

안드로이드 프로그래밍 - Term Project -

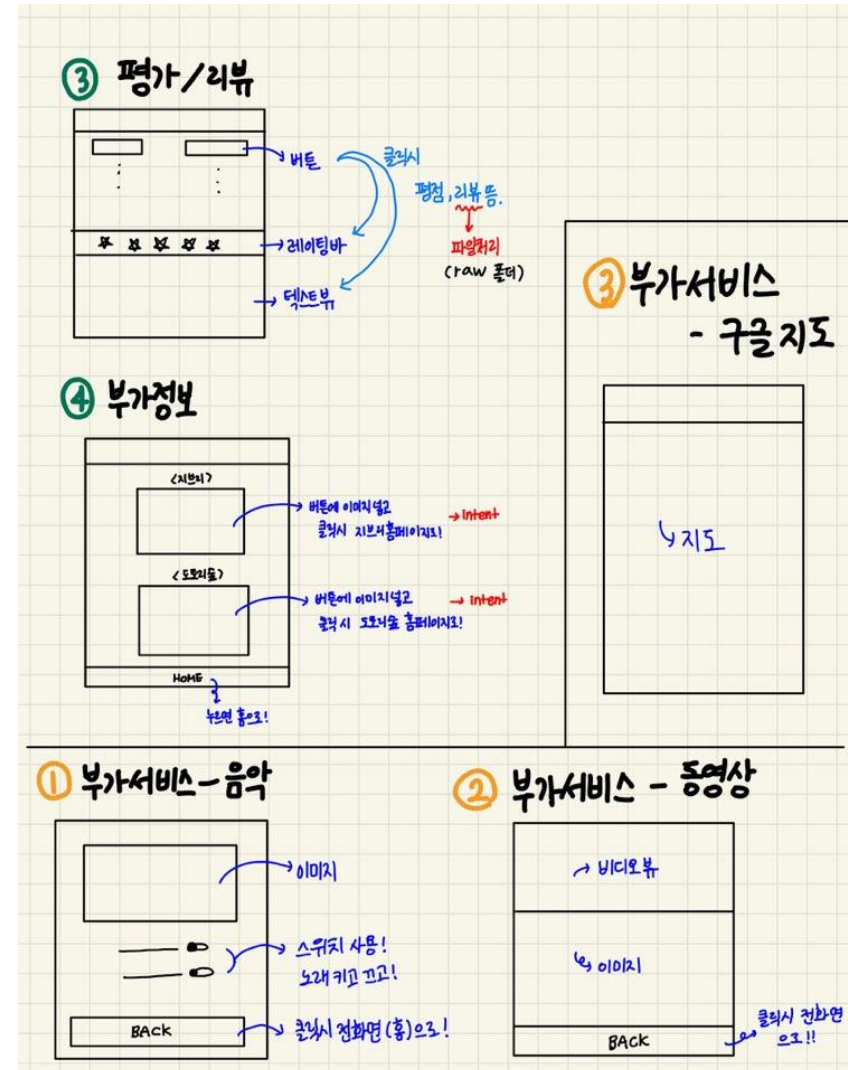
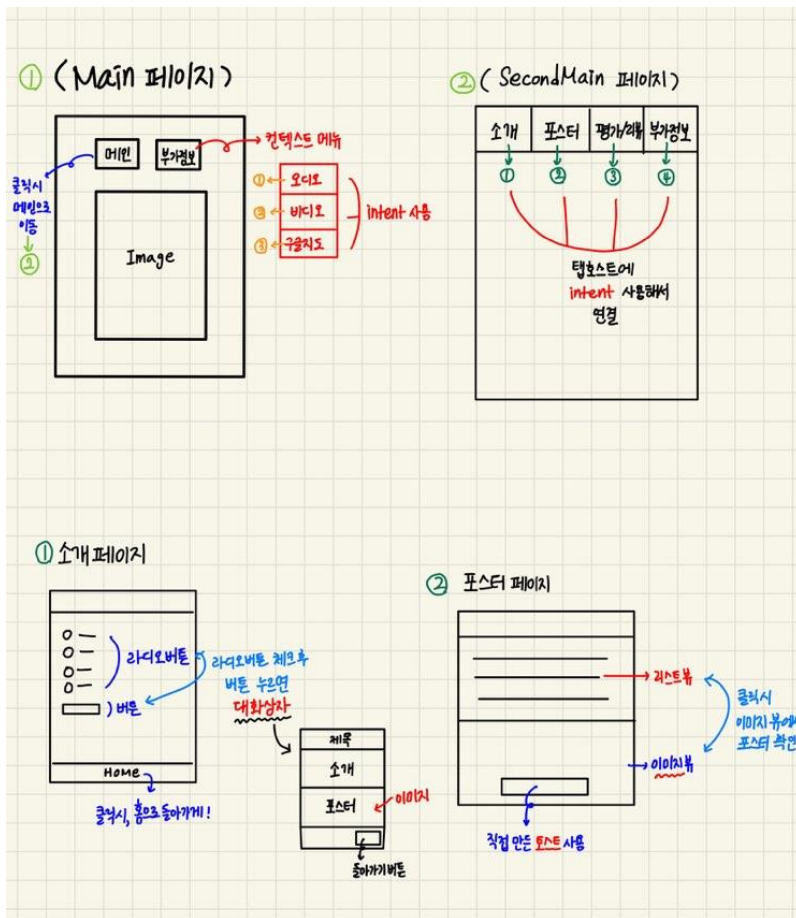
201858103 컴퓨터공학부 김지윤

2020년 2학기 프로젝트

Term Project

2

□ 구상도

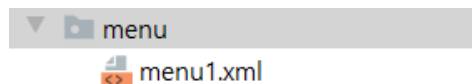


Term Project

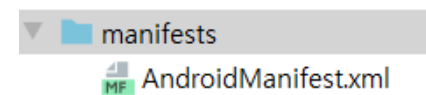
3

□ 프로젝트 파일

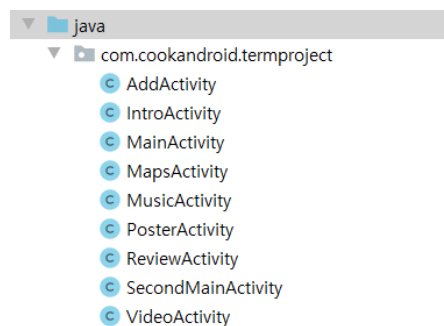
(menu)



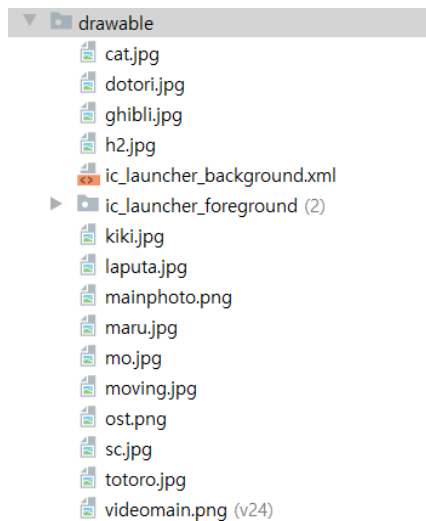
(manifest)



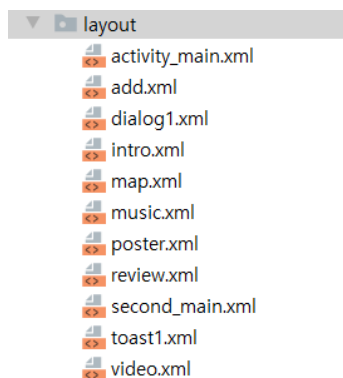
(java)



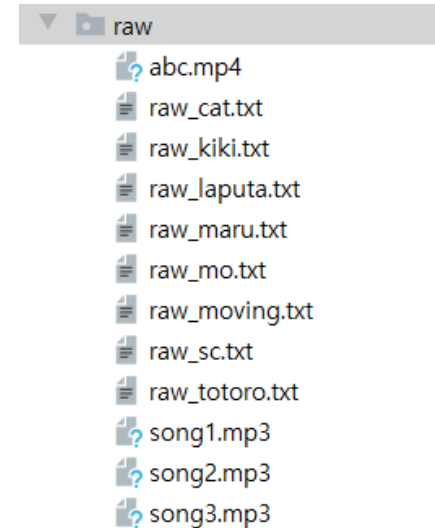
(drawable)



(layout)



(raw)



Term Project

(MainActivity.java / activity_main.xml / menu1.xml)

4

□ 첫 화면 (메인 화면)



(부가 서비스! 길게 클릭 시)

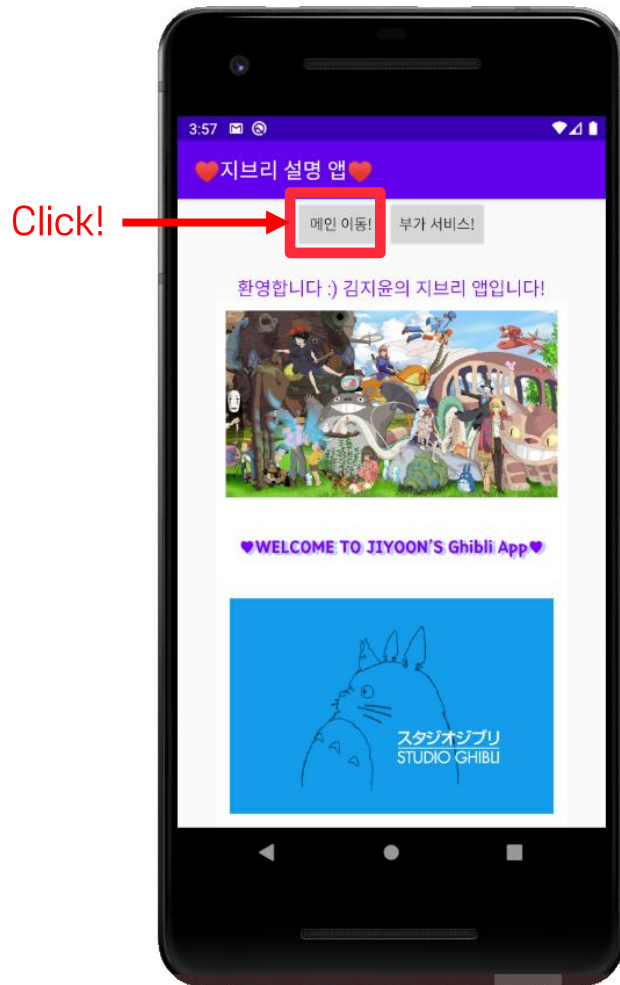


Term Project

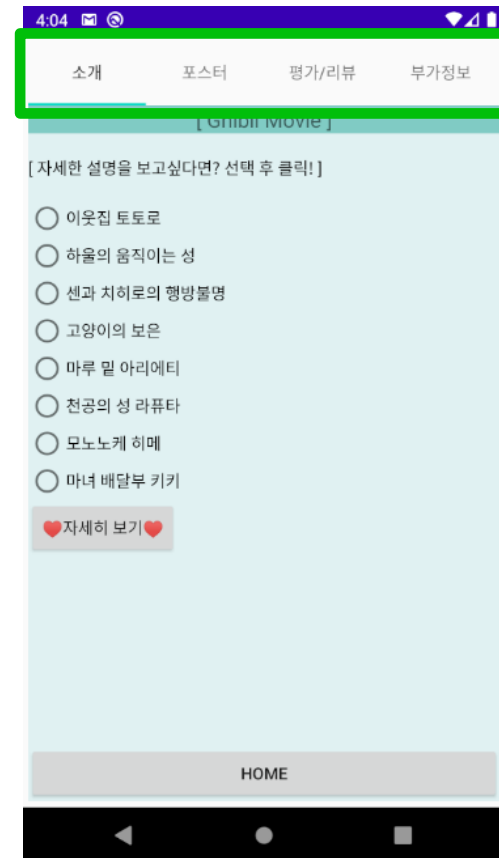
(SecondMainActivity.java / second_main.xml)

5

□ 첫 화면 (메인 화면)



(메인 이동! 클릭 시)



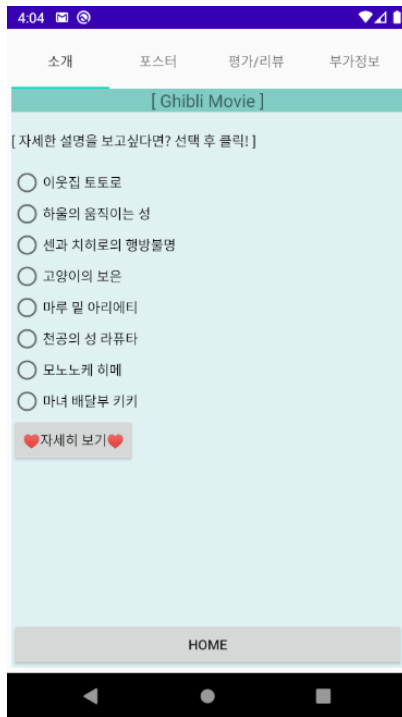
← 탭 호스트 사용

Term Project

6

□ ‘메인 이동!’ 클릭 후 -> 두번째 메인 화면

(소개)



(포스터)



(평가/리뷰)



(부가 정보)



(intro.xml / IntroActivity.java) (poster.xml / Poster.java) (review.xml / ReviewActivity.java) (add.xml / AddActivity.java)
(+ dialog1.xml) (+ toast.xml)

Term Project

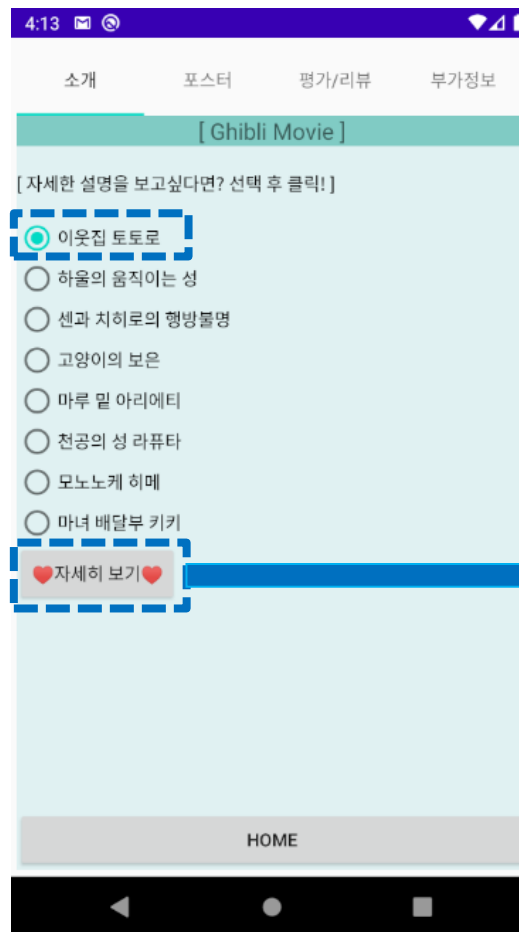
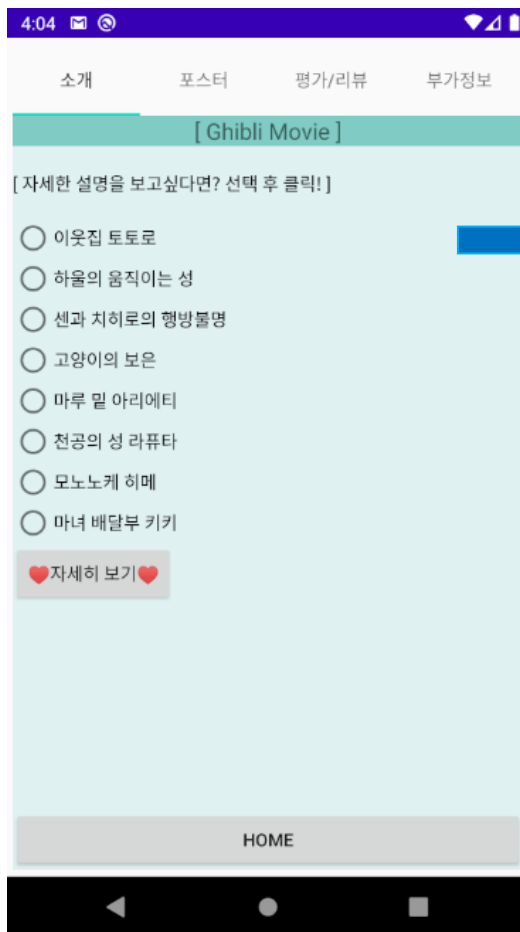
(intro.xml / IntroActivity.java / dialog1.xml)

7

□ 두번째 메인 화면 - 소개

라디오 버튼 체크 후 '자세히 보기' 클릭!

설정된 대화 상자가 나옴!

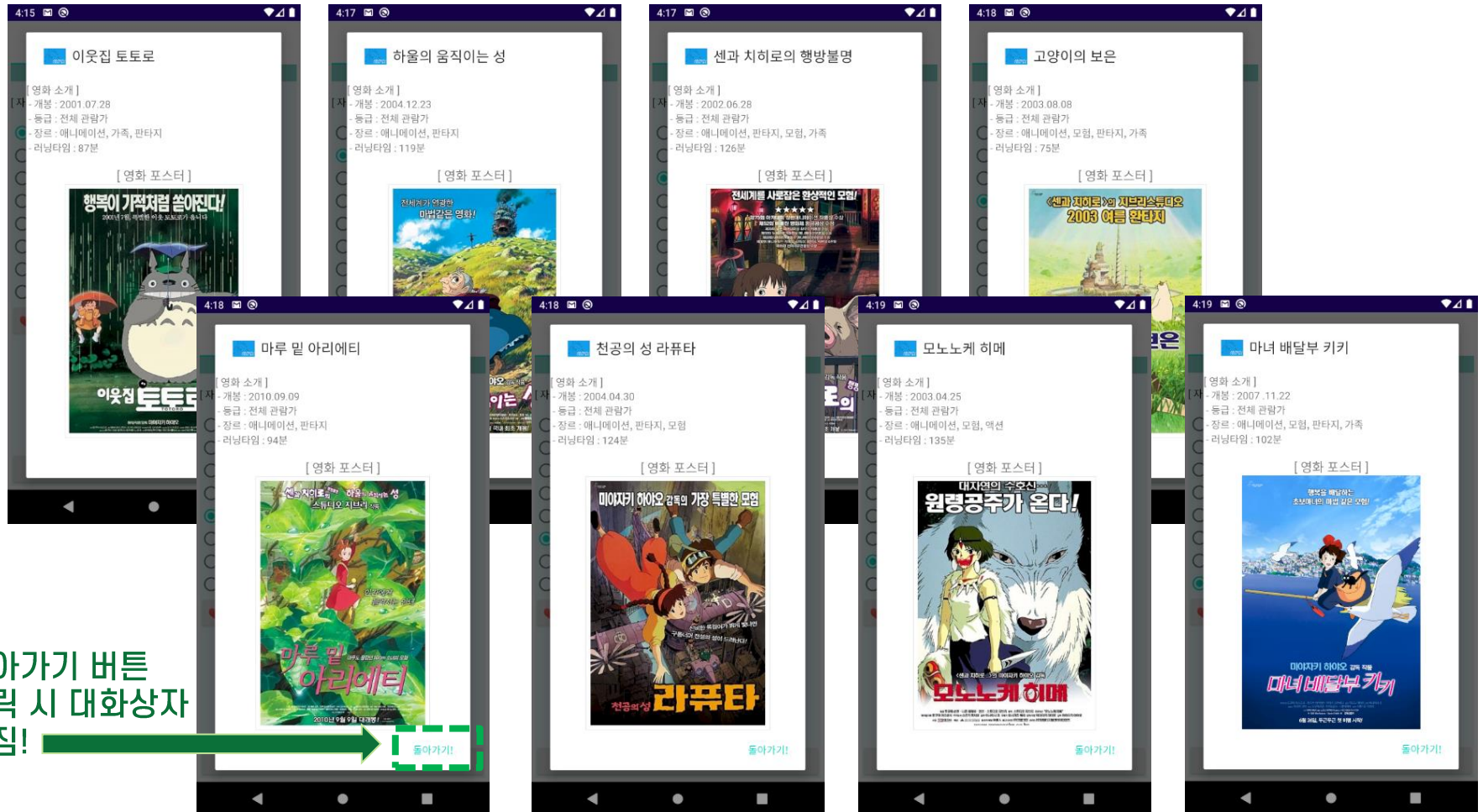


Term Project

8

□ 두번째 메인 화면 - 소개

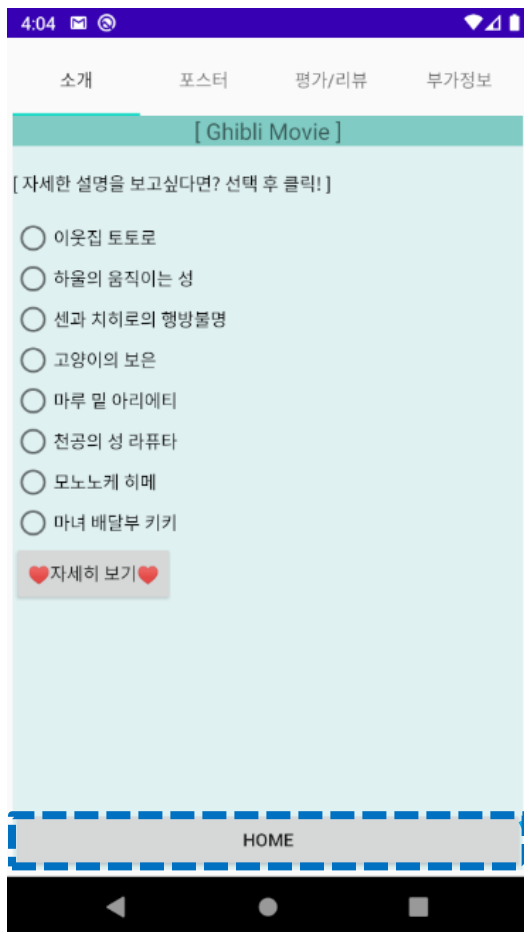
해당 라디오 버튼 + '자세히 보기' 클릭 시 보이는 대화상자들



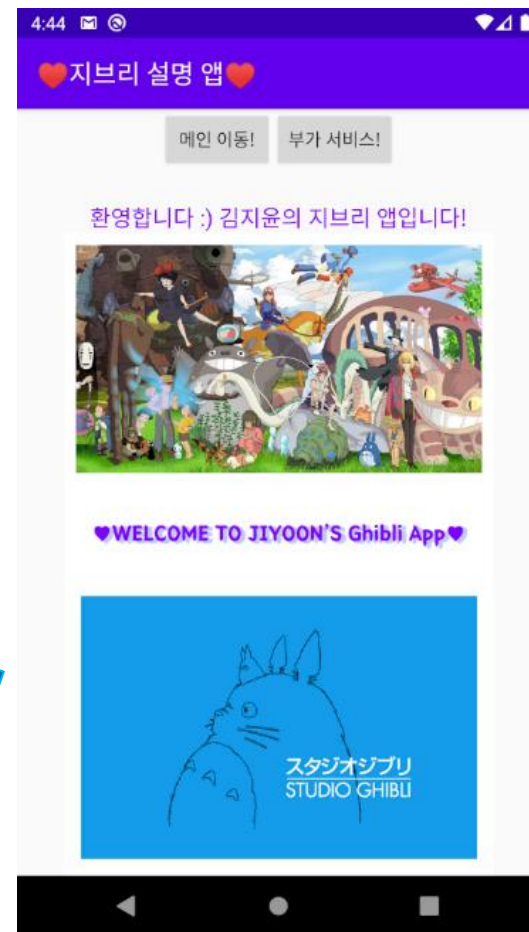
Term Project

9

□ 두번째 메인 화면 - 소개



홈 화면(메인 화면) 나옴!

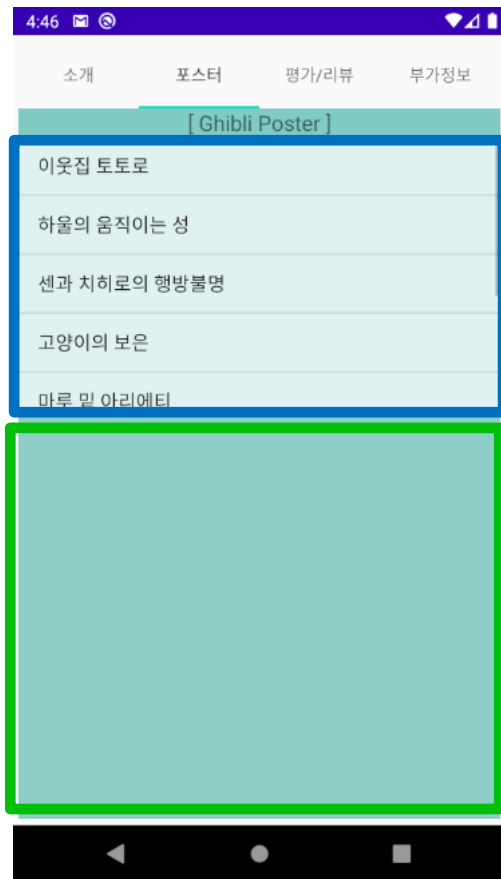


Term Project

(poster.xml / Poster.java / toast.xml)

10

□ 두번째 메인 화면 - 포스터



리스트 뷰에서 해당 값 클릭 시 밑의
이미지 뷰에서 포스터 사진 보임!

Click ! →

리스트 뷰 사용!

포스터 ! →

이미지 뷰 사용!

직접 만든 토스트 사용 ! →

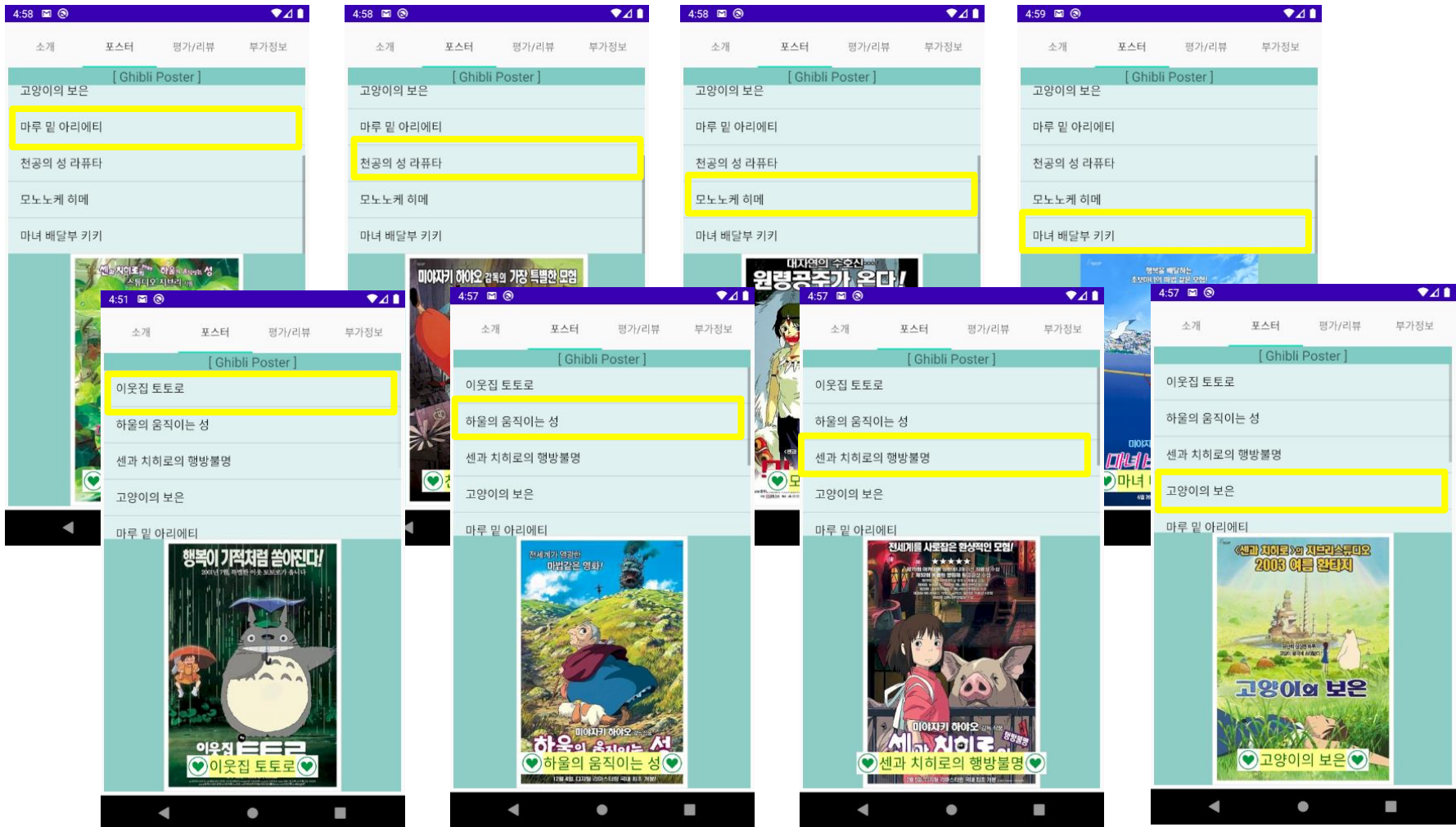


Term Project

11

□ 두번째 메인 화면 - 포스터

리스트 뷰에서 해당 값 클릭 시 밑의
이미지 뷰에서 포스터 사진 보임!



Term Project

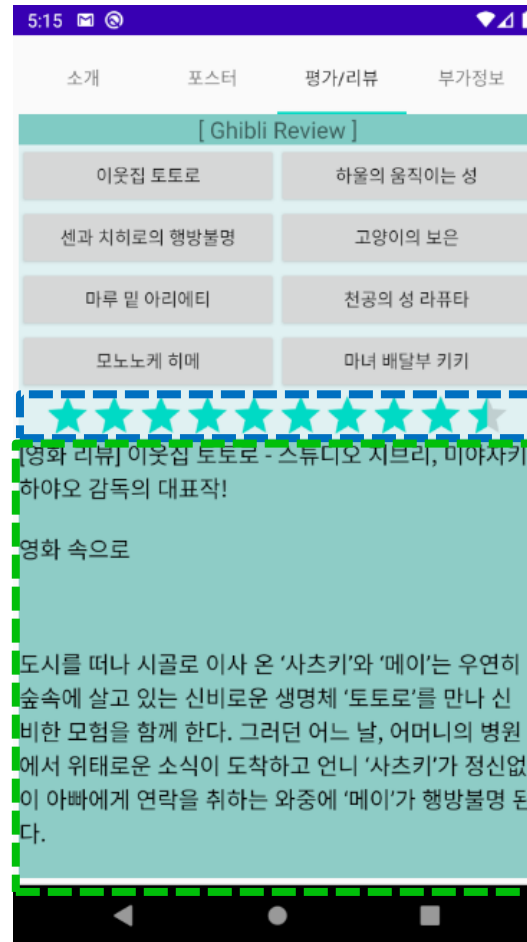
(review.xml / ReviewActivity.java)

12

□ 세번째 메인 화면 – 평가/리뷰

버튼 클릭 시, 해당 영화의 평점과 리뷰를 확인할 수 있다.

버튼 클릭!



레이팅 바 사용

스크롤 뷰/텍스트 뷰 사용
파일 처리 (raw폴더 이용)

Term Project

13

□ 세번째 메인 화면 – 평가/리뷰

버튼 클릭 시, 해당 영화의 평점과 리뷰를 확인할 수 있다.



Term Project

(add.xml / AddActivity.java)

14

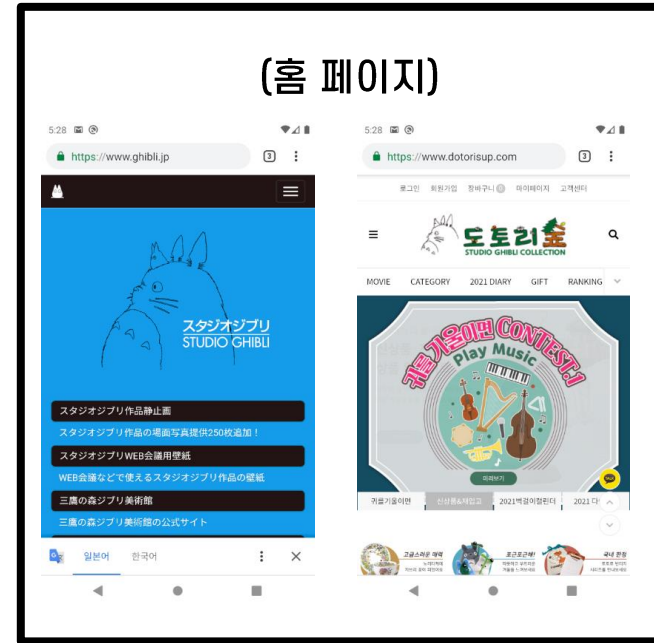
□ 네번째 메인 화면 – 부가 정보



버튼 클릭 시,
지브리 공식 홈페이지로 이동

버튼 클릭 시,
도토리숲 공식 홈페이지로 이동

버튼 클릭 시,
홈 화면(메인 페이지) 으로 이동



Term Project

15

□ 네번째 메인 화면 – 부가 정보



(버튼 클릭 시, 토스트 확인 가능)

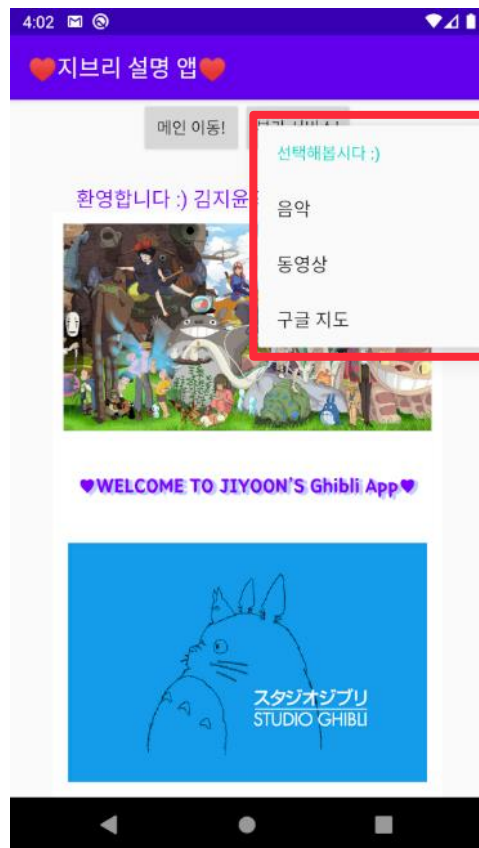


Term Project

(MusicActivity.java / music.xml)

16

□ 부가 서비스 (컨텍스트 메뉴)



- (1) 음악
- (2) 동영상
- (3) 구글 지도

Switch를 이용하여
음악을 켜고 끌 수 있다!

BACK 버튼 클릭 시,
홈 (메인) 화면으로 돌아간다!

(1) 음악



Term Project

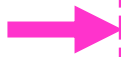
(VideoActivity.java / video.xml)

17

□ 부가 서비스 (컨텍스트 메뉴)

(2) 동영상

비디오 뷰 사용!



BACK 버튼 클릭 시,
홈(메인) 화면으로
이동!

Term Project

(MapsActivity.java / map.xml)

18

□ 부가 서비스 (컨텍스트 메뉴)

(3) 구글 지도



구글 지도 클릭 시 일반 지도로 볼 수 있도록 설정하였습니다!

마커 위치 : 지브리 샵 (도토리 숲)

(+) 클릭 시



(+)와 (-)를 사용하여, 지도에서 멀고, 가깝게 볼 수 있습니다 ☺

Term Project : Code

19

[AndroidManifest.xml]

```
<!-- 메인 Activity -->
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

```
<!-- 탭호스트 Activity -->
<activity
    android:name=".SecondMainActivity"
    tools:ignore="WrongManifestParent"></activity>
```

```
<!-- 소개 Activity -->
<activity
    android:name=".IntroActivity"
    tools:ignore="WrongManifestParent"></activity>
```

```
<!-- 포스터 Activity -->
<activity
    android:name=".PosterActivity"
    tools:ignore="WrongManifestParent"></activity>
```

```
<!-- 평가/리뷰 Activity -->
<activity
    android:name=".ReviewActivity"
    tools:ignore="WrongManifestParent"></activity>
```

```
<!-- 부가정보 Activity -->
<activity
    android:name=".AddActivity"
    tools:ignore="WrongManifestParent"></activity>
```

```
<!-- 오디오 Activity -->
<activity
    android:name=".MusicActivity"
    tools:ignore="WrongManifestParent"></activity>
```

```
<!-- 비디오 Activity -->
<activity
    android:name=".VideoActivity"
    tools:ignore="WrongManifestParent"></activity>
```

```
<!-- 구글 지도 Activity -->
<activity
    android:name=".MapsActivity"
    android:label="Map"></activity>
```

- 각 액티비티들을 AndroidManifest.xml 에 기재 해주어야 한다. (하지 않으면 실행 자체가 되지 않는다.)

Term Project : Code

20

[MainActivity.java]

- 컨텍스트 메뉴
(menu.xml에서 만든 형태의 메뉴를 컨텍스트 메뉴로 사용)

```
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);

    MenuInflater mInflater = getMenuInflater();
    if (v == btnStart) {
        menu.setHeaderTitle("선택해봅시다 :) ");

        mInflater.inflate(R.menu.menu1, menu);
    }
}
```

[menu.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:tools="http://schemas.android.com/tools"
      xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/goMusic"
        android:title="음악" />
    <item
        android:id="@+id/goVideo"
        android:title="동영상" />
    <item
        android:id="@+id/goMap"
        android:title="구글 지도" />

</menu>
```

- 메뉴의 Item 별로 선택되었을 시, **인텐트**를 사용하여 각각에 해당하는 클래스로 이동

Ex) 동영상 클릭 시 -> VideoActivity.java

```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()) {
        case R.id.goMap:
            Intent mapintent=new Intent(getApplicationContext(),MapActivity.class);
            startActivity(mapintent);
            return true;
        case R.id.goVideo:
            Intent videointent=new Intent(getApplicationContext(),VideoActivity.class);
            startActivity(videointent);
            return true;
        case R.id.goMusic:
            Intent musicintent=new Intent(getApplicationContext(),MusicActivity.class);
            startActivity(musicintent);
            return true;
    }
    return false;
}
```

- 메인 이동 버튼 클릭 시, **인텐트**를 사용하여 SecondMainActivity.java 로!

```
btnMain.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent appintent=new Intent(getApplicationContext(),SecondMainActivity.class);
        startActivity(appintent);
    }
});
```

Term Project : Code

21

[SecondMainActivity.java]

- TabHost와 **intent**를 사용하여 소개, 포스터, 평가/리뷰, 부가 정보에 맞는 class를 연결시켜줬다.

//첫 번째 탭

```
TabHost tabhost = getTabHost();  
TabHost.TabSpec spec;  
Intent intent = new Intent().setClass(this, IntroActivity.class);
```

```
spec = tabhost.newTabSpec( tag: "intro").setIndicator("소개").setContent(intent);  
tabhost.addTab(spec);
```

//두 번째 탭

```
intent = new Intent().setClass(this, PosterActivity.class);  
spec = tabhost.newTabSpec( tag: "poster").setIndicator("포스터").setContent(intent);  
tabhost.addTab(spec);
```

//세 번째 탭

```
intent = new Intent().setClass(this, ReviewActivity.class);  
spec = tabhost.newTabSpec( tag: "review").setIndicator("평가/리뷰").setContent(intent);  
tabhost.addTab(spec);
```

//네 번째 탭

```
intent = new Intent().setClass(this, AddActivity.class);  
spec = tabhost.newTabSpec( tag: "add").setIndicator("부가정보").setContent(intent);  
tabhost.addTab(spec);
```

Term Project : Code

22

[IntroActivity.java / intro.xml]

- 대화상자(dialog1.xml) 사용.
- Switch 문을 사용하여 라디오 버튼 클릭 시, 대화상자에 저장 될 title, 내용, 그림 값을 변경하게 설정.

```
btnDetail.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        dialogView = View.inflate( context: IntroActivity.this, R.layout.dialog1, root: null);
        AlertDialog.Builder dlg = new AlertDialog.Builder( context: IntroActivity.this);

        ivPoster = dialogView.findViewById(R.id.ivPoster);
        tvContext = dialogView.findViewById(R.id.tvContext);
        dlg.setView(dialogView);
        dlg.setIcon(R.drawable.ahibli);

        switch (rGroup1.getCheckedRadioButtonId()){
            case R.id.totoro:
                dlg.setTitle("여우집 토토로");
                ivPoster.setImageResource(R.drawable.totoro);
                tvContext.setText(" - 개봉 : 2001.07.28 \n - 등급 : 전체 관람가 \n - 장르 : 애니메이션, 가족, 판타지 \n - 러닝타임 : 87분\n");
                break;
            case R.id.moving:
                dlg.setTitle("하늘의 움직이는 성");
                ivPoster.setImageResource(R.drawable.moving);
                tvContext.setText(" - 개봉 : 2004.12.23 \n - 등급 : 전체 관람가 \n - 장르 : 애니메이션, 판타지 \n - 러닝타임 : 119분\n");
                break;
            case R.id.laputa:
                dlg.setTitle("천공의 성 라퓨타");
                ivPoster.setImageResource(R.drawable.laputa);
                tvContext.setText(" - 개봉 : 2004.04.30 \n - 등급 : 전체 관람가 \n - 장르 : 애니메이션, 판타지, 모험 \n - 러닝타임 : 124분\n");
                break;
            case R.id.mo:
                dlg.setTitle("모노노케 히메");
                ivPoster.setImageResource(R.drawable.mo);
                tvContext.setText(" - 개봉 : 2003.04.25 \n - 등급 : 전체 관람가 \n - 장르 : 애니메이션, 모험, 액션 \n - 러닝타임 : 135분\n");
                break;
            case R.id.kiki:
                dlg.setTitle("마녀 배달부 키키");
                ivPoster.setImageResource(R.drawable.kiki);
                tvContext.setText(" - 개봉 : 2007 .11.22 \n - 등급 : 전체 관람가 \n - 장르 : 애니메이션, 모험, 판타지, 가족 \n - 러닝타임 : 102분\n");
                break;
        }

        dlg.setPositiveButton( text: "돌아가기!", listener: null);
        dlg.show();
    }
});
```

- HOME 버튼 클릭 시, 메인 화면으로 이동하도록 finish()사용

```
btnHome.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
```


Term Project : Code

23

[PosterActivity.java / poster.xml]

- 리스트 뷰와 이미지 뷰를 사용한다.

```
<ListView
    android:id="@+id/listview"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="2"
    android:background="#E0F1F2">
</ListView>
```

```
<ImageView
    android:id="@+id/photo"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="3"
    android:background="#8ECCC6"/>
```

- Title과 posterID 배열을 사용하여, 선택된 제목과 이미지가 맞게 사용한다.
- 리스트 뷰를 사용하여 선택되었을 때, 해당 값에 맞는 포스터 이미지를 이미지 뷰에서 보여준다.
- 직접 만든 토스트(toast1.xml)를 사용한다.

```
final String[] title = {"어췌집 도토로", "하울의 움직이는 성", "센과 치히로의 행방불명", "고양이의 보은", "마루 밑 아리여터",
    "천공의 성 라푸타", "모노노케 히메", "마녀 배달부 키키"};
final Integer[] posterID = {R.drawable.totoro, R.drawable.moving, R.drawable.sc, R.drawable.cat, R.drawable.maru, R.drawable.laputa,
    R.drawable.mo, R.drawable.kiki};

ListView list = findViewById(R.id.ListView);

ArrayAdapter<String> adapter = new ArrayAdapter<>(context, this, android.R.layout.simple_list_item_1, title);
list.setAdapter(adapter);

list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2, long arg3) {
        photo.setImageResource(posterID[arg2]);
        //Toast.makeText(getApplicationContext(), title[arg2], Toast.LENGTH_SHORT).show();
        toastText.setText(title[arg2]);
        toast.setView(toastView);
        toast.show();
    }
});
```

Term Project : Code

24

[ReviewActivity.java / review.xml]

- 버튼 8개, 레이팅 바, 스크롤 뷰, 텍스트 뷰를 이용한다.
- 버튼 클릭 시, 레이팅 바에 평점을 보여주고 텍스트 뷰에 저장된 리뷰 파일을 불러온다.
- **파일 처리**를 사용한다. (raw에 텍스트 파일을 넣고 불러온다.)

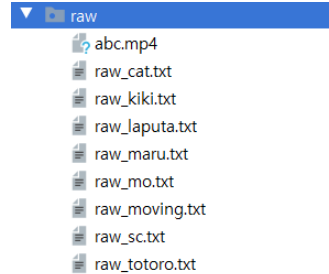
```
<RatingBar
    android:id="@+id/star"
    android:layout_gravity="center"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="10"
    style="@style/Widget.AppCompat.RatingBar.Indicator"/>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#8ECCC6">

    <TextView
        android:id="@+id/tvReview"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:textColor="#000000"
        android:textSize="18dp"/>

</ScrollView>
```

```
//아웃집 토토로 클릭시 -> 평점과 리뷰 확인
btnTotoro.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try{
            star.setRating((float) 9.5);
            InputStream inputs = getResources().openRawResource(R.raw.raw_totoro);
            byte[] txt = new byte[inputs.available()];
            inputs.read(txt);
            tvReview.setText(new String(txt));
            inputs.close();
        }catch (IOException e) {}
    }
});
```



Term Project : Code

25

[AddActivity.java / add.xml]

- 버튼 클릭 시, **intent**를 사용하여 기재된 Uri로 이동되도록 한다.
- 이때, 토스트를 통해 어디로 이동하는지 다시 한 번 확인할 수 있다.

```
//지브리 공식 홈페이지 클릭시
btnGhibli.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(getApplicationContext(), text: "지브리 공식 홈페이지로 이동합니다.", Toast.LENGTH_SHORT).show();
        Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.ghibli.jp/"));
        startActivity(mIntent);
    }
});

//도토리숲 공식 홈페이지 클릭시
btnDotori.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(getApplicationContext(), text: "도토리숲 공식 홈페이지로 이동합니다.", Toast.LENGTH_SHORT).show();
        Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.dotorisup.com/"));
        startActivity(mIntent);
    }
});
```

- HOME 버튼 클릭 시, 메인 화면으로 이동하도록 finish()사용

```
btnHome.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
```

Term Project : Code

26

[MusicActivity.java / music.xml]

- Switch를 사용하여 노래를 듣고 끌 수 있도록 설정하였다.

```
<Switch
    android:id="@+id/switch1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="아웃집 토토로(1) 듣기"/>
```

```
<Switch
    android:id="@+id/switch2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="아웃집 토토로(2) 듣기"
/>
```

```
<Switch
    android:id="@+id/switch3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="센과 치히로의 행방불명 듣기"/>
```

- BACK 버튼 클릭 시, finish()를 사용하여 메인(홈)화면으로 가도록 설정해주었다.

```
//BACK 버튼 클릭시
btnBack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
```

- 미디어 플레이어를 사용하여, raw폴더에 있는 음악을 연결시켰다.

```
final MediaPlayer mPlayer1, mPlayer2, mPlayer3;
mPlayer1=MediaPlayer.create( context: this, R.raw.song1);
mPlayer2=MediaPlayer.create( context: this, R.raw.song2);
mPlayer3=MediaPlayer.create( context: this, R.raw.song3);
```

- 스위치를 키면, 음악이 재생되고, 스위치를 끄면 음악이 꺼지도록 설정하였다.

```
//아웃집 토토로 듣기(1) 스위치 키고 끌때
switch1.setOnClickListener((v) -> {
    if (switch1.isChecked() == true){
        mPlayer1.start();
    }
    else
        mPlayer1.stop();
});
```

```
//아웃집 토토로 듣기(2) 스위치 키고 끌때
switch2.setOnClickListener((v) -> {
    if (switch2.isChecked()==true) {
        mPlayer2.start();
    }
    else
        mPlayer2.stop();
});
```

```
//센과 치히로의 행방불명 스위치 키고 끌때
switch3.setOnClickListener((v) -> {
    if (switch3.isChecked()==true) {
        mPlayer3.start();
    }
    else
        mPlayer3.stop();
});
```

Term Project : Code

27

[VideoActivity.java / video.xml]

- 비디오 뷰를 사용하여 비디오 사용을 가능하게 하였다.

```
<VideoView
    android:id="@+id/vv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

- BACK 버튼 클릭 시, finish()를 사용하여 메인(홈)화면으로 가도록 설정해주었다.

```
btnBack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
```

- 비디오의 URI를 설정하였고, 이 프로젝트에서는 raw 파일에 있는 abc 라는 이름의 비디오를 사용하였다.
- 컨트롤 바를 붙여주어 비디오 뷰의 재생, 일시정지 등의 기능을 가능하게 하였다.

```
//비디오의 URI
Uri videoUri = Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.abc);

//비디오뷰의 재생, 일시정지 등을 할 수 있는 '컨트롤바'를 붙여주는 작업
vv.setMediaController(new MediaController(context, this));

vv.setVideoURI(videoUri);

vv.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
    @Override
    public void onPrepared(MediaPlayer mediaPlayer) {
        vv.start();
    }
});
```



Term Project : Code

28

[AndroidManifest.xml]

```
<uses-feature android:glEsVersion="0x00020000" android:required="true" />
<permission android:name="com.cookandroid.termproject.permission.MAPS_RECEIVE" android:label="signature" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<meta-data android:name="com.google.android.gms.version" android:value="@integer.full_version" />

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyAeZrg1tbLW2H_B_RBW1TpqkUfTvAt9hdY" />

<uses-library
    android:name="org.apache.http.legacy"
    android:required="false" />
```

[build.gradle]

```
dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    implementation 'com.google.android.gms:play-services-maps:16.0.0'
    implementation 'com.google.android.gms:play-services-location:16.0.0'
```

API Key
AIzaSyAeZrg1tbLW2H_B_RBW1TpqkUfTvAt9hdY

Android 앱의 사용량 제한

Android 앱의 사용량을 제한하려면 패키지 이름과 SHA-1 서명 인증서 디지털 지문을 추가하세요.

com.cookandroid.termproject,
1A:49:A8:77:6C:F0:CA:15:96:C7:7B:9E:CC:5F:2C:AB:A7:E8:C8:14

항목 추가

[MapsActivity.java / map.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" />
```

```
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
    .findFragmentById(R.id.map);
mapFragment.getMapAsync(new OnMapReadyCallback() {
    @Override
```

```
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    LatLng ghibli = new LatLng(37.513463, 127.103509);
    mMap.getUiSettings().setZoomControlsEnabled(true);
    mMap.addMarker(new MarkerOptions().position(ghibli).title("Marker in Ghibli Shop"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(ghibli));
}
```

- Xml에서 프래그먼트로 설정해준다.
- LatLng를 통해 내가 원하는 좌표 값을 입력한 후, addMarker를 통해 입력한 좌표와, 마커의 이름을 설정해 보여준다.
- setZoomControlsEnabled를 사용하여 (+),(-)를 사용할 수 있게 한다.

Term Project 후기

29

- 텀 프로젝트를 진행하면서 실제 배워온 이론 내용을 토대로 실습을 진행해 보았다. 아무래도 이론에서 멈추는 것이 아닌 이를 직접 실습으로 실행을 시키다 보니 어려운 부분도 매우 많았다. 이론을 공부할 때와는 다르게, 실제로 디테일하게 신경 쓸 부분이 매우 많았고, 사소한 오타에도 프로그램 전체가 돌아가지 않아서 애를 먹는 때도 있었다. 정말 수많은 오류를 겪으면서 다시 코드에 관한 내용이나, 주의할 점을 상기시키고 기억할 수 있었고, 개념에서 멈추는 것이 아닌 이를 직접 내가 만들고, 실행시키며 얻는 만족감을 얻기도 했다. XML에서 전체적인 틀을 짜고 구성하며, 기능들을 각각의 Java 클래스와 연결하면서 많은 기능을 구현하는 앱을 구상할 수도 있었다. XML보다 자바에서 오류가 굉장히 자주 일어났고 이를 고치는 게 쉽지만은 않았다. 처음에 계획한 것보다 기능이 줄어들기도 하고, 오류를 수정하는 데 오랜 시간이 걸리기도 했다. 방학 때, 안드로이드 공부를 깊이 있게 해보고 이번에 텀 프로젝트에서 사용하지 못한 코드들을 사용하여, 실제로 완성도 높은 앱을 만들어보고자 한다.
- 한 학기 동안 학생들과 수시로 피드백해주시고, 개선해주신 점이 가장 감사드리고, 많은 범위의 내용을 꼼꼼하게 가르쳐 주신 점 감사드립니다 :) 초반의 수업에서 코드 작성하는 데 속도가 빨라 따라가지 못해 굉장히 어려움을 겪기도 했는데, 나중에 LMS로 코드 작성하는 부분 따로 영상녹화를 해 주셔서 다시 돌려볼 수 있다는 게 굉장히 좋았습니다! 화질 개선도 되며 개선되는 점도 굉장히 감사드립니다! 한 학기동안 정말 많은 것을 배워갈 수 있었습니다. 감사합니다 교수님 :)