

A stylized graphic of a blue folder with a white sheet of paper inside. A blue pen is positioned diagonally across the top right corner of the paper. The background is a light blue dotted pattern.

졸업작품

교과목 포트폴리오

컴퓨터정보공학과 20202248 오예림

차례

- 졸업작품 프로젝트

- 기획
- 프로젝트 팀
- 프로젝트 계획
- 프로젝트 수정
- 프로젝트 구현
- 깃허브를 통한 협업
- 기능 추가 및 수정
- 최종 발표

- Android Studio

- navigation bar
- fragment
- sensor
- firebase

- 후기

졸업작품 프로젝트

- 기획
- 프로젝트 팀
- 프로젝트 계획
- 프로젝트 수정
- 프로젝트 구현
- 깃허브를 통한 협업
- 기능 추가 및 수정
- 최종 발표

기획

- 개별 기획 제안서
 - 모두 모이개

개요	지도 위에 견주들의 위치를 표시하고, 특정 장소에 대한 정보를 저장 및 공유한다.
주요 기능	<p>산책 모드</p> <ul style="list-style-type: none">- 산책모드를 활성화 하면 앱의 지도에 나의 위치와 산책모드를 활성화 하고 있는 이용자들의 위치가 지도에 표시된다- 지도에 표시된 다른 이용자들의 위치를 터치하면 해당 이용자의 반려동물 정보가 보여진다- 100m 이내에 산책모드를 활성화시키고 있는 앱 이용자가 있으면 알람이 온다 <p>장소 표시</p> <ul style="list-style-type: none">- 앱의 지도위에 원하는 위치에 핀(pin)을 찍는 형태로 표시를 할 수 있다.- 간단한 설명과 사진 첨부 가능- 표시한 장소는 다른 사람과 공유 가능- 아두이노와 연동하여 표시된 장소 근처에 가면 LED가 빛난다 <p>달력</p> <ul style="list-style-type: none">- 산책모드를 종료하면 이동하는 화면- 달력에서 산책 여부와 산책한 거리를 확인 할 수 있다- 날짜에 따라 간단한 일과표를 작성할 수 있다

프로젝트 팀

- 팀명

- 견주 개발자

- 팀원

- 김규식 (조장)
- 김시현
- 오예림
- 석현우
- 김송희

- 최종 프로젝트 선정

- 모두 모이개

- 프로젝트 발표


- ppt 제작 및 디자인

개발 아이템 소개 [1/3]


프로젝트 이름

모두 모이개

개요



지도 위에 견주들의 위치를 표시한다.



특정 장소에 대한 정보를 저장 및 공유한다.

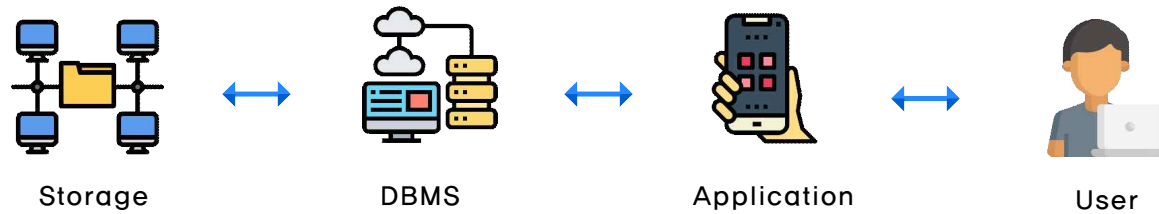
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21

달력으로 산책 일정을 관리한다.

프로젝트 계획

- 개발 환경
 - Android Studio
 - Fire Base
- 개발 구성 모듈

DBMS



Server



■ 깃허브 연동

- 코딩에 참여하는 팀원들이 깃허브로 프로젝트 연동

The screenshot shows a GitHub repository page for 'hyun29570 / Graduation_Project'. The repository is public and has 1 watch, 0 stars, and 0 forks. The main content area displays a file tree for the 'master' branch, showing files like '.idea', 'app', 'gradle/wrapper', '.gitignore', 'build.gradle', 'gradle.properties', 'gradlew', 'gradlew.bat', and 'settings.gradle'. The 'app' directory is highlighted. To the right of the file tree, there are sections for 'About', 'Releases', 'Packages', 'Contributors', and 'Languages'. The 'Contributors' section lists three contributors: yelim0308, hyun29570, and kuchic0918. The 'Languages' section shows that the repository is 100.0% Java.

Search or jump to... Pull requests Issues Marketplace Explore

hyun29570 / Graduation_Project Public Watch 1 Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Wiki Security Insights

master 2 branches 0 tags Go to file Add file Code

File	Commit	Time
yelim0308 member_init edit	0155e4b	21 days ago 76 commits
.idea	member_init edit	21 days ago
app	member_init edit	21 days ago
gradle/wrapper	Initial commit	2 months ago
.gitignore	Initial commit	2 months ago
build.gradle	rr	2 months ago
gradle.properties	Initial commit	2 months ago
gradlew	Initial commit	2 months ago
gradlew.bat	Initial commit	2 months ago
settings.gradle	rr	2 months ago

Help people interested in this repository understand your project by adding a README. Add a README

About No description, website, or topics provided.

Releases No releases published. Create a new release.

Packages No packages published. Publish your first package.

Contributors 3

- yelim0308 오예림
- hyun29570 김시현
- kuchic0918 KKS 김규식

Languages

Java 100.0%

프로젝트 수정

■ 기능 수정

- 산책기능 활성화 하고 있는 앱 이용자들의 위치를 지도에 표시하는 기능 및 지도에 장소 표시하는 기능 제거
 - 범죄 악용 우려
 - 참여 팀원 감소로 인한 무리한 기획 불가능
 - 장소 표시 기능을 제거함에 따라 아두이노 활용 계획도 무산
- 커뮤니티 기능 추가
 - 로그인을 통해 생성한 사용자 계정을 더 적극적으로 활용할 수 있고 사용자끼리의 보다 활발하고 편리한 소통 가능
- 병원 검색 기능 추가
 - 앱 이용자들의 편의성을 위해 추가
 - 공공 API 활용

■ 레이아웃 수정

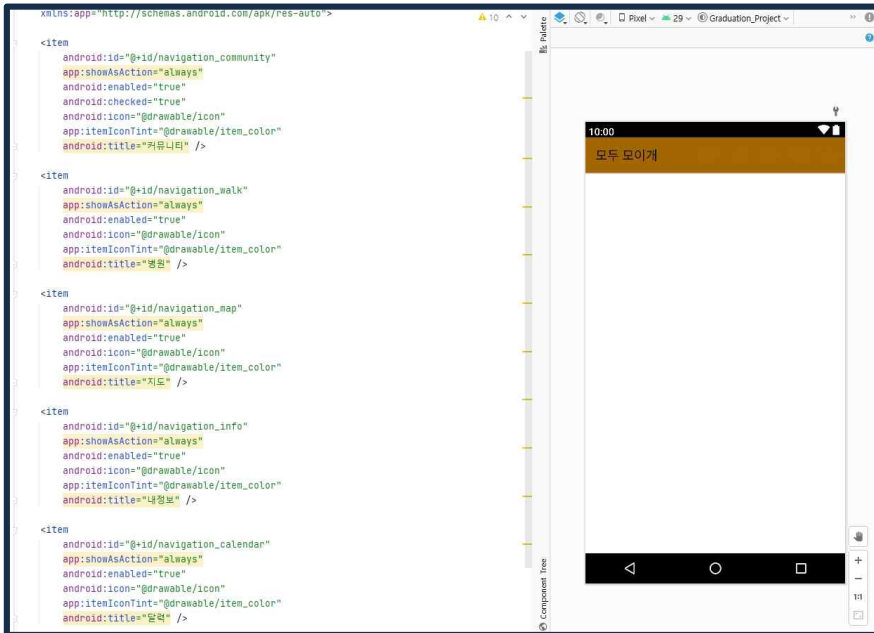
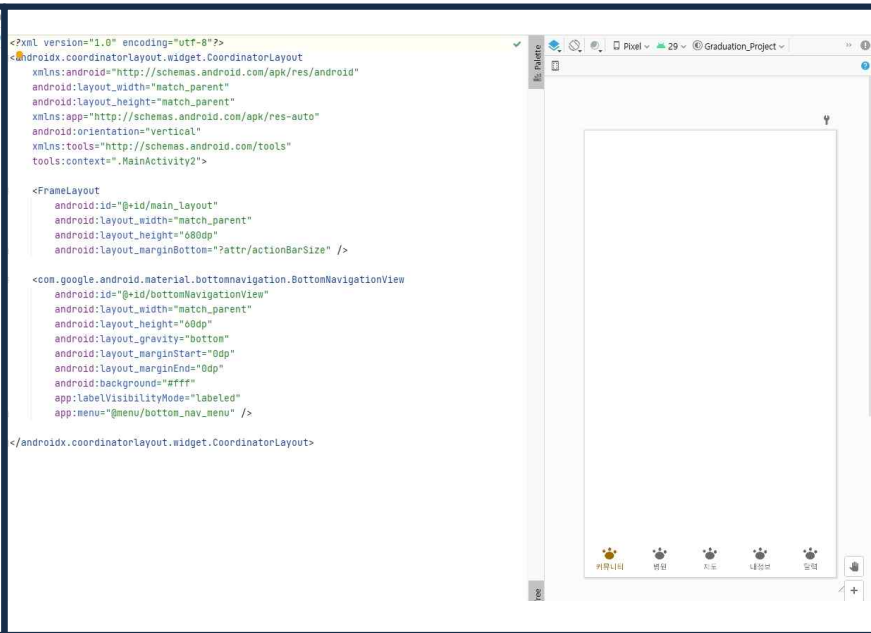
- 편의성을 위해 바텀 네비게이션 바 활용



프로젝트 구현

■ 메인화면 구현

- 메인 액티비티에 바텀 네비게이션 바를 구현
- 네비게이션바로 이동하는 화면은 프래그먼트로 구현

 <pre><item android:id="@+id/navigation_community" app:showAsAction="always" android:enabled="true" android:checked="true" android:icon="@drawable/icon" app:iconTint="@drawable/item_color" android:title="커뮤니티" /> <item android:id="@+id/navigation_walk" app:showAsAction="always" android:enabled="true" android:icon="@drawable/icon" app:iconTint="@drawable/item_color" android:title="병원" /> <item android:id="@+id/navigation_map" app:showAsAction="always" android:enabled="true" android:icon="@drawable/icon" app:iconTint="@drawable/item_color" android:title="지도" /> <item android:id="@+id/navigation_info" app:showAsAction="always" android:enabled="true" android:icon="@drawable/icon" app:iconTint="@drawable/item_color" android:title="내정보" /> <item android:id="@+id/navigation_calendar" app:showAsAction="always" android:enabled="true" android:icon="@drawable/icon" app:iconTint="@drawable/item_color" android:title="도움" /></pre>	 <pre><?xml version="1.0" encoding="utf-8"?> <include xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="match_parent" android:layout_height="match_parent" xmlns:app="http://schemas.android.com/apk/res-auto" android:orientation="vertical" xmlns:tools="http://schemas.android.com/tools" tools:context=".MainActivity2"> <FrameLayout android:id="@+id/main_layout" android:layout_width="match_parent" android:layout_height="580dp" android:layout_marginBottom="?attr/actionBarSize" /> <com.google.android.material.bottomnavigation.BottomNavigationView android:id="@+id/bottomNavigationView" android:layout_width="match_parent" android:layout_height="80dp" android:layout_gravity="bottom" android:layout_marginStart="0dp" android:layout_marginEnd="0dp" android:background="#ffff" app:labelVisibilityMode="labeled" app:menu="@menu/bottom_nav_menu" /> </include></pre>
<p>바텀 네비게이션 바 구현</p>	<p>메인화면 구현</p>

■ 스플래시 화면 구현

■ 스플래시 화면

- 앱 실행 시 처음 뜨는 화면. 약 1 ~ 3초정도까지 나타났다 사라진다
- 서비스와 관련된 시각적 요소를 통해 앱에 대한 긍정적인 첫 인상을 심어줄 수 있다



스플래시 화면 이미지 제작
- ppt를 활용하여 제작

```
package com.example.graduation_project;

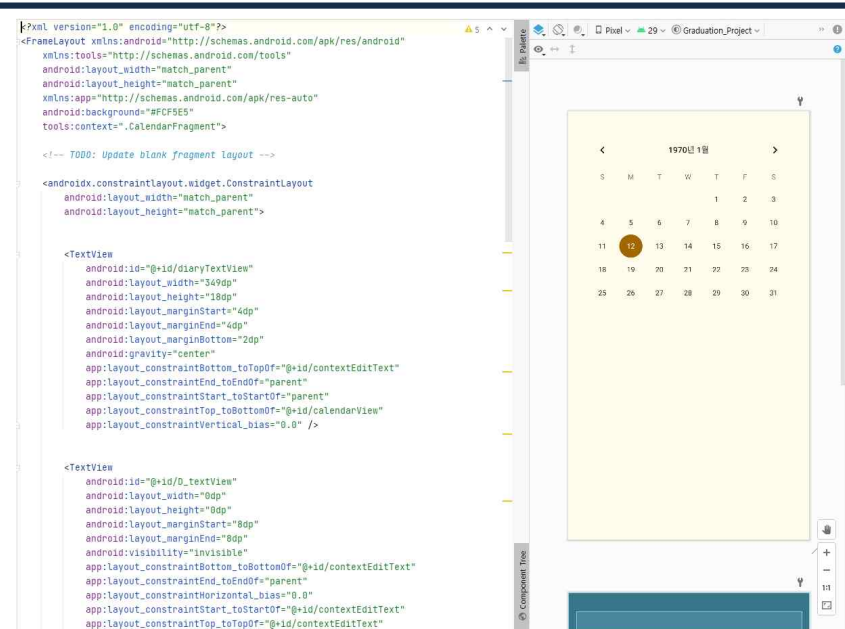
import ...

public class SplashActivity extends Activity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        try{
            Thread.sleep( millis: 3000);
            FirebaseAuth user = FirebaseAuth.getInstance().getCurrentUser();
            if (user != null) {
                Intent intent = new Intent( packageContext: SplashActivity.this, MainActivity2.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
                startActivity(intent);
                SplashActivity.this.finish();
            } else{
                Intent intent = new Intent( packageContext: SplashActivity.this, SignUpActivity.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
                startActivity(intent);
                SplashActivity.this.finish();
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

java 파일을 통한 구현

- 달력 화면 구성
 - Calendar View 활용



xml

```
calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {  
    @Override  
    public void onSelectedDayChange(@NonNull CalendarView view, int year, int month, int dayOfMonth) {  
  
        diaryTextView.setVisibility(View.VISIBLE);  
        save_Btn.setVisibility(View.VISIBLE); // 저장 버튼이 Visible  
        contextEditText.setVisibility(View.VISIBLE); // EditText가 Visible  
        textView.setVisibility(View.INVISIBLE); // 저장된 일기 textView가 Invisible  
        cha_Btn.setVisibility(View.INVISIBLE); // 수정 Button이 Invisible  
        del_Btn.setVisibility(View.INVISIBLE); // 삭제 Button이 Invisible  
        diaryTextView.setText(String.format("%d / %d / %d", year, month+1, dayOfMonth));  
        contextEditText.setText(""); // EditText에 공백값 넣기  
        checkDay(year, month, dayOfMonth);  
  
        if (bundle != null) {  
            t = bundle.getInt( key: "t");  
            if (t == 1 ) {  
  
                int step = bundle.getInt( key: "step");  
                String sStep = Integer.toString(step);  
                Toast.makeText(getApplicationContext(), text: sStep + "일을 걸었습니다.", Toast.LENGTH_LONG).show();  
                save_Btn.setVisibility(View.VISIBLE); // 저장 버튼이 Visible  
                contextEditText.setVisibility(View.VISIBLE); // EditText가 Visible  
                textView.setVisibility(View.INVISIBLE); // 저장된 일기 textView가 Invisible  
                cha_Btn.setVisibility(View.INVISIBLE); // 수정 Button이 Invisible  
                del_Btn.setVisibility(View.INVISIBLE); // 삭제 Button이 Invisible  
                contextEditText.setText("오늘 걸은 걸음 수 : " + sStep);  
  
                calendarView.setDate(System.currentTimeMillis(), animate: false, center: true);  
                t++;  
            }  
        }  
    }  
});  
|  
save_Btn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        saveDiary(f.name);  
        str = contextEditText.getText().toString();  
        textView.setText(str);  
        save_Btn.setVisibility(View.INVISIBLE);  
        cha_Btn.setVisibility(View.VISIBLE);  
        del_Btn.setVisibility(View.VISIBLE);  
    }  
});
```

java

■ 달력 화면 구성

■ Calendar View 활용

```
calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {  
    @Override  
    public void onSelectedDayChange(@NonNull CalendarView view, int year, int month, int dayOfMonth) {  
  
        diaryTextView.setVisibility(View.VISIBLE);  
        save_Btn.setVisibility(View.VISIBLE); // 저장 버튼이 Visible  
        contextEditText.setVisibility(View.VISIBLE); // EditText가 Visible  
        textView.setVisibility(View.INVISIBLE); // 저장된 일기 textView가 Invisible  
        cha_Btn.setVisibility(View.INVISIBLE); // 수정 Button이 Invisible  
        del_Btn.setVisibility(View.INVISIBLE); // 삭제 Button이 Invisible  
        diaryTextView.setText(String.format("%d / %d / %d", year, month+1, dayOfMonth));  
        contextEditText.setText(""); // EditText에 공백값 넣기  
        checkDay(year, month, dayOfMonth);  
  
        if(bundle != null) {  
            t = bundle.getInt( key: "t");  
            if (t == 1 ) {  
  
                int step = bundle.getInt( key: "step");  
                String sStep = Integer.toString(step);  
                Toast.makeText(getActivity().getApplicationContext(), text: sStep + "걸음 걸었습니다.", Toast.LENGTH_LONG).show();  
                save_Btn.setVisibility(View.VISIBLE); // 저장 버튼이 Visible  
                contextEditText.setVisibility(View.VISIBLE); // EditText가 Visible  
                textView.setVisibility(View.INVISIBLE); // 저장된 일기 textView가 Invisible  
                cha_Btn.setVisibility(View.INVISIBLE); // 수정 Button이 Invisible  
                del_Btn.setVisibility(View.INVISIBLE); // 삭제 Button이 Invisible  
                contextEditText.setText("오늘 걸은 걸음 수 : " + sStep);  
  
                calendarView.setDate(System.currentTimeMillis(), animate: false, center: true);  
                t++;  
            }  
        }  
    }  
});
```

- (View.VISIBLE)과 (View.INVISIBLE)을 통해 메모 작성 기능을 날짜를 터치할 시 보이게(활성 되게) 구현

■ 달력 화면 구성

```
@SuppressWarnings("WrongConstant")
public void removeDiary (String readDay){
    FileOutputStream fos = null;

    try {
        fos = getActivity().openFileOutput(readDay, MODE_NO_LOCALIZED_COLLATORS);
        String content = "";
        fos.write((content).getBytes());
        fos.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

@SuppressWarnings("WrongConstant")
public void saveDiary (String readDay){
    FileOutputStream fos = null;

    try {
        fos = getActivity().openFileOutput(readDay, MODE_NO_LOCALIZED_COLLATORS);
        String content = contextEditText.getText().toString();
        fos.write((content).getBytes());
        fos.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

- FileOutputStream을 통해 작성한 메모 파일 생성 및 저장
- fragment에선 기능을 사용할 수 없어 오류가 발생
→ getActivity()을 사용함으로써 해결

- 산책을 완료한 날짜에 해당하는 달력 칸에 발바닥 그림이 찍히게 구현하고 싶었으나 Calendar View의 기능의 한계로 인해 구현 실패
- 커스텀 달력 제작 법을 찾아보았으나 마찬가지로 특정 그림을 달력 칸에 나타내는 기능은 찾지 못하고, 커스텀 달력을 제작할 시 달력의 분량이 불필요하게 커졌기에 기능 포기

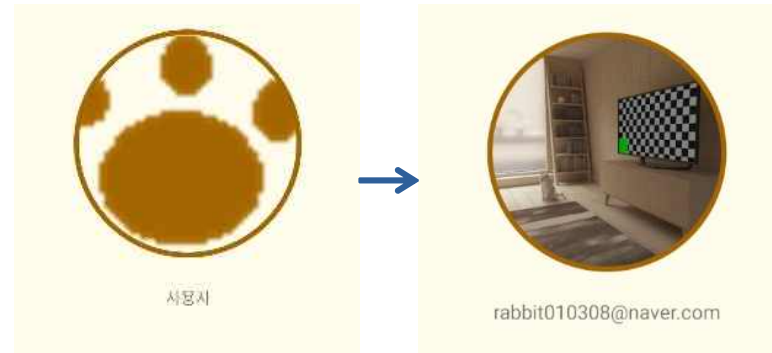
■ 내 정보 화면 구현

- Firebase로 저장되는 회원 정보를 수정하고 로그아웃 등의 기능이 있는 화면 구현

 <pre><?xml version="1.0" encoding="utf-8"?> <FrameLayout 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:layout_width="match_parent" android:layout_height="match_parent" android:background="#FCF5E5" tools:context=".InfoFragment"> <androidx.constraintlayout.widget.ConstraintLayout android:layout_width="match_parent" android:layout_height="match_parent"> <ImageView android:id="@+id/imageInfoTitle" android:layout_width="41dp" android:layout_height="54dp" android:src="@drawable/info_title" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="1.0" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" android:background="#F60FAB" app:layout_constraintVertical_bias="0.0" /> <de.hdodenhof.circleimageview.CircleImageView android:id="@+id/imageProfile" android:layout_width="180dp" android:layout_height="180dp" android:src="@drawable/icon" app:civ_border_color="#9E6401" app:civ_border_width="4dp" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toBottomOf="@+id/imageInfoTitle" app:layout_constraintVertical_bias="0.101" /> <TextView android:id="@+id/userId" android:layout_width="150dp" xml</pre>	<pre>package com.example.graduation_project; import ... public class InfoFragment extends Fragment{ ViewGroup viewGroup; ImageView imageProfile; TextView userName; private static final String TAG = "MainActivity"; //logout @Nullable @Override public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) { viewGroup = (ViewGroup) inflater.inflate(R.layout.fragment_info,container, attachToRoot: false); imageProfile = viewGroup.findViewById(R.id.imageProfile); imageProfile.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { Intent intent = new Intent(Intent.ACTION_GET_CONTENT); intent.setType("image/*"); startActivityForResult(intent, requestCode: 101); } }); viewGroup.findViewById(R.id.logout).setOnClickListener(onClickListener); viewGroup.findViewById(R.id.info_edit).setOnClickListener(onClickListener); //set user name & profile photo FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser(); if (user != null) { String name = user.getEmail(); userName = viewGroup.findViewById(R.id.userName); userName.setText(name); Uri uri = user.getPhotoUrl(); imageProfile.setImageURI(uri); } //set user name java</pre>
---	---

■ 내 정보 화면 구현

```
//set user name & profile photo
FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
if (user != null) {
    String name = user.getEmail();
    userName = viewGroup.findViewById(R.id.userId);
    userName.setText(name);
    Uri uri = user.getPhotoUrl();
    imageProfile.setImageURI(uri);
}
//set user name
```



- Firebase에서 사용자의 정보를 받아와 프로필 사진과 사용자 계정을 표시

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == 101) {
        if(resultCode == RESULT_OK) {
            try {
                InputStream inputStream = getActivity().getContentResolver().openInputStream(data.getData());
                Bitmap image = BitmapFactory.decodeStream(inputStream);

                imageProfile.setImageBitmap(image);

                //firebase profile
                FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
                //Context context = getContext();

                //Uri u = getImageUri(context, image);

                //uri-----

                ByteArrayOutputStream bytes = new ByteArrayOutputStream();
                image.compress(Bitmap.CompressFormat.JPEG, 100, bytes);
                String path = MediaStore.Images.Media.insertImage(getActivity().getApplicationContext().getContentResolver(), image, "title", "description");
                Uri u = Uri.parse(path);

                //uri-----

                UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()
                    .setPhotoUri(u)
                    .build();

                user.updateProfile(profileUpdates)
                    .addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if (task.isSuccessful()) {
                                Toast.makeText(getActivity().getApplicationContext(), "이미지 변경", Toast.LENGTH_LONG).show();
                            }
                        }
                    });
            } catch (Exception e) {
                e.printStackTrace();
                Toast.makeText(getActivity().getApplicationContext(), "이미지 로드 오류"+e, Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

- 프로필 사진 터치 시 갤러리로 이동해서 원하는 사진으로 프로필 사진을 등록하는 기능 구현
- Bitmap의 이미지를 Uri로 바꾸어서 Firebase의 PhotoUri에 저장

■ 내 정보 화면 구현

```
//logout
View.OnClickListener onClickListener = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.logout:
                FirebaseAuth.getInstance().signOut();
                myStartActivity(SignUpActivity.class);
                break;

            case R.id.info_edit:
                myStartActivity(MemberInitActivity.class);
                break;
        }
    }
};
```

- switch를 통해 로그아웃과 정보수정 페이지로 이동하는 onClickListener 구현
- 정보 수정은 Firebase 담당하신 조원 분이 구현한 Activity로 이동하게 구현

- 정보 수정 페이지에서 정보 수정만 가능하고 기존 입력한 정보를 조회할 수 없었기에 정보 수정 페이지로 이동하면 기존에 입력한 정보가 보여지게끔 코딩하려 하였으나 Firebase에 등록된 정보를 가져오는 방식이 대부분 수정 후 반영된 결과를 가져오는 방식이었기에 아직 수정하지 않은 상태에서 기존의 정보를 가져오는 방식을 찾기 어려웠다. 기존의 정보를 불러오는 코딩을 몇 가지 방법으로 해 보았으나 Firebase 오류로 인해 결국 구현하지 못하였다.

🐾 회원 정보 🐾

이름

전화번호

생년월일

주소

확인

깃허브를 통한 협업

Commits on Nov 9, 2021		
member_init edit yelim0308 committed 21 days ago	0155e4b	<>
Merge remote-tracking branch 'origin/master' yelim0308 committed 21 days ago	33675d9	<>
yelim0308 committed 21 days ago	2a31d54	<>
Merge remote-tracking branch 'origin/master' kuchic0918 committed 21 days ago	de1d884	<>
1109 12:44 kim kuchic0918 committed 21 days ago	e840e07	<>
Commits on Nov 8, 2021		
label&icon edit yelim0308 committed 21 days ago	2241fec	<>
Merge remote-tracking branch 'origin/master' yelim0308 committed 21 days ago	e7567b7	<>
yelim0308 committed 21 days ago	146e908	<>
rr hyun29570 committed 21 days ago	8aba679	<>
Commits on Nov 4, 2021		
11/02 23:07 KKS kuchic0918 committed 25 days ago	0f3a64c	<>
Commits on Nov 2, 2021		
11/02 23:07 KKS kuchic0918 committed 27 days ago	47c6f19	<>
Merge remote-tracking branch 'origin/master' kuchic0918 committed 27 days ago	be6912a	<>

- 각자 맡은 부분을 코딩을 한 뒤
commit & push를 통해 깃허브에
업로드하려 각자 Android Studio의
프로젝트에 팀원들이 push한 내용을
update하여 합치면서 진행
- 팀원들이 코딩한 xml파일의 레이아웃
및 디자인을 수정함

기능 추가 및 수정

■ 지도에 만보계 기능 추가

```
//-----
SensorManager sensorManager;
Sensor stepCountSensor;
TextView stepCountView;
int currentSteps = 0;
//-----
CalendarFragment fragment;

public MapFragment() {}

public static MapFragment newInstance()
{
    MapFragment fragment = new MapFragment();
    return fragment;
}

@Override
public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); }

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState)
{
    // Inflate the layout for this fragment
    ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment_map,
        container, attachToRoot: false);

    mapView = (MapView) rootView.findViewById(R.id.navermap);
    mapView.onCreate(savedInstanceState);
    mapView.getMapAsync( callback: this);
    mLocationSource = new FusedLocationSource( fragment: this, PERMISSION_REQUEST_CODE);

    chronometer_t = rootView.findViewById(R.id.et_placeName);
    chronometer_t.setFormat("시간: %s");
    //-----만보계

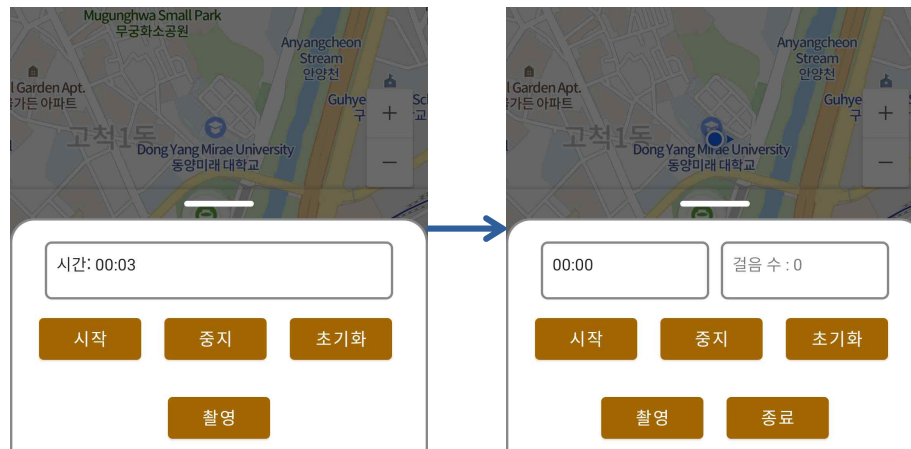
    stepCountView = rootView.findViewById(R.id.et_placeStep);
    stepCountView.setText("걸음 수 : "+currentSteps);

    sensorManager = (SensorManager) getActivity().getSystemService(Context.SENSOR_SERVICE);
    stepCountSensor = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_DETECTOR);

    //-----
```

- 센서를 통해 만보계 기능 구현
- 조원이 구현해 놓았던 타이머의 시작&종료 버튼과 같이 작동하도록 구현
- 거리측정 기능을 넣으려 했으나 지도 기능을 구현하는데 사용된 NaverMap에서는 거리를 측정할 수 있는 기능이 없었기에 대신 만보계 기능을 추가했다.

■ 지도에 종료 버튼 추가



- 만보계 기능을 추가함과 함께 종료 버튼 추가
- 버튼을 누를 시 달력화면으로 이동하면서 걸음 수에 저장된 int 값을 bundle을 통해 달력 fragment로 전송한다

```
//-----종료 버튼
fragment = new CalendarFragment();

endBtn = rootView.findViewById(R.id.sendCButton);
endBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        bundle.putInt("step", currentSteps);
        bundle.putInt("t", 1);
        fragment.setArguments(bundle);
        /*
        Intent intent = new Intent();
        intent.putExtra("steps", currentSteps);
        getActivity().setResult(RESULT_OK, intent);
        getActivity().finish();
        */
        getActivity().getSupportFragmentManager().beginTransaction().replace(R.id.main_layout, fragment).commitAllowingStateLoss();
    }
});
//-----
```

■ 종료버튼으로 인한 달력 fragment에 기능 추가

```
Bundle bundle = this.getArguments();

//-----get step

if(bundle != null){
    int step = bundle.getInt( key: "step");
    String sStep = Integer.toString(step);
    Toast.makeText(getActivity().getApplicationContext(), text: sStep + "걸음 걸었습니다.", Toast.LENGTH_SHORT).show();

    calendarView.setDate(System.currentTimeMillis(), animate: false, center: true);

    textView.setVisibility(View.VISIBLE); // 저장된 알기 textView가 visible
    textView.setText("-----산책 종료!----- \n *오늘 걸은 걸음 수 : " + sStep); // EditText에 공백값 넣기
}

//-----
```

- 받은 bundle이 있을 시 bundle 값인 걸을 수가 토스트 창으로 표시된다
- 텍스트창에 산책 종료 문구가 표시된다



최종 발표

■ 발표 ppt 제작

🐾 작품 개요

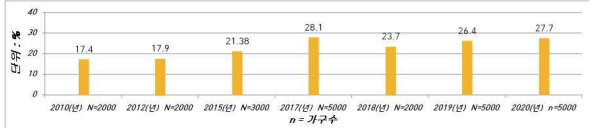
Mo Mo 모두 모이개

- 개요

강아지들에 대한 지식과 정보를 공유 하고 반려동물의 산책 요일과 산책 일정을 보다 편하게 관리하게 할 수 있게 지도를 통해 자신과 반려동물의 위치를 확인하고 시간을 기록할 수 있게 해주는 앱 입니다.

- 배경 및 필요성

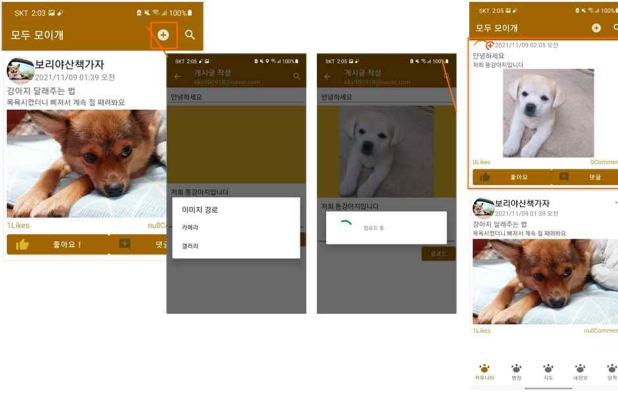
요즘 반려동물을 키우는 사람이 증가하고 있습니다. 반려동물 중 특히 많이 키우는 강아지는 산책시키는 것이 매우 중요한 동물입니다. 그래서 강아지를 산책 시킬 때 유용한 기능들을 넣은 어플리케이션을 만들면 좋겠다는 생각을 하여 이 어플리케이션을 개발하게 되었습니다.



연도	개수 (백만 명)
2010(년) N=2000	17.4
2012(년) N=2000	17.9
2015(년) N=3000	21.38
2017(년) N=5000	28.1
2018(년) N=2000	23.7
2019(년) N=5000	26.4
2020(년) n=5000	27.7

n = 개구수

🐾 화면 설계 [4/8]



■ 제작한 어플리케이션 시연

- 팀원들의 휴대기기가 ios였기에 android를 사용하는 본인의 휴대기기로 실제 시연

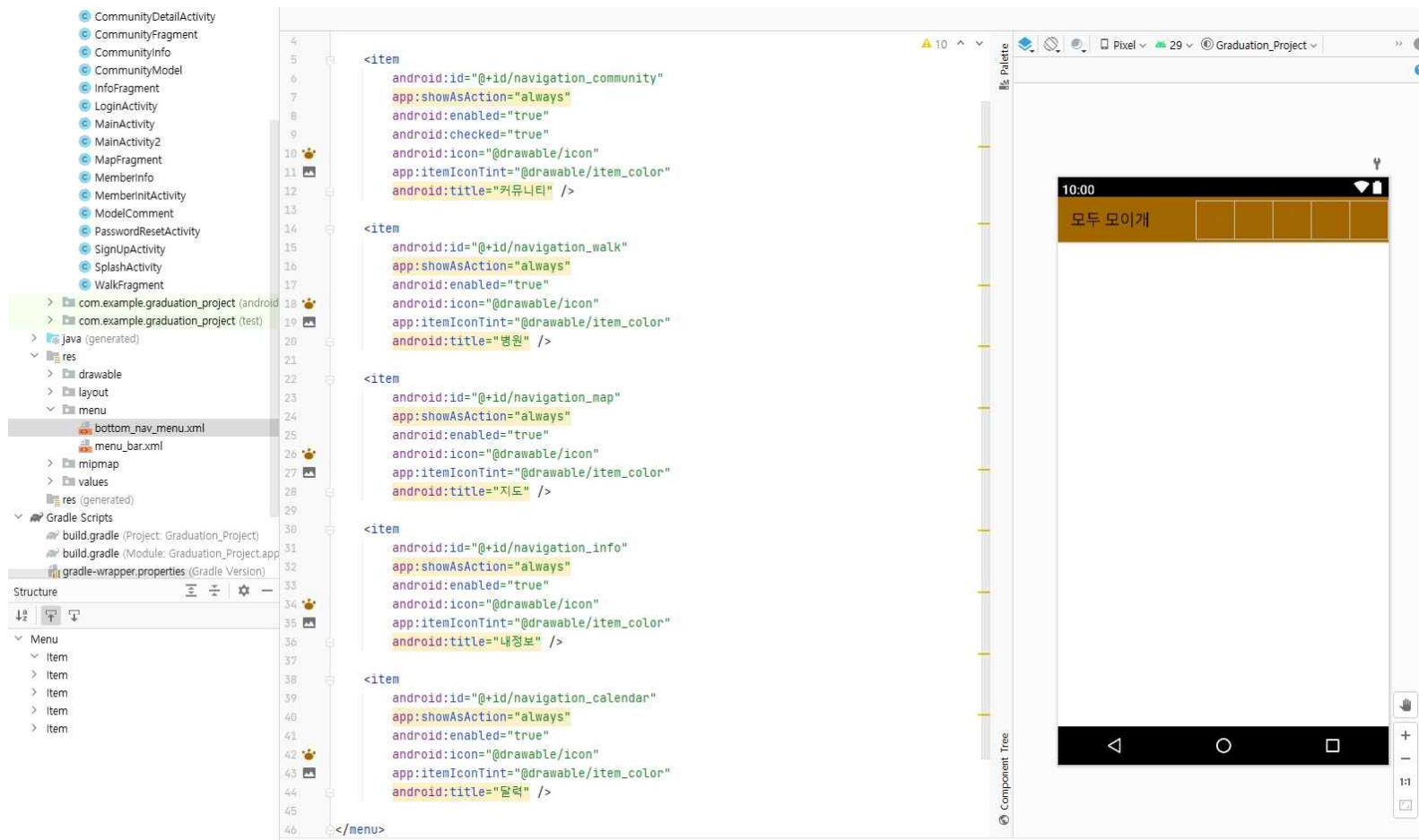


Android Studio

- navigation bar
- fragment
- sensor
- firebase

navigation bar

- 사용할 화면이 많기에 모든 화면들에 쉽게 접근할 수 있는 방법을 모색하다 사용하게 된 기능
- Activity를 이동하는 것보다 가볍고 편리하게 화면을 이동할 수 있다는 장점이 있다




```

bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        switch(item.getItemId()){
            case R.id.navigation_community : {
                getSupportFragmentManager().beginTransaction().replace(R.id.main_layout,fragment1).commitAllowingStateLoss();
                return true;
            }

            case R.id.navigation_walk : {
                getSupportFragmentManager().beginTransaction().replace(R.id.main_layout,fragment2).commitAllowingStateLoss();
                return true;
            }

            case R.id.navigation_map : {
                getSupportFragmentManager().beginTransaction().replace(R.id.main_layout,fragment3).commitAllowingStateLoss();
                return true;
            }

            case R.id.navigation_info : {
                getSupportFragmentManager().beginTransaction().replace(R.id.main_layout,fragment4).commitAllowingStateLoss();
                return true;
            }

            case R.id.navigation_calendar : {
                getSupportFragmentManager().beginTransaction().replace(R.id.main_layout,fragment5).commitAllowingStateLoss();
                return true;
            }
            default: return false;
        }
    }
});
}

```

- java 파일에서 OnNavigationItemSelectedListener를 통해 fragment로 이동할 수 있게 한다

fragment

- 기존에는 Activity를 주로 사용했기에 fragment에 대한 지식이 부족했는데, navigation bar를 사용하게 되면서 fragment를 주로 사용하게 되었다
- Activity에서 사용 가능했던 기능이 사용 불가능하거나, 오류가 뜨는 경우가 많았기에 처음에는 어려움이 많았으나 사용하면서 익숙해지고 다양한 오류를 수정해나가면서 fragment에 대해 많이 배우게 되었다

- Activity에서 사용 가능했지만 Fragment에서 사용 불가능한 대부분의 메소드는 대부분 앞에 getActivity().를 사용하면 해결되었다

```
Toast.makeText(getActivity().getApplicationContext(), text: sStep + "걸음 걸었습니다.", Toast.LENGTH_LONG).show();
```

```
fis = getActivity().openFileInput(f_name);
```

- findViewById를 사용하기 위해서는 앞에 viewGroup.를 사용해야 한다

```
public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container,
    @Nullable Bundle savedInstanceState) {
    viewGroup = (ViewGroup) inflater.inflate(R.layout.fragment_calendar, container, attachToRoot: false);

    calendarView = (CalendarView)viewGroup.findViewById(R.id.calendarView);
    contextEditText = viewGroup.findViewById(R.id.contextEditText);
    diaryTextView = viewGroup.findViewById(R.id.diaryTextView);
    textView = viewGroup.findViewById(R.id.D_textView);
    cha_Btn = viewGroup.findViewById(R.id.cha_Btn);
    del_Btn = viewGroup.findViewById(R.id.del_Btn);
    save_Btn = viewGroup.findViewById(R.id.save_Btn);
}
```

sensor

- 만보계 기능을 구현하며 센서 활용하는 방법에 대해 알게되었다

```
//-----  
SensorManager sensorManager;  
Sensor stepCountSensor;  
TextView stepCountView;  
int currentSteps = 0;  
//-----  
  
//-----만보계-----  
  
stepCountView = rootView.findViewById(R.id.et_placeStep);  
stepCountView.setText("걸음 수 : "+currentSteps);  
  
sensorManager = (SensorManager) getActivity().getSystemService(Context.SENSOR_SERVICE);  
stepCountSensor = sensorManager.getDefaultSensor(Sensor.TYPE_STEP_DETECTOR);  
  
//-----
```

- 센서 이벤트 발생 시 걸음 수가 증가하는 방식으로 코딩을 하였는데 직접 센서 값의 변화에 따라 걸음 수를 조절할 수 있게 민감도를 조절하는 방법도 있었다

```
//-----만보계-----  
public void s_onStart() {  
    super.onStart();  
    if (stepCountSensor == null) {  
        Toast.makeText(getActivity().getApplicationContext(), text: "No Step Sensor", Toast.LENGTH_SHORT).show();  
    }  
    else if(stepCountSensor !=null) {  
        // 센서 속도 설정  
        // * 옵션  
        // - SENSOR_DELAY_NORMAL: 20,000 초 딜레이  
        // - SENSOR_DELAY_UI: 0,000 초 딜레이  
        // - SENSOR_DELAY_GAME: 20,000 초 딜레이  
        // - SENSOR_DELAY_FASTEST: 딜레이 없음  
        //  
        sensorManager.registerListener( listener: this,stepCountSensor,SensorManager.SENSOR_DELAY_FASTEST);  
    }  
}  
  
@Override  
public void onSensorChanged(SensorEvent event) {  
    // 걸음 센서 이벤트 발생시  
    if(event.sensor.getType() == Sensor.TYPE_STEP_DETECTOR && running == true){  
        switch(event.sensor.getType()){  
            // 센서 이벤트가 발생할때 마다 걸음수 증가  
            case Sensor.TYPE_STEP_DETECTOR :  
                currentSteps++;  
                stepCountView.setText("걸음 수 : "+currentSteps);  
            }  
        }  
    }  
}
```

Firebase

- 이번 프로젝트를 진행하며 처음 접해본 데이터베이스이다
- DB담당은 다른 조원분이 담당했기에 자세한 내용은 모르지만 내정보(InfoFragment)를 구현하며 어느정도 연동하는 방법을 알게 되었다

```
//set user name & profile photo
FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
if (user != null) {
    String name = user.getEmail();
    userName = viewGroup.findViewById(R.id.userId);
    userName.setText(name);
    Uri uri = user.getPhotoUrl();
    imageProfile.setImageURI(uri);
}
```

- firebase에서 사용자 정보 가져오기

```
UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()
    .setPhotoUri(u)
    .build();

user.updateProfile(profileUpdates)
    .addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            if (task.isSuccessful()) {
                Toast.makeText(getActivity().getApplicationContext(), text: "이미지 변경", Toast.LENGTH_LONG).show();}}});
```

- firebase에 정보 업로드 하기

후기

- 구현한 프로젝트가 처음 작성한 기획안의 내용과는 많이 달라져서 아쉽다
- 코딩을 하며 배웠던 내용을 활용할 수 있어 좋았지만 프로젝트를 구현하기 위해서는 배웠던 내용보다는 배운 적 없는 것들을 많이 구현해야했기에 어려움이 컸다. 구글링에도 한계가 있었기에 구현하지 못한 기능이 많은게 제일 아쉽다. 그래도 구글링을 하며 다양한 코딩관련 웹 사이트를 알게되어 좋았다. 특히 깃허브는 다른 사람과 협업해볼 기회가 없었는데 조원들과 협업해보면서 깃허브의 기능을 사용해 볼 수 있다는 것이 좋은 기회였던 것 같다
- 5인 1조임에도 불구하고 빠진 팀원이 두 명이나 되어서 다양한 부분에서 디테일한 부분을 신경쓰지 못하고 결국 완성도가 떨어진 것이 아쉽다.
그래도 남은 인원들이 열심히 해 결과물을 완성했다는 것이 뿌듯하다
특히 내가 기획했던 것이 실제로 구현된 것을 보니 신기했다
- 어플리케이션을 개발할 때 신경써야 하는 점이 아주 많다는 것을 알게 되었다.
간단해 보이는 기능도 많은 노력을 필요로 하고 한군데라도 오류가 나면 프로젝트 진행이 되지 않기에 시간과 노력이 많이 필요하다는 것을 느꼈다
- 나의 안드로이드 스튜디오 코딩 실력이 좋지 않다고 생각하여 걱정이 앞섰는데, 직접 부딪히며 일단 코딩해보고 오류를 고쳐가며 기능을 하나씩 구현해가면서 조금은 성장한 것 같다. 좋은 경험인 것 같다.



감사합니다