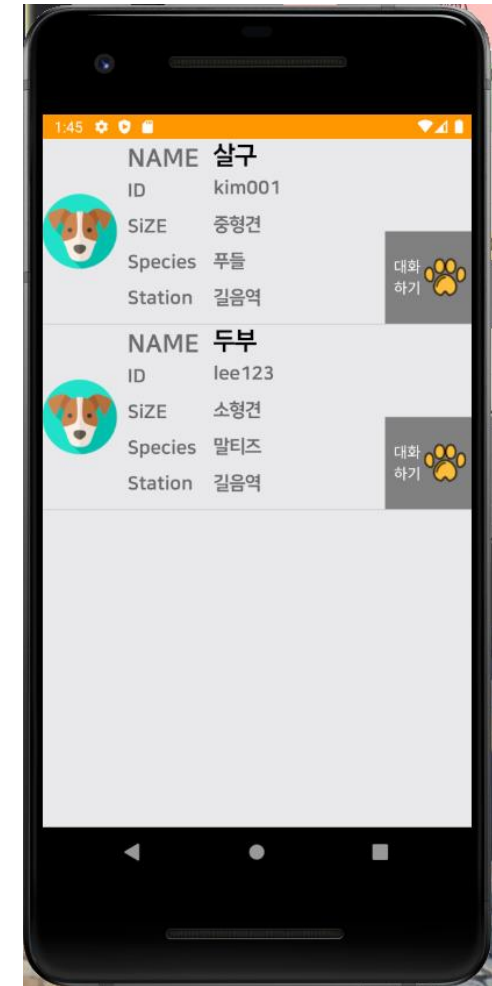
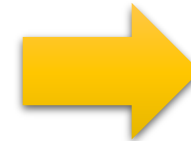
SixthActivity View

친구 신청



ThirdActivity View

친구 목록
확인



SeventhActivity View

3. 나의 친구 목록

MainActivity.java

```
public static Context mContext;           // 현재 액티비티

private String IdStorage = new String(); // 현재 아이디 값

public String getIdStorage() { return IdStorage; }
```

SeventhActivity.java

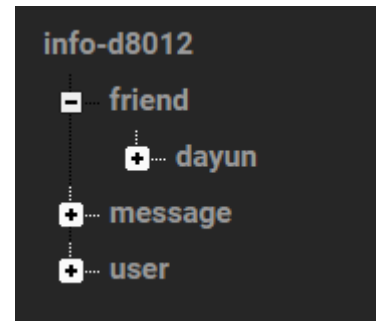
```
// 현재 데이터 베이스를 접근할 수 있는 진입점 받기.
mDatabase = FirebaseDatabase.getInstance();
mReference = mDatabase.getInstance().getReference();

// 파이어베이스에 프로젝트명/friend/User ID 형식으로 저장
DatabaseReference downloadsRef = mReference.child("friend").child(((MainActivity)MainActivity.mContext).getIdStorage());
```

현재 로그인된 ID는 MainActivity에서 입력된 ID값

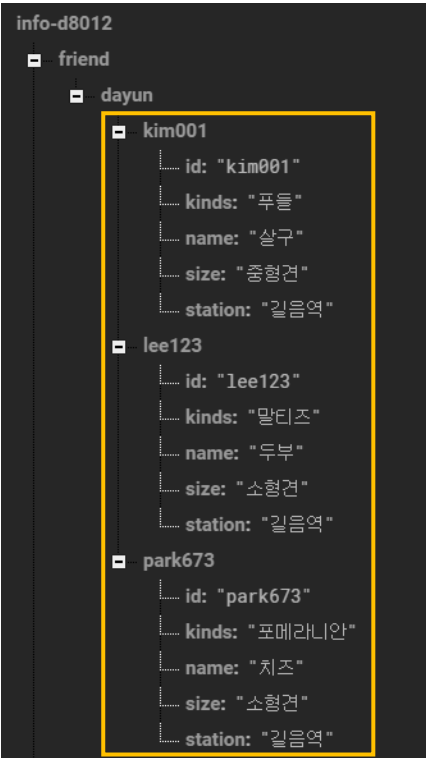
MainActivity에 Context를 생성한 뒤 getIdStorage를 통해 현재 ID를 가져옴

파이어베이스에 “프로젝트명”-“friend”-“User ID” 형식으로 쓰여짐

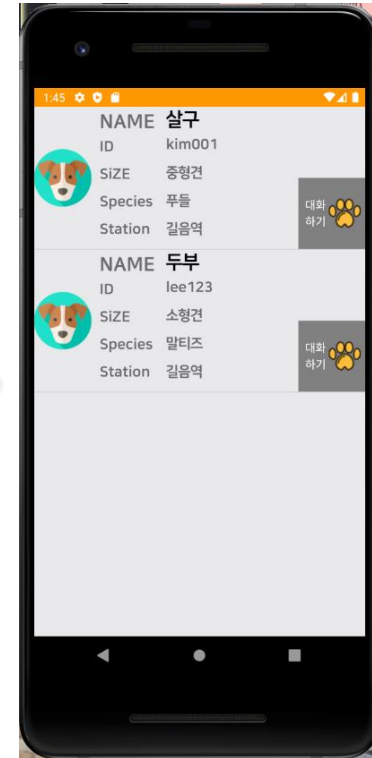
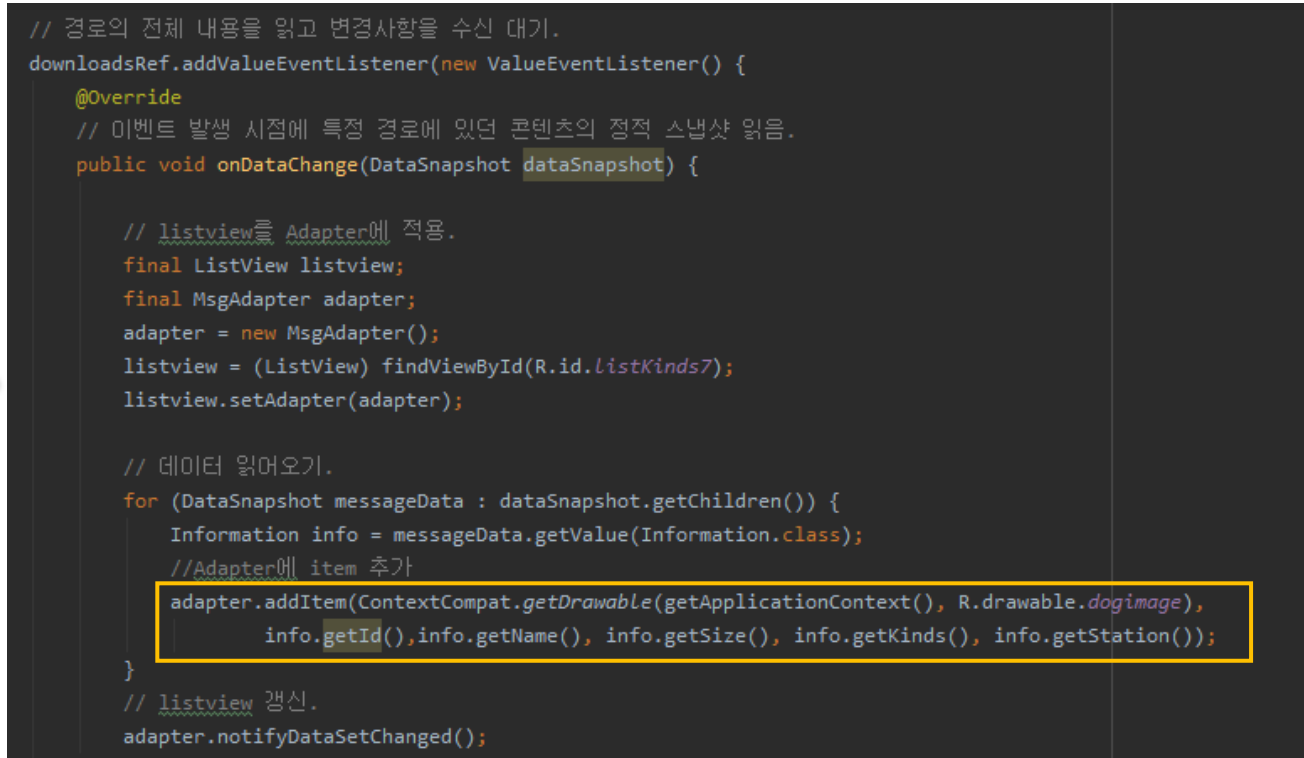


3. 나의 친구 목록

FireBase

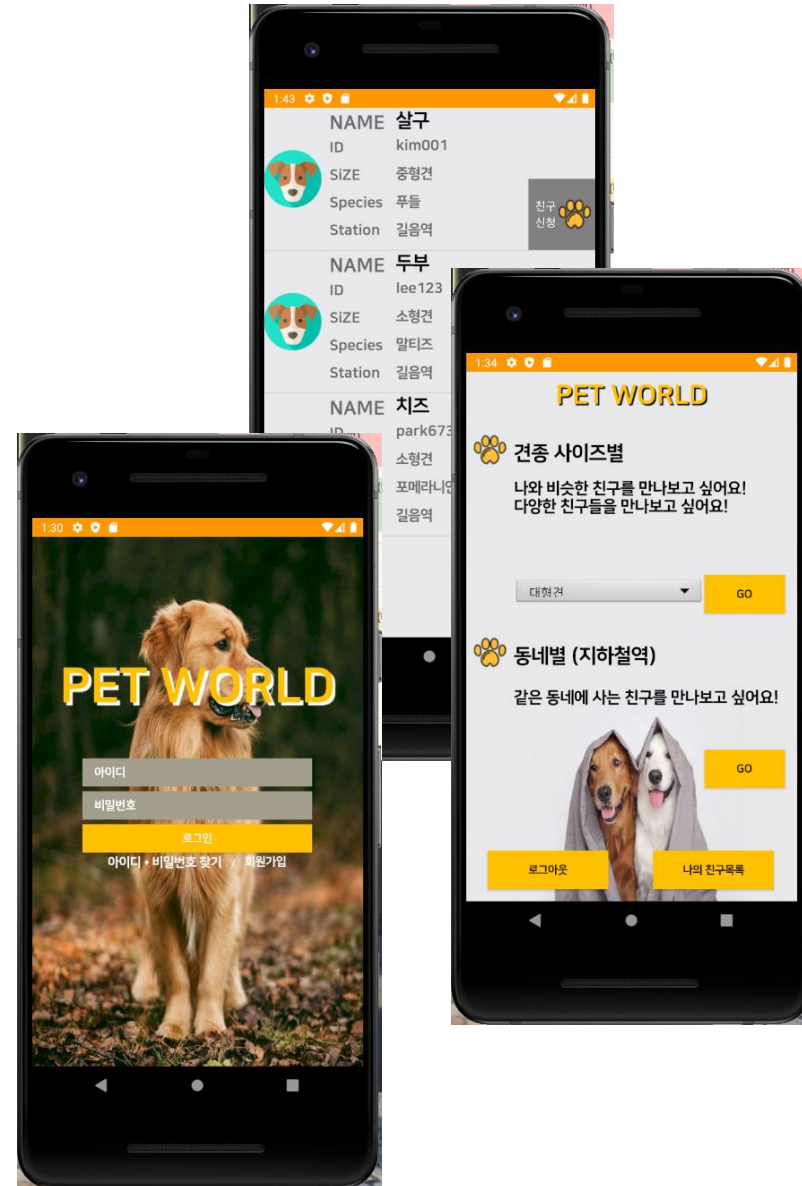
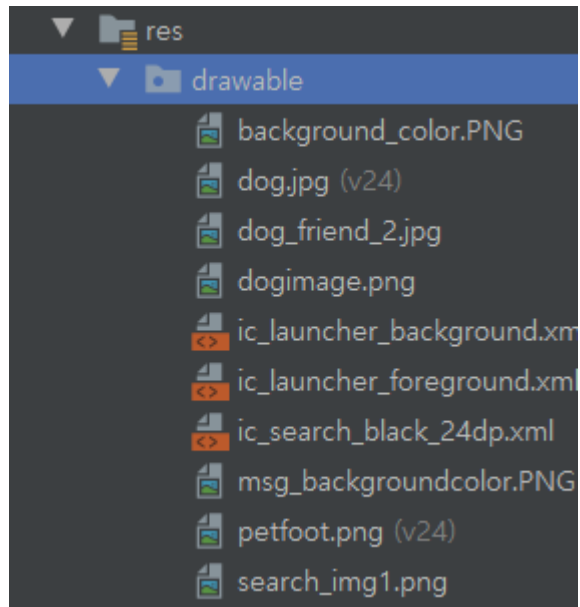


SeventhActivity.java



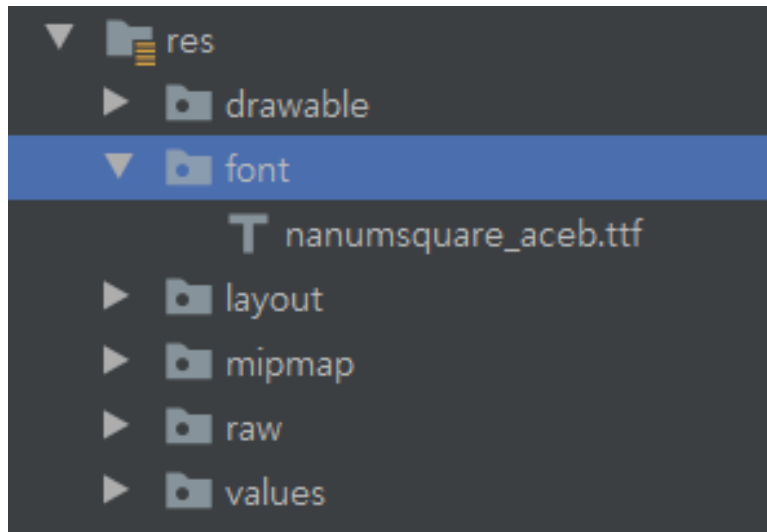
“프로젝트명” - “friend” - “User ID” 하위의 Children을 불러옴
Information Class의 get 메소드를 통해 각각의 데이터를
Adapter에 추가

4. UI : Image & Icon



어플과 어울리는 이미지와 아이콘을
선택하여 추가

4. UI : Font



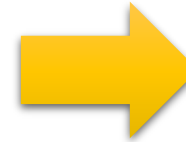
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:fontFamily="@font/nanumsquare_aceb"
    android:gravity="center_horizontal"
    android:text="PET WORLD"
    android:textColor="#FFC107"
    android:shadowColor="#000000"
    android:shadowDx="5"
    android:shadowDy="5"
    android:shadowRadius="1.5"
    android:textAlignment="center"
    android:textSize="33dp" />
```

res 디렉토리 하위에 font 디렉토리 추가
res/font/ 에 글꼴 추가

4. UI : Color & Style



```
styles.xml X
1  <resources>
2      <!-- Base application theme. -->
3      <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
4          <!-- Customize your theme here. -->
5          <item name="colorPrimary">#FFC107</item>
6          <item name="colorPrimaryDark">#FF9800</item>
7          <item name="colorAccent">#FFC107</item>
8          <item name="windowActionBar">false</item>
9          <item name="windowNoTitle">true</item>
10     </style>
11
12 </resources>
```



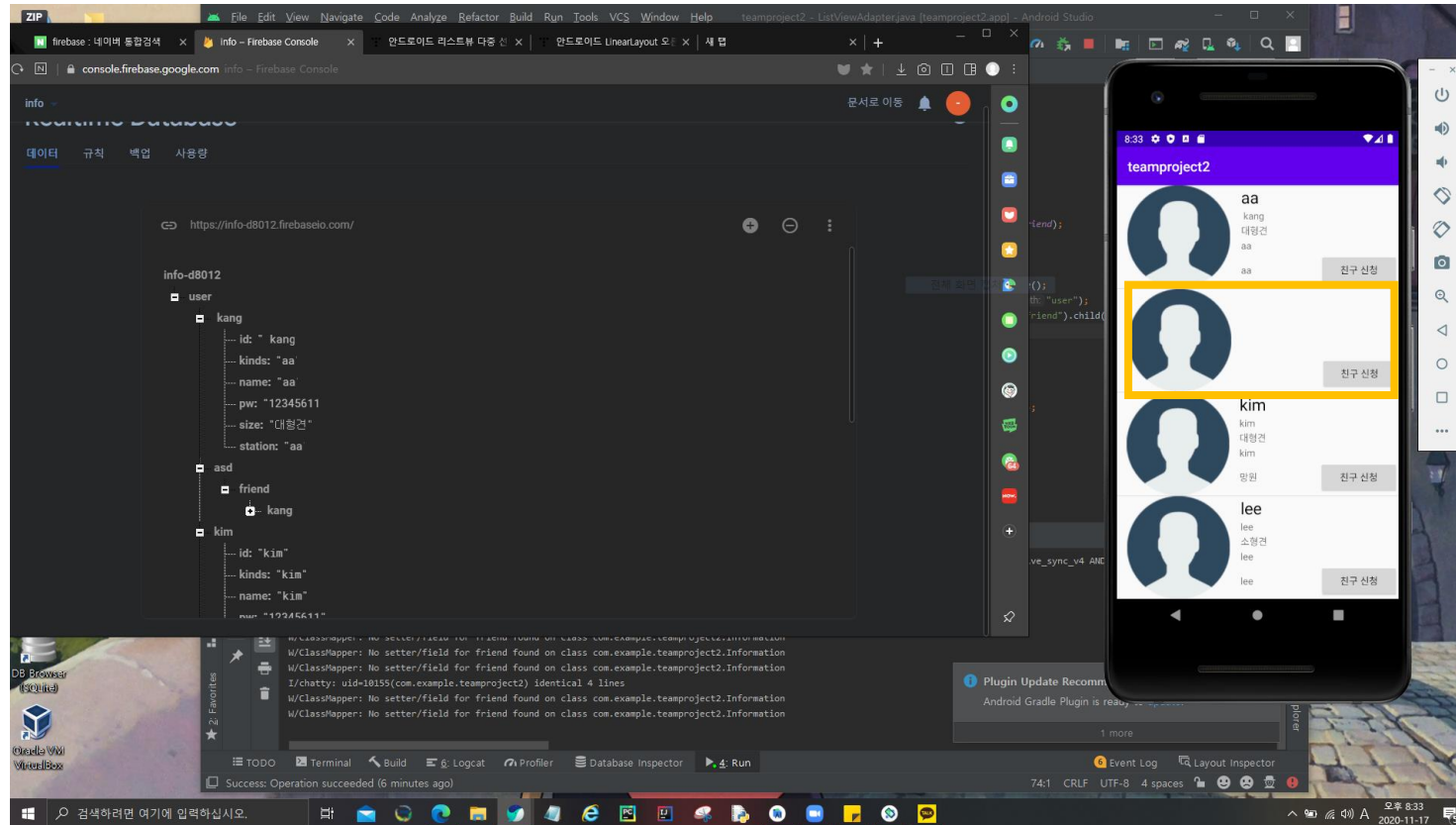
대표 색상을 선택해 디자인에 통일감을 줌
타이틀바를 삭제하여 더 깔끔하게 어플을 볼 수 있음

개발 시 발생한 문제점 및 해결방법



1. 처음 : 지하철 searchview를 menu에 만듦
문제 : 타이틀바를 없애고 나니 menu도 사라져 searchview 기능 사라짐
해결 : searchview 기능 구현을 위해 listview위에 다시 만듦

개발 시 발생한 문제점 및 해결방법



2. 처음 : 친구 목록을 firebase에 저장 시, '프로젝트명'-'user'-'friend'형식으로 저장함
문제 : 내가 선택할 수 있는 친구 목록에 나도 뜨게 됨(오류 발생)
해결 : 'user'와 동등한 위치인 '프로젝트명'-'friend'형식으로 저장함

발전방향과 개선점

개선점

- 회원가입 시, 등록한 사진을 프로필 사진으로 이용
- 친구 신청 시, 상대방에게 알림 메시지 설정

발전방향

- GPS 서비스 제공 -> 범죄 예방
- 반려견 훈련법 영상 제공 (산책 관련)
- 반려동물 정보 공유 게시판 생성(샵, 동물병원, 애완동반 카페, 음식점, 산책길)
- 애완용품 나눔 또는 중고거래 게시판 생성



감사합니다 :)