

스마트앱프로그래밍 Term Project 결과 보고서

분반	1 분반		
작 품 명	IMDB 장르별 추천 영화 어플리케이션		
개발기간	2018년 9 월 1 일 ~ 2018년 12 월 21 일		
지도교수	컴퓨터공학과	오병우	
구 분	학 년	학 번	성 명
책임자(팀장)	3	20161376	윤예준
팀원1	해당 사항 없음		
팀원2	해당 사항 없음		
팀원3	해당 사항 없음		
<p>본인은 스마트앱프로그래밍 Term Project 결과 보고서를 첨부과 같이 제출합니다. 프로젝트를 진행할 때 부정확한 방법을 사용했거나 보고서에 사실과 다른 내용을 적었을 때는 모든 처벌을 감수하겠습니다.</p> <p style="text-align: right;">2018년 12월 23일</p> <p style="text-align: right;">팀 장 윤 예 준 .</p> <p style="text-align: center;">컴퓨터공학심화프로그램</p>			

목차

0. 결과요약	<u>3</u>
□ Level 1 목표 완료 증명	<u>3</u>
□ Level 2 목표 완료 증명	<u>10</u>
□ Level 3 목표 완료 증명	<u>12</u>
□ Level 4 목표 완료 증명	<u>13</u>
1. 서론	<u>16</u>
2. 작품 개요	<u>16</u>
3. 설계	<u>16</u>
4. 구현	<u>20</u>
5. 결론	<u>27</u>
6. 자체 분석과 평가	<u>27</u>
7. 부록	<u>20</u>

결과 보고서

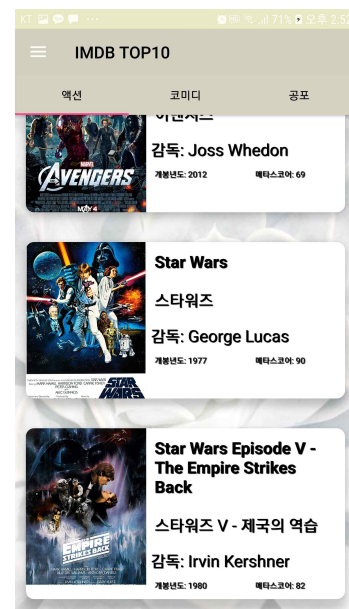
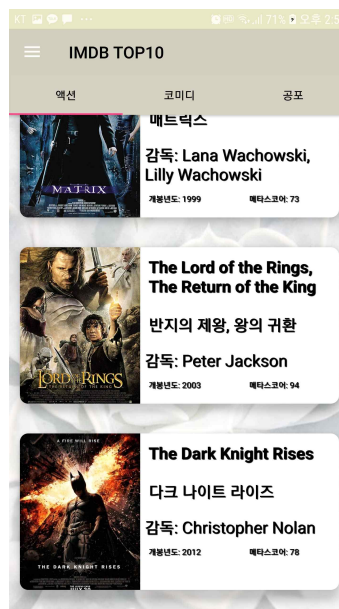
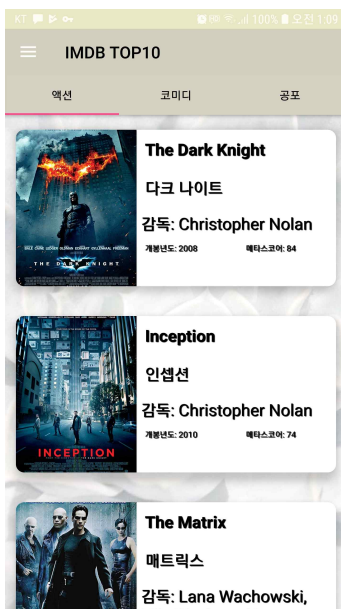
IMDB 장르별 추천 영화 어플리케이션

0. 결과 요약

Level	목표	완료	세부 목표 완성도	기타 (미완성 사유/문제점 등)
1 (70점)	1.1 RecyclerView 및 CardView를 사용한 리스트 개발	○	RecyclerView 사용: ○ CardView 사용: ○	
	1.2 DB 구축 및 PHP를 통해 select query 실행한 후 결과를 JSON 형태로 반환	○	DB 구축: ○ Volley 사용: ○ JSON 반환 PHP 개수: 3개	
	1.3 JSON 결과를 분석하여 리스트 출력	○	DB 내용이 리스트에 출력됨: ○	
	1.4 Custom List Item Layout에서 NetworkImageView 사용	○	NetworkImageView 사용: ○	
2 (80점)	2.1 Item을 누르면 Intent로 Activity 실행하여 세부 내용 출력	○	Activity로 세부 내용 출력: ○	
3 (90점)	3.1 TabView 사용	○	Fragment에 RecyclerView 및 CardView 출력: ○ 슬라이딩 메뉴에서	
4 (100점)	4.1 Navigation Drawer 통합 및 Tab 연결	○	누르면 해당 Tab으로 이동: ○	

□ Level 1 목표 완료 증명

- 1.1 RecyclerView 및 CardView를 사용한 리스트 개발
 - 리스트 화면 캡처



- RecyclerView가 들어간 xml 파일 부분

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/constraintLayout"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <android.support.v7.widget.RecyclerView
8         android:id="@+id/recyclerView"
9         android:layout_width="match_parent"
10        android:layout_height="match_parent" />
11 </LinearLayout>
```

- CardView가 들어간 xml 파일 부분

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:card_view="http://schemas.android.com/apk/res-auto"
5     android:layout_width="match_parent"
6     android:layout_height="wrap_content"
7     app:cardCornerRadius="10dp"
8     app:cardElevation="10dp"
9     card_view:cardUseCompatPadding="true">
10
11     <LinearLayout
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:orientation="horizontal">
15
16         <com.android.volley.toolbox.NetworkImageView
17             android:id="@+id/imageUrl"
18             android:layout_width="145dp"
19             android:layout_height="189dp"
20             android:scaleType="centerCrop"
21             app:layout_constraintStart_toStartOf="parent"
22             app:layout_constraintTop_toTopOf="parent" />
23
24         <LinearLayout
25             android:layout_width="match_parent"
26             android:layout_height="wrap_content"
27             android:orientation="vertical">
28
29             <TextView
30                 android:id="@+id/engName"
31                 android:layout_width="wrap_content"
32                 android:layout_height="wrap_content"
33                 android:layout_weight="1"
34                 android:padding="10dp"
35                 android:shadowColor="#000"
36                 android:shadowDx="3.0"
37                 android:shadowDy="2.0"
38                 android:shadowRadius="2.0"
39                 android:textColor="#000"
40                 android:textSize="20sp"
41                 android:textStyle="bold" />
42
43             <TextView
44                 android:id="@+id/korName"
45                 android:layout_width="wrap_content"
46                 android:layout_height="53dp"
47                 android:padding="10dp"
48                 android:shadowColor="#000"
49                 android:shadowDx="3.0"
50                 android:shadowDy="2.0"
51                 android:shadowRadius="2.0"
52                 android:textColor="#000"
53                 android:textSize="20sp" />
54
55             <TextView
56                 android:id="@+id/dir"
57                 android:layout_width="wrap_content"
58                 android:layout_height="wrap_content"
59                 android:layout_marginLeft="5dp"
60                 android:shadowColor="#000"
61                 android:shadowDx="3.0"
```

```

62         android:shadowDy="2.0"
63         android:shadowRadius="2.0"
64         android:textColor="#000"
65         android:textSize="20sp" />
66
67     <LinearLayout
68         android:layout_width="match_parent"
69         android:layout_height="wrap_content"
70         android:orientation="horizontal">
71
72         <TextView
73             android:id="@+id/year"
74             android:layout_width="wrap_content"
75             android:layout_height="wrap_content"
76             android:layout_weight="1"
77             android:padding="10dp"
78             android:shadowColor="#000"
79             android:shadowDx="3.0"
80             android:shadowDy="2.0"
81             android:shadowRadius="2.0"
82             android:textColor="#000"
83             android:textSize="10sp" />
84
85         <TextView
86             android:id="@+id/metaScore"
87             android:layout_width="wrap_content"
88             android:layout_height="wrap_content"
89             android:layout_weight="1"
90             android:padding="10dp"
91             android:shadowColor="#000"
92             android:shadowDx="3.0"
93             android:shadowDy="2.0"
94             android:shadowRadius="2.0"
95             android:textColor="#000"
96             android:textSize="10sp" />
97     </LinearLayout>
98 </LinearLayout>
99 </LinearLayout>
100 </android.support.v7.widget.CardView>
101

```

○ 1.2 DB 구축 및 PHP를 통해 select query 실행한 후 결과를 JSON 형태로 반환

- DB 구축 SQL

```

1  drop database if exists IMDB;
2  create database IMDB DEFAULT CHARACTER SET utf8 collate utf8_general_ci;
3  use IMDB;
4  set names utf8;
5  create table action (
6      id int(11) not null auto_increment,
7      engname varchar(100),
8      korname varchar(100),
9      director varchar(50),
10     year dec(4),
11     imageUrl varchar(200),
12     metascore int(11),
13     note varchar(255),
14     primary key (id),
15     index(engname)
16 ) ENGINE=InnoDB default character set utf8 collate utf8_general_ci;
17 set names euckr;
18
19 insert into action values ( null, "The Dark Knight", "다크 나이트", "Christopher Nolan", 2008, "movie/The Dark Knight.jpg", 84,
20 "When the menace known as the Joker emerges from his mysterious past, he wreaks havoc and chaos on the people of Gotham. The Dark Knight must accept one of the gre
21 insert into action values ( null, "Inception", "인셉션", "Christopher Nolan", 2010, "movie/Inception.jpg", 74,
22 "A thief who steals corporate secrets through the use of dream-sharing technology is given the inverse task of planting an idea into the mind of a CEO.");
23 insert into action values ( null, "The Matrix", "매트릭스", "Lana Wachowski, Lilly Wachowski ", 1999, "movie/The Matrix.jpg", 73,
24 "A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.");
25 insert into action values ( null, "The Lord of the Rings, The Return of the King", "반지의 제왕, 왕의 귀환", "Peter Jackson ", 2003, "movie/The Lord of the Rings, Th
26 "Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.");
27 insert into action values ( null, "The Dark Knight Rises ", "다크 나이트 라이즈", "Christopher Nolan", 2012, "movie/The Dark Knight Rises.jpg", 78,
28 "Eight years after the Joker's reign of anarchy, Batman, with the help of the enigmatic Catwoman, is forced from his exile to save Gotham City, now on the edge of
29 insert into action values ( null, "Gladiator ", "글래디에이터", "Ridley Scott", 2000, "movie/Gladiator.jpg", 67,
30 "A former Roman General sets out to exact vengeance against the corrupt emperor who murdered his family and sent him into slavery.");
31 insert into action values ( null, "Batman Begins ", "배트맨 비긴즈", "Christopher Nolan", 2005, "movie/Batman Begins.jpg", 70,
32 "After training with his mentor, Batman begins his fight to free crime-ridden Gotham City from corruption.");
33 insert into action values ( null, "The Avengers", "어벤져스", "Joss Whedon", 2012, "movie/The Avengers.jpg", 69,
34 "Earth's mightiest heroes must come together and learn to fight as a team if they are going to stop the mischievous Loki and his alien army from enslaving humanity
35 insert into action values ( null, "Star Wars ", "스타워즈", "George Lucas ", 1977, "movie/Star Wars.jpg", 90,
36 "Luke Skywalker joins forces with a Jedi Knight, a cocky pilot, a Wookiee and two droids to save the galaxy from the Empire's world-destroying battle station, whil
37 insert into action values ( null, "Star Wars Episode V - The Empire Strikes Back", "스타워즈 V - 제국의 역습", "Irvin Kershner", 1980, "movie/Star Wars Episode V - 1
38 "After the rebels are brutally overpowered by the Empire on the ice planet Hoth, Luke Skywalker begins Jedi training with Yoda, while his friends are pursued by Da
39
40 select * from action;

```

<action.sql>


```

1 use IMDB;
2 set names utf8;
3 create table comedy (
4   id int(11) not null auto_increment,
5   engname varchar(100),
6   korname varchar(100),
7   director varchar(50),
8   year dec(4),
9   imageUrl varchar(200),
10  metascore int(11),
11  note varchar(255),
12  primary key (id),
13  index(engname)
14 ) ENGINE=InnoDB default character set utf8 collate utf8_general_ci;
15 set names euckr;
16
17 insert into comedy values ( null, "The Wolf of Wall Street", "더 울프 오브 스트리트", "Martin Scorsese", 2013, "movie/The Wolf of Wall Street.jpg", 75,
18 "Based on the true story of Jordan Belfort, from his rise to a wealthy stock-broker living the high life to his fall involving crime, corruption and the fedel
19 insert into comedy values ( null, "Back to the Future", "백 투 더 퓨처", "Robert Zemeckis", 1985, "movie/Back to the Future.jpg", 87,
20 "Marty McFly, a 17-year-old high school student, is accidentally sent thirty years into the past in a time-traveling DeLorean invented by his close friend, th
21 insert into comedy values ( null, "Guardians of the Galaxy", "가디언즈 오브 갤럭시", "James Gunn", 2014, "movie/Guardians of the Galaxy.jpg", 76,
22 "A group of intergalactic criminals are forced to work together to stop a fanatical warrior from taking control of the universe.");
23 insert into comedy values ( null, "Finding Nemo", "니모를 찾아서", "Andrew Stanton, Lee Unkrich", 2009, "movie/Finding Nemo.jpg", 90,
24 "After his son is captured in the Great Barrier Reef and taken to Sydney, a timid clownfish sets out on a journey to bring him home.");
25 insert into comedy values ( null, "UP", "업", "Pete Docter, Bob Peterson", 1998, "movie/UP.jpg", 88,
26 "Seventy-eight year old Carl Fredricksen travels to Paradise Falls in his home equipped with balloons, inadvertently taking a young stowaway.");
27 insert into comedy values ( null, "The Truman Show", "트루먼 쇼", "Peter Weir", 1998, "movie/The Truman Show.jpg", 90,
28 "An insurance salesman/adjuster discovers his entire life is actually a television show.");
29 insert into comedy values ( null, "Deadpool", "데드풀", "Tim Miller", 2016, "movie/Deadpool.jpg", 65,
30 "A fast-talking mercenary with a morbid sense of humor is subjected to a rogue experiment that leaves him with accelerated healing powers and a quest for rev
31 insert into comedy values ( null, "Toy Story", "토이스토리", "John Lasseter", 1995, "movie/Toy Story.jpg", 95,
32 "A cowboy doll is profoundly threatened and jealous when a new spaceman figure supplants him as top toy in a boy's room.");
33 insert into comedy values ( null, "Monsters, Inc", "몬스터 주식회사", "Pete Docter, David Silverman", 2001, "movie/ Monsters, Inc.jpg", 78,
34 "In order to power the city, monsters have to scare children so that they scream. However, the children are toxic to the monsters, and after a child gets thr
35 insert into comedy values ( null, "Snatch", "스내치", "Guy Ritchie", 2000, "movie/Snatch.jpg", 55,
36 "Unscrupulous boxing promoters, violent bookmakers, a Russian gangster, incompetent amateur robbers and supposedly Jewish jewelers fight to track down a pric
37
38 select * from comedy;

```

<comedy.sql>

```

1 use IMDB;
2 set names utf8;
3 create table horror (
4   id int(11) not null auto_increment,
5   engname varchar(100),
6   korname varchar(100),
7   director varchar(50),
8   year dec(4),
9   imageUrl varchar(200),
10  metascore int(11),
11  note varchar(255),
12  primary key (id),
13  index(engname)
14 ) ENGINE=InnoDB default character set utf8 collate utf8_general_ci;
15 set names euckr;
16
17 insert into horror values ( null, "The Shining", "샤이닝", "Stanley Kubrick", 1980, "movie/The Shining.jpg", 63,
18 "A family heads to an isolated hotel for the winter where an evil spiritual presence influences the father into violence, while his psychic son sees horrific forebodings from bot
19 insert into horror values ( null, "Alien", "매탈리언", "Ridley Scott", 1979, "movie/Alien.jpg", 83,
20 "After a space merchant vessel perceives an unknown transmission as a distress call, its landing on the source moon finds one of the crew attacked by a mysterious lifeform, and t
21 insert into horror values ( null, "I Am Legend", "나는 전설이다", "Francis Lawrence", 2007, "movie/I Am Legend.jpg", 65,
22 "Years after a plague kills most of humanity and transforms the rest into monsters, the sole survivor in New York City struggles valiantly to find a cure.");
23 insert into horror values ( null, "World War Z", "월드워Z", "Andrew Stanton, Lee Unkrich", 2013, "movie/World War Z.jpg", 63,
24 "Former United Nations employee Gerry Lane traverses the world in a race against time to stop the Zombie pandemic that is toppling armies and governments, and threatening to dest
25 insert into horror values ( null, "Psycho", "싸이코", "Alfred Hitchcock", 1960, "movie/Psycho.jpg", 97,
26 "A Phoenix secretary embezzles forty thousand dollars from her employer's client, goes on the run, and checks into a remote motel run by a young man under the domination of his m
27 insert into horror values ( null, "Shaun of the Dead", "새벽의 황당함 저주", "Edgar Wright", 2004, "movie/Shawn of the Dead.jpg", 76,
28 "A man decides to turn his moribund life around by winning back his ex-girlfriend, reconciling his relationship with his mother and dealing with an entire community that has retu
29 insert into horror values ( null, "Zombieland", "좀비랜드", "Ruben Fleischer", 2009, "movie/Zombieland.jpg", 73,
30 "A shy student trying to reach his family in Ohio, a gun-toting tough guy trying to find the last Twinkie, and a pair of sisters trying to get to an amusement park join forces to
31 insert into horror values ( null, "The Conjuring", "컨저링", "James Wan", 2013, "movie/The Conjuring.jpg", 68,
32 "Paranormal investigators Ed and Lorraine Warren work to help a family terrorized by a dark presence in their farmhouse.");
33 insert into horror values ( null, "Get Out, Inc", "겟 아웃", "Jordan Peele", 2017, "movie/Get Out.jpg", 84,
34 "A young African-American visits his white girlfriend's parents for the weekend, where his simmering uneasiness about their reception of him eventually reaches a boiling point.")
35 insert into horror values ( null, "Saw", "쏘우", "James Wan", 2004, "movie/Saw.jpg", 46,
36 "Two strangers, who awaken in a room with no recollection of how they got there, soon discover they're pawns in a deadly game perpetrated by a notorious serial killer.");
37
38 select * from horror;

```

<horror.sql>

- JSON을 반환하는 PHP 소스

```

1 <?php
2 $dbc = mysqli_connect('127.0.0.1', 'root',
3 'apmsetup', 'imdb')
4 or die('Error Connecting to MySQL server.');
```

<action.php>

```

1 <?php
2 $dbc = mysqli_connect('127.0.0.1', 'root',
3 'apmsetup', 'imdb')
4 or die('Error Connecting to MySQL server.');
```

<comedy.php>

```

1 <?php
2 $dbc = mysqli_connect('127.0.0.1', 'root',
3 'apmsetup', 'imdb')
4 or die('Error Connecting to MySQL server.');
```

<horror.php>

○ 1.3 JSON 결과를 분석하여 리스트 출력

- Volley 사용한 소스 코드 (수업 시간의 request...() 함수 부분)

```

1 @Override
2 public View onCreateView(LayoutInflater inflater, ViewGroup container,
3 Bundle savedInstanceState) {
4     View rootView = inflater.inflate(R.layout.fragment_main, container, false);
5     mRecyclerView = (RecyclerView) rootView.findViewById(R.id.recyclerView);
6     mRecyclerView.setAdapter(mAdapter);
7     mRecyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
8     mRecyclerView.setItemAnimator(new DefaultItemAnimator());
9     CookieHandler.setDefault(new CookieManager());
10    mQueue = Volley.newRequestQueue(getContext());
11    mImageLoader = new ImageLoader(mQueue, new ImageLoader.ImageCache() {
12        private final LruCache<String, Bitmap> cache = new LruCache<String, Bitmap>(20);
13        @Override
14        public Bitmap getBitmap(String url) {
15            return cache.get(url);
16        }
17        @Override
18        public void putBitmap(String url, Bitmap bitmap) {
19            cache.put(url, bitmap);
20        }
21    });
22    requestMovie();
23    return rootView;
24 }
25 protected void requestMovie() {
26     String url = SERVER_URL + genre;
27     JsonObjectRequest request =
28         new JsonObjectRequest(Request.Method.GET, url, null,
29             new Response.Listener<JSONObject>() {
30         @Override
31         public void onResponse(JSONObject response) {
```

```

32         Log.i("result", response.toString());
33         mResult = response;
34         parseJSON();
35     }
36 },
37     new Response.ErrorListener() {
38         @Override
39         public void onResponse(VolleyError error) {
40             Toast.makeText(getApplicationContext()
41 , error.getMessage(), Toast.LENGTH_LONG).show();
42             Log.i("result", error.toString());
43         }
44     });
45     request.setTag(Queue.TAG);
46     mQueue.add(request);
47 }

```

- JSON 분석 소스 코드 (수업 시간의 drawList() 함수 부분)

```

1 protected void parseJSON() {
2     mArray.clear();
3     try {
4         JSONArray array = mResult.getJSONArray("list");
5         for (int i = 0; i < array.length(); i++) {
6             JSONObject item = (JSONObject) array.get(i);
7             int id = item.getInt("id");
8             String engName = item.getString("engname");
9             String korName = item.getString("korname");
10            String director = item.getString("director");
11            String year = item.getString("year");
12            String imageUrl = item.getString("imageurl");
13            String metascore = item.getString("metascore");
14            String note = item.getString("note");
15            mArray.add(new Movie(id, engName, korName, director, year, imageUrl, metascore, note));
16        }
17    } catch (JSONException | NullPointerException e) {
18        Toast.makeText(getApplicationContext(), "Error " + e.toString(), Toast.LENGTH_LONG).show();
19        mResult = null;
20    }
21    mAdapter.notifyDataSetChanged();
22 }

```

○ 1.4 Custom List Item Layout에서 NetworkImageView 사용

- Custom List Item Layout 소스 코드

=> CardView가 들어간 xml 파일 부분 동일함 (1 ~ 101라인)

- NetworkImageView에 URL 지정하는 Java 소스 코드 부분

```
1 public class MovieAdapter extends RecyclerView.Adapter<MovieAdapter.MovieView> {
2     public class MovieView extends RecyclerView.ViewHolder implements View.OnClickListener {
3         public TextView txtEngName;
4         public TextView txtKorName;
5         public TextView txtDirector;
6         public TextView txtYear;
7         public NetworkImageView imageUrl;
8         public String txtNote;
9         public TextView txtMetascore;
10
11         public MovieView(View root) {
12             super(root);
13             root.setOnClickListener(this);
14             txtEngName = (TextView) root.findViewById(R.id.engName);
15             txtKorName = (TextView) root.findViewById(R.id.korName);
16             txtYear = (TextView) root.findViewById(R.id.year);
17             txtDirector = (TextView) root.findViewById(R.id.dir);
18             txtMetascore = (TextView) root.findViewById(R.id.metaScore);
19             imageUrl = (NetworkImageView) root.findViewById(R.id.imageUrl);
20         }
21         @Override
22         public void onClick(View view) {
23             Intent result = new Intent(getContext(), Item.class);
24             String movie = txtEngName.getText().toString();
25             String note = txtNote;
26             String year = txtYear.getText().toString();
27             result.putExtra("movie", movie);
28             result.putExtra("year", year);
29             result.putExtra("note", note);
30             startActivity(result);
31         }
32     }
33     public MovieAdapter() { }
34     @Override
35     public MovieView onCreateViewHolder(ViewGroup parent, int viewType) {
36         View root = LayoutInflater.from(parent.getContext()).inflate(R.layout.movie_item, parent, false);
37         return new MovieView(root);
38     }
39     @Override
40     public void onBindViewHolder(MovieView holder, int position) {
41         Movie g = mArray.get(position);
42         holder.txtEngName.setText(g.getEngName());
43         holder.txtKorName.setText(g.getKorName());
44         holder.imageUrl.setImageUrl(SERVER_URL + g.getImageUrl(), mImageLoader);
45         holder.txtDirector.setText("감독: " + g.getDirector());
46         holder.txtYear.setText("개봉년도: " + g.getYear());
47         holder.txtMetascore.setText("메타스코어: " + g.getMetascore());
48         holder.txtNote = g.getNote();
49     }
50     @Override
51     public int getItemCount() {
52         return mArray.size();
53     }
54 }
```

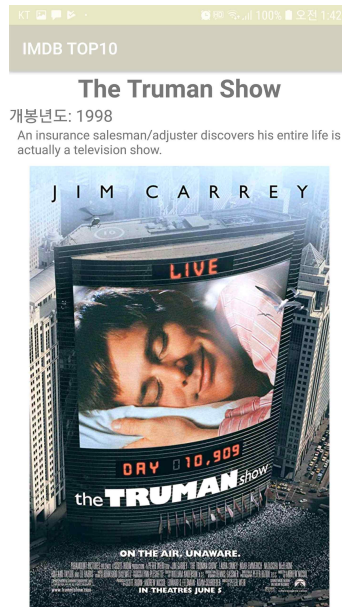
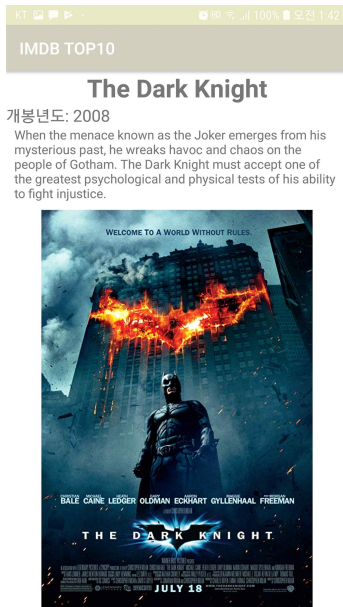
- DB에서 이미지 경로 저장하기 위한 DDL (create table) 및 DML (insert) SQL 부분 (1.2에 포함되면 생략 가능)

=> DB 구축 및 PHP를 통해 select query 실행한 후 결과를 JSON 형태로 반환 코드 에 동일

□ Level 2 목표 완료 증명

- 2.1 Item을 누르면 Intent로 Activity 실행하여 세부 내용 출력

- 화면 캡처



- 세부 내용 출력 Activity 소스 코드

```

1 public class Item extends AppCompatActivity {
2     protected ImageLoader mImageLoader = null;
3     protected RequestQueue mQueue = null;
4     public static final String SERVER_URL = "http://125.185.23.23/movie/";
5     @Override
6     protected void onCreate(Bundle savedInstanceState) {
7         super.onCreate(savedInstanceState);
8         setContentView(R.layout.activity_item);
9
10
11         Intent intent = getIntent();
12         TextView movie = (TextView)findViewById(R.id.movie);
13         movie.setText(intent.getStringExtra("movie"));
14         TextView year = (TextView)findViewById(R.id.date);
15         year.setText(intent.getStringExtra("year"));
16         TextView note = (TextView)findViewById(R.id.note);
17         note.setText(intent.getStringExtra("note"));
18
19         String temp = intent.getStringExtra("movie");
20         final String query = temp;
21         temp = temp+".jpg";
22
23         mQueue = Volley.newRequestQueue(this);
24         mImageLoader = new ImageLoader(mQueue,
25             new ImageLoader.ImageCache() {
26                 private final LruCache<String, Bitmap>

```

```

26         cache = new LruCache<String, Bitmap>(20);
27         @Override
28         public Bitmap getBitmap(String url) {
29             return cache.get(url);
30         }
31         @Override
32         public void putBitmap(String url, Bitmap bitmap) {
33             cache.put(url, bitmap);
34         }
35     });
36     NetworkImageView image = findViewById(R.id.img);
37     image.setImageUrl(SERVER_URL + temp, mImageLoader);
38
39     image.setOnClickListener(new View.OnClickListener(){
40         @Override
41         public void onClick(View v) {
42             String link = "https://www.google.co.kr/search?q="+ query;
43             Uri uri = Uri.parse(link);
44             Intent intent = new Intent(Intent.ACTION_VIEW,uri);
45             startActivity(intent);
46         }
47     });
48 }
49 }

```

- Intent를 생성하고 startActivity()를 호출하는 OnItemClickListener 소스 코드

```

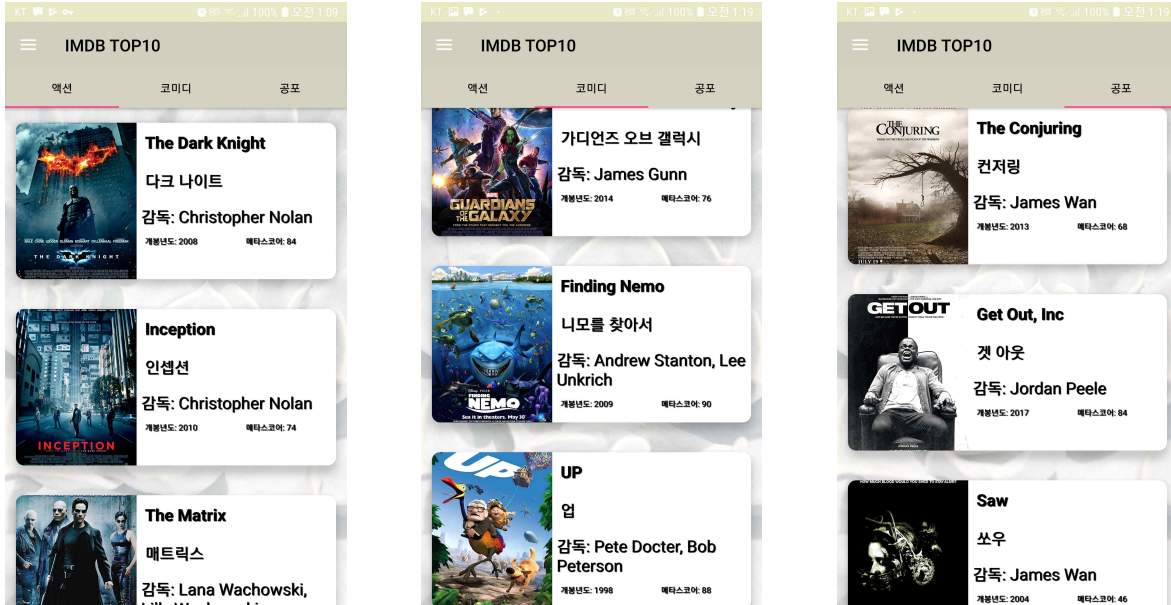
1 public class MovieAdapter extends RecyclerView.Adapter<MovieAdapter.MovieView> {
2     public class MovieView extends RecyclerView.ViewHolder implements View.OnClickListener {
3         public TextView txtEngName;
4         public TextView txtKorName;
5         public TextView txtDirector;
6         public TextView txtYear;
7         public NetworkImageView imageUrl;
8         public String txtNote;
9         public TextView txtMetascore;
10
11         public MovieView(View root) {
12             super(root);
13             root.setOnClickListener(this);
14             txtEngName = (TextView) root.findViewById(R.id.engName);
15             txtKorName = (TextView) root.findViewById(R.id.korName);
16             txtYear = (TextView) root.findViewById(R.id.year);
17             txtDirector = (TextView) root.findViewById(R.id.dir);
18             txtMetascore = (TextView) root.findViewById(R.id.metaScore);
19             imageUrl = (NetworkImageView) root.findViewById(R.id.imageUrl);
20         }
21
22         @Override
23         public void onClick(View view) {
24             Intent result = new Intent(getContext(), Item.class);
25             String movie = txtEngName.getText().toString();
26             String note = txtNote;
27             String year = txtYear.getText().toString();
28             result.putExtra("movie", movie);
29             result.putExtra("year", year);
30             result.putExtra("note", note);
31             startActivity(result);
32         }
33     }
34 }

```

□ Level 3 목표 완료 증명

○ 3.1 TabView 사용

- 각 Tab별 화면 캡처



- 각 Tab별 Fragment Class 소스 코드 부분

=> 각각의 탭의 비슷한 형식의 DB출력을 위하여 각각 페이지를 생성하지 않고 php파일을름을 String으로 전달하여 MovieFragment 클래스를 활용함

```
1 public class MovieFragment extends Fragment {
2
3     protected String genre;
4
5     public MovieFragment() {
6
7     }
8
9     @SuppressWarnings("ValidFragment")
10    public MovieFragment(String php) {
11        this.genre = php;
12    }
13    public class Movie {
14        int id;
15        String engName;
16        String korName;
17        String director;
18        String year;
19        String imageUrl;
20        String metascore;
21        String note;
22        public Movie(int id, String engName, String korName,
23            String director, String year, String imageUrl, String metascore, String note) {
24            //생성자 및 Getter함수 생략
25        }
26
27        public static final String QUEUE_TAG = "ActionRequest";
28        public static final String SERVER_URL = "http://125.185.23.23/";
29        protected ImageLoader mImageLoader = null;
30        protected RequestQueue mQueue = null;
31        protected JSONObject mResult = null;
32        protected ArrayList<Movie> mArray = new ArrayList<Movie>();
33        protected MovieAdapter mAdapter = new MovieAdapter();
34        protected RecyclerView mRecyclerView = null;
35    }
```

```

36     @Override
37     public View onCreateView(){} //1.3번 항목
38     protected void requestMovie(){} //1.3번 항목
39     protected void parseJSON() {} //1.3번 항목
40
41
42     public class MovieAdapter {} //1.3번 항목
43 }

```

- ViewPagerAdapter Class 소스 코드 부분

```

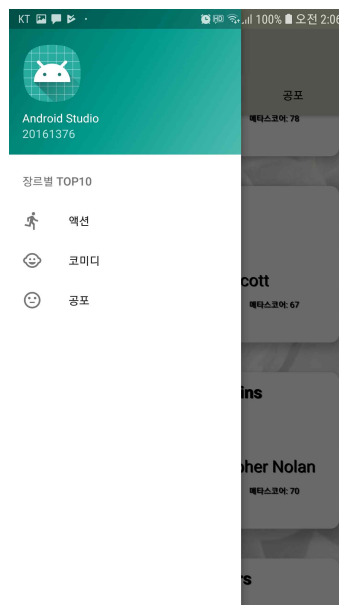
1 public class SectionsPagerAdapter extends FragmentPagerAdapter {
2
3     public SectionsPagerAdapter(FragmentManager fm) {
4         super(fm);
5     }
6
7     @Override
8     public Fragment getItem(int position) {
9         if (position == 0)
10             return new MovieFragment("action.php");
11         else if (position == 1)
12             return new MovieFragment("comedy.php");
13         else
14             return new MovieFragment("horror.php");
15     }
16
17     @Override
18     public int getCount() {
19         return 3;
20     }
21 }

```

□ Level 4 목표 완료 증명

○ 4.1 Navigation Drawer 통합 및 Tab 연결

- 화면 캡처



- 통합된 activity_main.xml

```
1 public class MainActivity extends AppCompatActivity
2     implements NavigationView.OnNavigationItemSelectedListener {
3
4
5     private SectionsPagerAdapter mSectionsPagerAdapter;
6     private ViewPager mViewPager;
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
13         setSupportActionBar(toolbar);
14
15         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
16         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
17             this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
18         drawer.addDrawerListener(toggle);
19         toggle.syncState();
20
21         NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
22         navigationView.setNavigationItemSelectedListener(this);
23
24         mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());
25
26         mViewPager = (ViewPager) findViewById(R.id.container);
27         mViewPager.setAdapter(mSectionsPagerAdapter);
28
29         TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);
30
31         mViewPager.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tabLayout));
32         tabLayout.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(mViewPager));
33     }
34
35     @Override
36     public void onBackPressed() {
37         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
38         if (drawer.isDrawerOpen(GravityCompat.START)) {
39             drawer.closeDrawer(GravityCompat.START);
40         } else {
41             super.onBackPressed();
42         }
43     }
44
45     @SuppressWarnings("StatementWithEmptyBody")
46     @Override
47     public boolean onNavigationItemSelected(MenuItem item) {
48         int id = item.getItemId();
49
50         if (id == R.id.nav_action) {
51             mViewPager.setCurrentItem(0);
52         } else if (id == R.id.nav_comedy) {
53             mViewPager.setCurrentItem(1);
54         } else if (id == R.id.nav_horror) {
55             mViewPager.setCurrentItem(2);
56         }
57
58         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
```

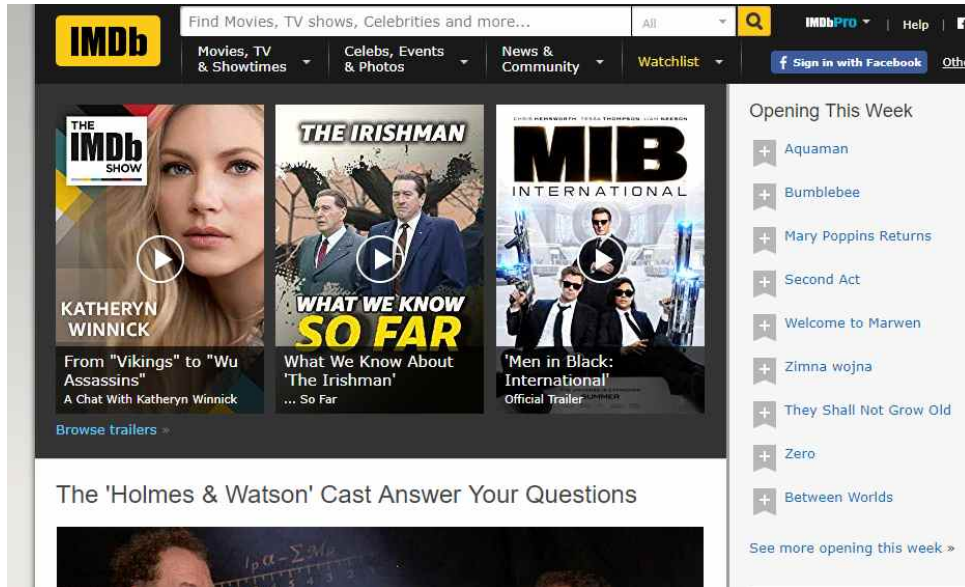
```
59     drawer.closeDrawer(GravityCompat.START);
60     return true;
61 }
62
63 public class SectionsPagerAdapter extends FragmentPagerAdapter {
64
65     public SectionsPagerAdapter(FragmentManager fm) {
66         super(fm);
67     }
68
69     @Override
70     public Fragment getItem(int position) {
71         if (position == 0)
72             return new MovieFragment("action.php");
73         else if (position == 1)
74             return new MovieFragment("comedy.php");
75         else
76             return new MovieFragment("horror.php");
77     }
78
79     @Override
80     public int getCount() {
81         return 3;
82     }
83 }
84 }
```

- 슬라이딩 메뉴에서 누르면 해당 Tab으로 이동하도록 수정한 onNavigationItemSelectedListener() 소스 코드

=> 4.1 통합된 activity_main.xml 47~61번라인

1. 서론

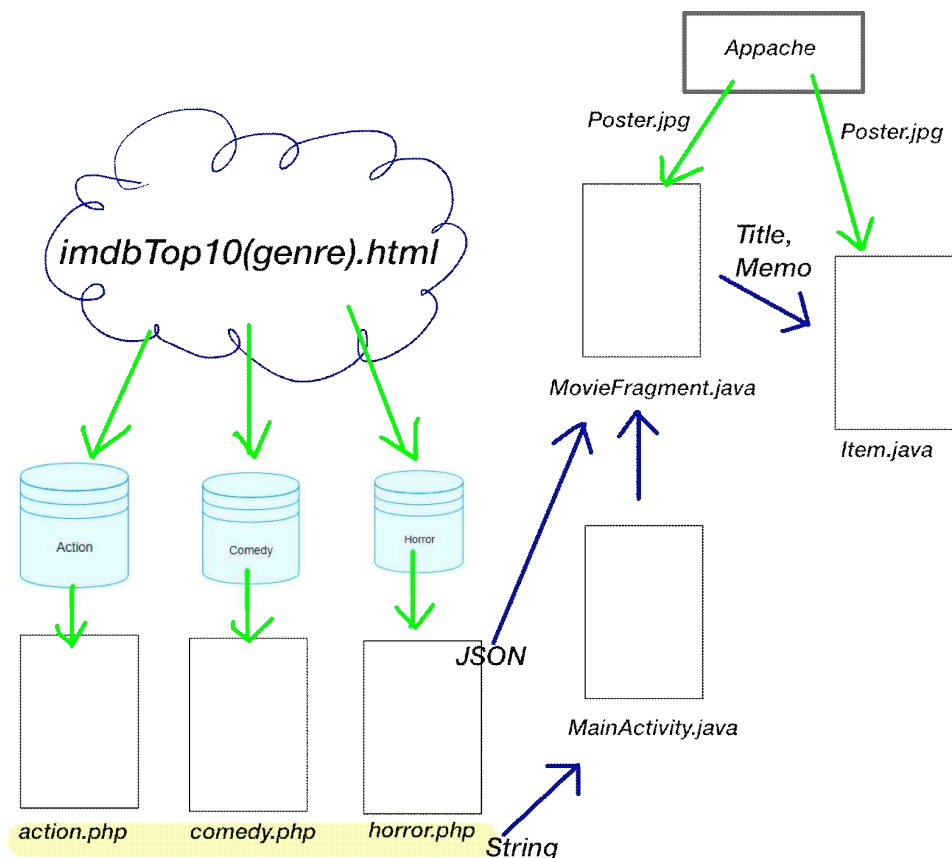
평소 저는 영화 감상하는 것을 즐겨 여러 영화를 보고 ‘왓차’ 라는 웹 및 어플리케이션으로 영화에 대한 평점을 내리고 추천을 받고 있습니다. 하지만 제가 여러 자료를 수집하면 ‘IMDB’ 라는 사이트를 발견했는데 해당사이트는 ‘왓차’에 비해 드라마, 게임, 배우 등 더 폭넓은 멀티미디어에 대한 자료와 평가들이 있습니다. 이 사이트의 정보를 통해 이용자에게 선호하는 장르의 영화, 드라마, 게임 등 다방면에서 사용자에게 추천하는 어플리케이션을 목표로 할 것입니다.



<IMDB 메인 화면>

2. 작품 개요

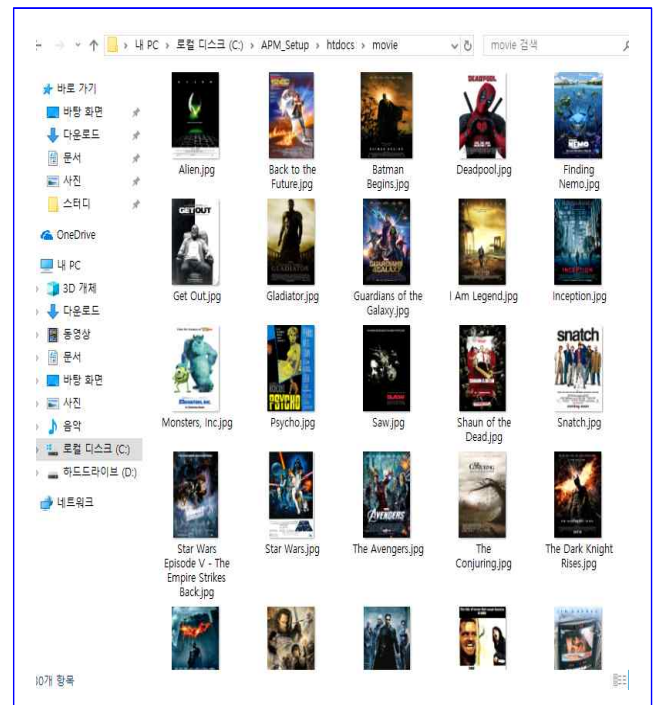
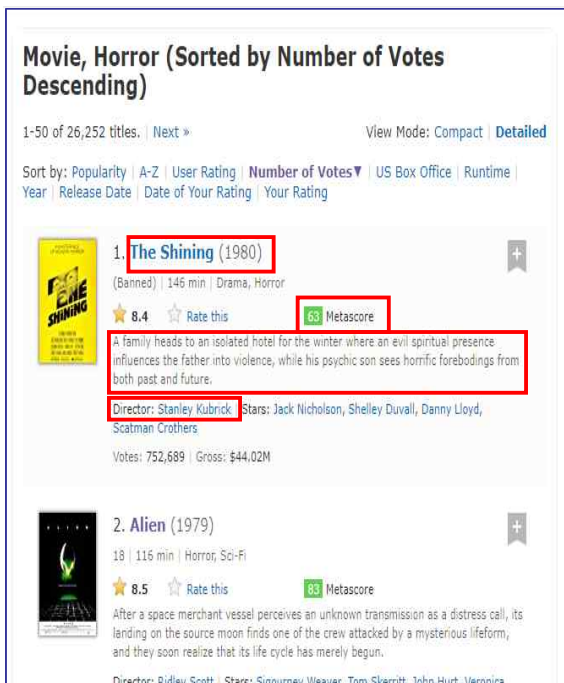
2.1 전체 구성도



2.2 작품 설명

해당 어플리케이션 IMDB 사이트의 액션, 코미디, 공포 영화 Top 10 데이터를 mysql 서버에 저장하여 사용자에게 어플리케이션 통신을 통해 알려주는 어플리케이션입니다.

IMDB 사이트에서 순위를 나타내는 페이지에서 액션, 코미디, 공포 장르를 선택하고 각각페이지에서 Number of Votes 항목(총 평가 수)을 내림차순 하여 해당 영화데이터들의 영어제목, 개봉년도, 감독, 메타스코어, 시놉시스(줄거리)를 각각의 mysql 테이블에 입력하며 한국어 영화 제목은 번역을 통해 직접 삽입하였습니다. 또한 영화포스터의 경우 apache를 통해 접근할 수 있도록 직접 이미지파일로 저장하였습니다.



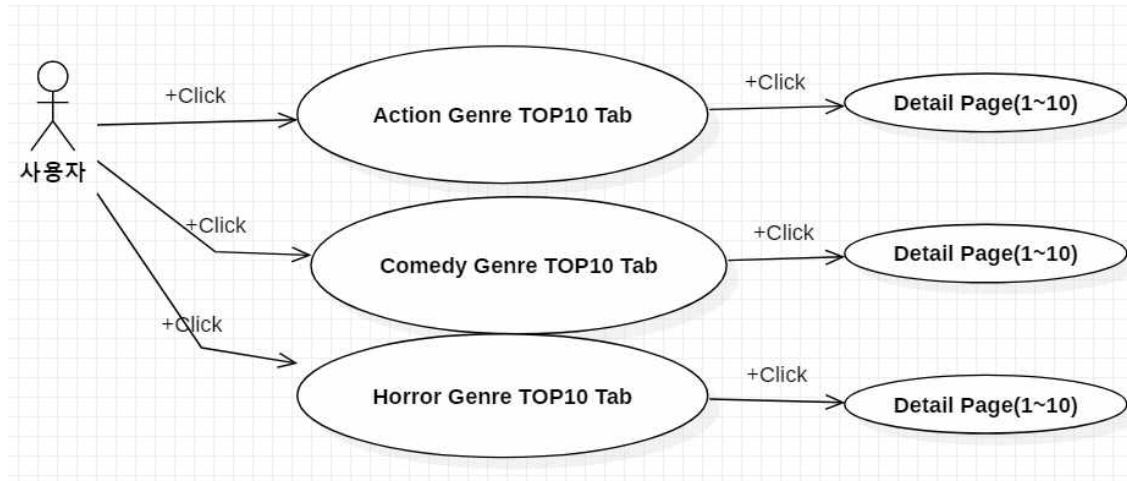
사용자 인터페이스는 네비게이션 드로우와 탭뷰를 통한 환경을 만들고 MainActivity.java에서 탭의 전환 및 네비게이션 항목의 전환이 가능하게 구현을 하였습니다. 그리고 MainActivity.java에서 php의 파일명을 String으로 지정하여 동일한 MovieFragment.java를 3개의 탭모두에게 적용을 하여 php파일마다 (action.php, comedy.php, horror.php) 해당 php를 MovieFragment.java에서 불러와 JSON을 분석한 후 커스텀 리스트형식으로 영화데이터(영어제목, 한국제목, 개봉년도, 메타스코어, 포스터)들을 출력하였습니다.

해당 항목들은 클릭되면 intent를 통해 Item.java가 호출되며 추가적인 영화데이터(영어제목, 영화줄거리, 포스터)들을 보여줍니다. 또한 해당 항목에서 포스터를 누를시 추가적인 intent를 통해 사용자의 웹 브라우저로 영화제목을 구글 검색을 한 화면을 띄워줍니다.

3. 설계

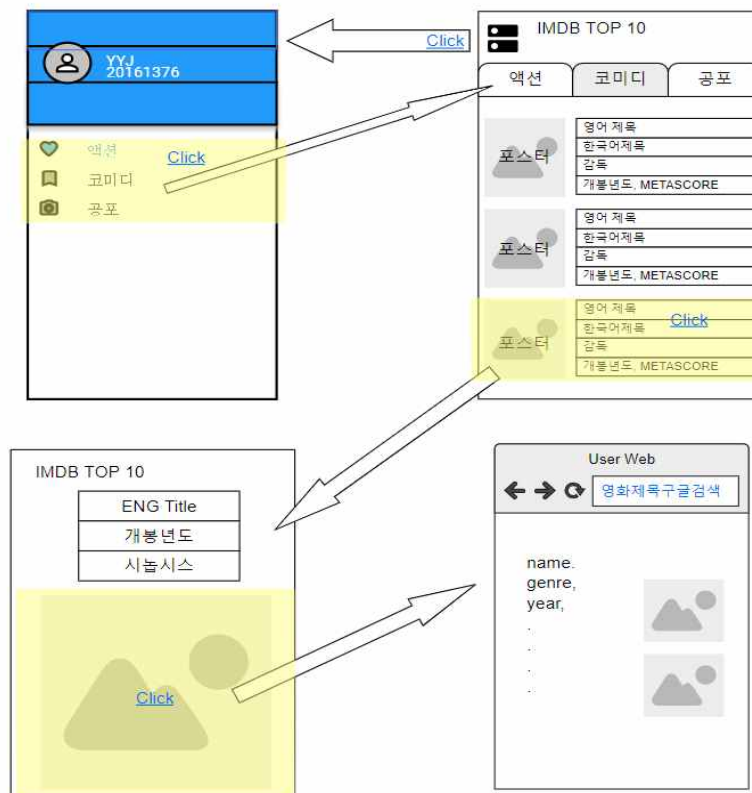
3.1 Use Case

- 사용자가 어플리케이션 클릭 시 어플리케이션 화면 전환



3.2 UI Design

- 어플리케이션 사용자 인터페이스



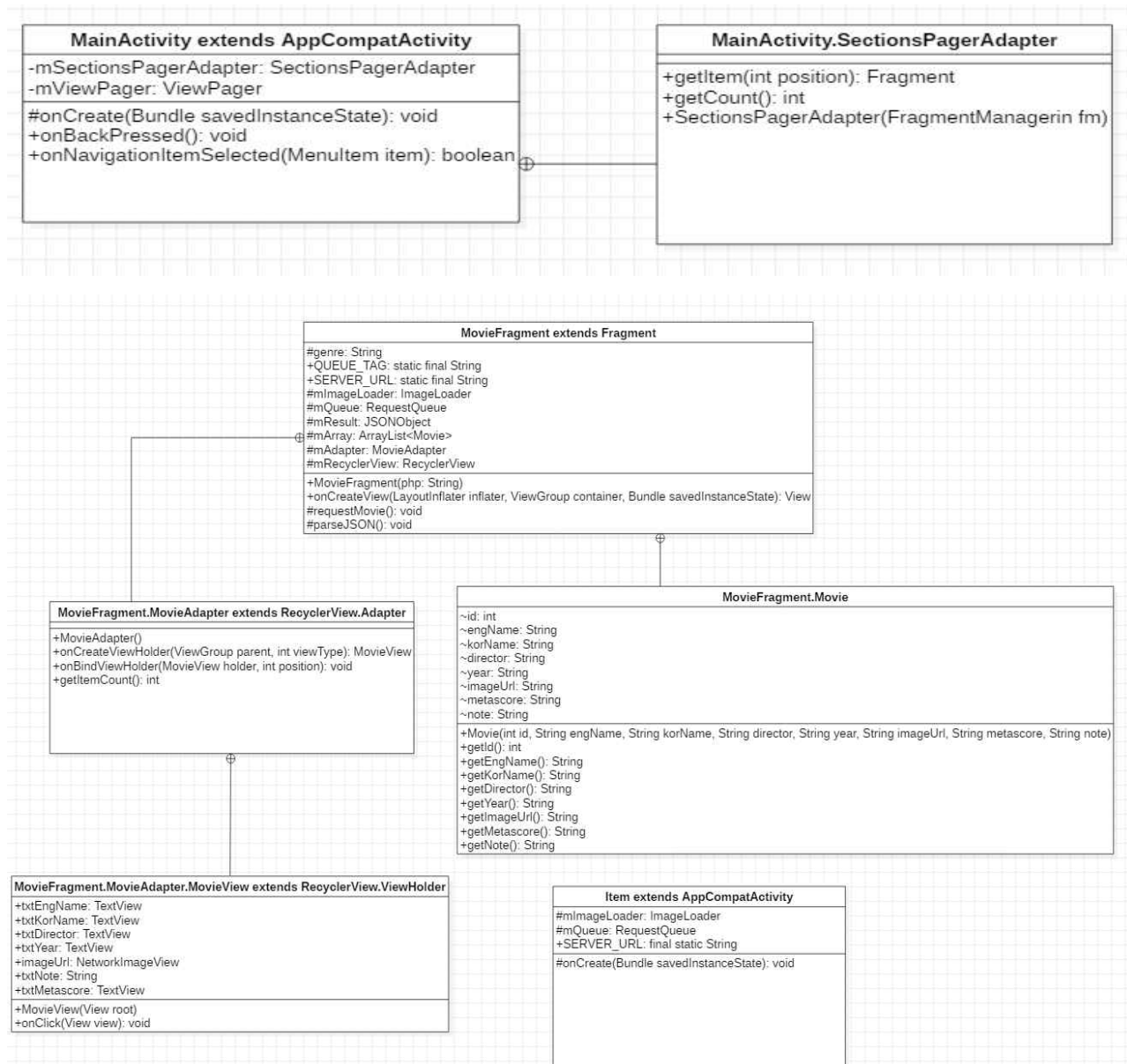
3.3 DB Design

- imdb 데이터베이스의 action, comedy, horror테이블 각 항목들은 모두 id값, 영어제목, 한국어제목, 감독, 개봉년도, 이미지URL, 메타스코어, 시놉시스로 총 8개의 필드를 가짐

imdb.action	imdb.comedy	imdb.horror
id : int(11)	id : int(11)	id : int(11)
engname : varchar(100)	engname : varchar(100)	engname : varchar(100)
korname : varchar(100)	korname : varchar(100)	korname : varchar(100)
director : varchar(50)	director : varchar(50)	director : varchar(50)
year : decimal(4,0)	year : decimal(4,0)	year : decimal(4,0)
imageurl : varchar(200)	imageurl : varchar(200)	imageurl : varchar(200)
metascore : int(11)	metascore : int(11)	metascore : int(11)
note : varchar(255)	note : varchar(255)	note : varchar(255)

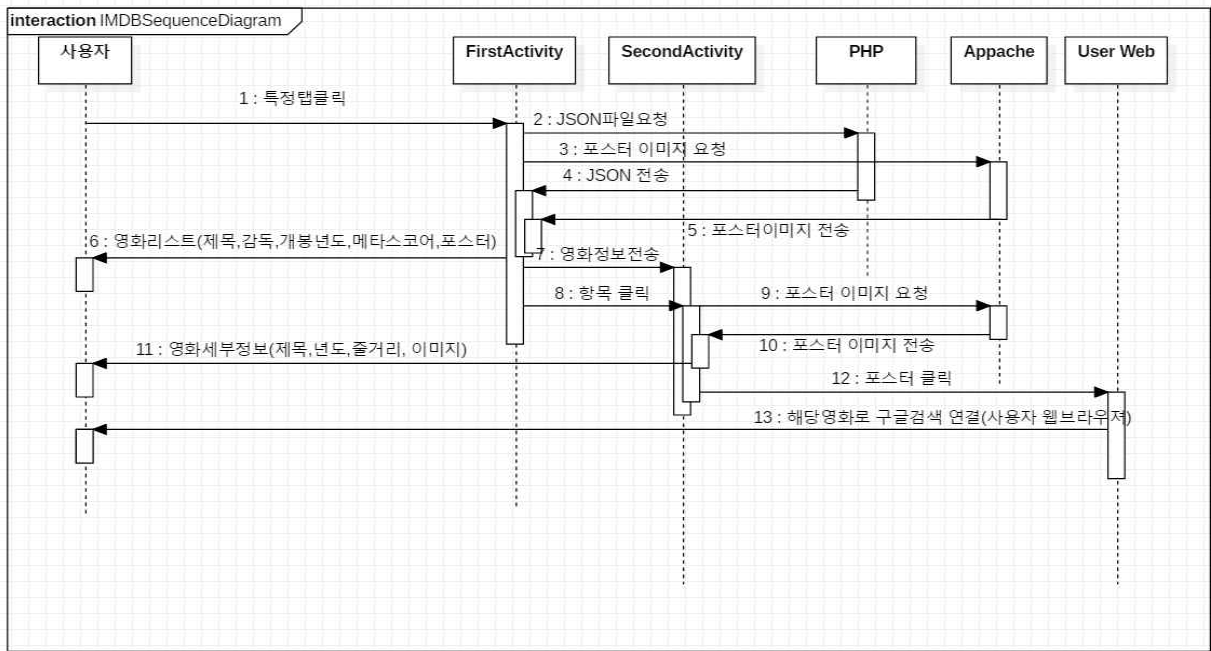
3.4 Class Diagram

- 전체 자바클래스 다이어그램
- MainActivity, MovieFragment, Item 클래스 및 inner클래스들의 속성 및 기능 표현



3.5 Use Case별 Sequence Diagram

-Use Case(어플리케이션 이용)Sequence Diagram을 시간에 따라 사용자가 얻는 정보 및 각 클래스별 요청 데이터들을 나타냄



4. 구현

- 중요 구현 코드 설명

MainActiviy.java

Class : MainActivity

- onCreate()
 - MainActivity.java 실행시 activiy_main.xml 호출함.
 - 네비게이션 드로우 및 탭 레이아웃 생성함.
- onBackPressed()
 - 네비게이션 드로우 호출시 휴대폰의 뒤로 가기 버튼 클릭 시 네비게이션 드로우를 닫아줌
- onNavigationItemSelected()
 - 네비게이션 드로우에서 각 항목을 클릭 시 다른 탭으로 이동시켜줌.

inner Class : MainActivity.SectionsPagerAdatpter

- getItem()
 - 탭클릭시의 항목처리부분.
- getCount()
 - 탭 숫자 반환함.


```

1 public class MainActivity extends AppCompatActivity
2     implements NavigationView.OnNavigationItemSelectedListener {
3
4
5     private SectionsPagerAdapter mSectionsPagerAdapter;
6     private ViewPager mViewPager;
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
13         setSupportActionBar(toolbar);
14
15         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
16         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
17             this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
18         drawer.addDrawerListener(toggle);
19         toggle.syncState();
20
21         NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
22         navigationView.setNavigationItemSelectedListener(this);
23
24         mSectionsPagerAdapter = new SectionsPagerAdapter(getSupportFragmentManager());
25
26         mViewPager = (ViewPager) findViewById(R.id.container);
27         mViewPager.setAdapter(mSectionsPagerAdapter);
28
29         TabLayout tabLayout = (TabLayout) findViewById(R.id.tabs);
30
31         mViewPager.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tabLayout));
32         tabLayout.addTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(mViewPager));
33     }
34
35     @Override
36     public void onBackPressed() {
37         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
38         if (drawer.isDrawerOpen(GravityCompat.START)) {
39             drawer.closeDrawer(GravityCompat.START);
40         } else {
41             super.onBackPressed();
42         }
43     }
44
45     @SuppressWarnings("StatementWithEmptyBody")
46     @Override
47     public boolean onNavigationItemSelectedListener(Menuitem item) {
48         int id = item.getItemId();
49
50         if (id == R.id.nav_action) {
51             mViewPager.setCurrentItem(0);
52         } else if (id == R.id.nav_comedy) {
53             mViewPager.setCurrentItem(1);
54         } else if (id == R.id.nav_horror) {
55             mViewPager.setCurrentItem(2);
56         }
57
58         DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
59         drawer.closeDrawer(GravityCompat.START);
60         return true;
61     }
62
63     public class SectionsPagerAdapter extends FragmentPagerAdapter {
64
65         public SectionsPagerAdapter(FragmentManager fm) {
66             super(fm);
67         }
68
69         @Override

```

```

70     public Fragment getItem(int position) {
71         if (position == 0)
72             return new MovieFragment("action.php");
73         else if (position == 1)
74             return new MovieFragment("comedy.php");
75         else
76             return new MovieFragment("horror.php");
77     }
78
79     @Override
80     public int getCount() {
81         return 3;
82     }
83 }
84 }

```

MovieFragment.java

Class: MovieFragment

- 해당클래스는 기본적으로 매개변수를 가지는 생성자는 컴파일 경고가 뜨기에 @SuppressWarnings("ValidFragment") 어노테이션을 추가하여 구현함. [1]

inner Class : MovieFragment.Movie

- 영화정보에 대한 id, 영어제목, 한국어제목, 감독, 개봉년도, 이미지소스URL, 메타스코어, 줄거리 정보를 생성자 및 getter()함수로 초기화 및 반환함.

- onCreateView()
 - Fragment에 실제 사용할 뷰를 생성.
 - Volley를 통한 네트워크 통신.
- requestMovie()
 - php파일을 불러와 parseJSON()에 연결함.
- parseJSON()
 - JSON을 분석하여 각 항목들을 저장함.

inner Class : MovieFragment.MovieAdapter

- RecyclerView를 통한 View 의 재활용 ViewHolder클래스를 통해 View inflate, 뷰가 존재하는 경우에는 불필요한 과정 없이 아이템을 보여줌.

- onClick()
 - 각 리스트 항목이 클릭되면 해당 리스트정보를 item.class로 intent 및 putExtra를 활용하여 item.class를 실행함.

```

1  public class MovieFragment extends Fragment {
2      protected String genre;
3      public MovieFragment() { }
4      @SuppressWarnings("ValidFragment")
5      public MovieFragment(String php) {
6          this.genre = php;
7      }
8      public class Movie {
9          int id;
10         String engName;
11         String korName;

```

```

12     String director;
13     String year;
14     String imageUrl;
15     String metascore;
16     String note;
17     public Movie(int id, String engName, String korName, String director,
18 String year, String imageUrl, String metascore, String note) {
19         this.id = id;
20         this.engName = engName;
21         this.korName = korName;
22         this.director = director;
23         this.year = year;
24         this.imageUrl = imageUrl;
25         this.metascore = metascore;
26         this.note = note;
27     }
28     public int getId() { return id; }
29     public String getEngName() { return engName; }
30     public String getKorName() { return korName; }
31     public String getDirector() { return director; }
32     public String getYear() { return year; }
33     public String getImageUrl() { return imageUrl; }
34     public String getMetascore() { return metascore; }
35     public String getNote() { return note; }
36 }
37 public static final String QUEUE_TAG = "ActionRequest";
38 public static final String SERVER_URL = "http://125.185.23.23/";
39 protected ImageLoader mImageLoader = null;
40 protected RequestQueue mQueue = null;
41 protected JSONObject mResult = null;
42 protected ArrayList<Movie> mArray = new ArrayList<Movie>();
43 protected MovieAdapter mAdapter = new MovieAdapter();
44 protected RecyclerView mRecyclerView = null;
45 @Override
46 public View onCreateView(LayoutInflater inflater, ViewGroup container,
47 Bundle savedInstanceState) {
48     View rootView = inflater.inflate(R.layout.fragment_main, container, false);
49     mRecyclerView = (RecyclerView) rootView.findViewById(R.id.recyclerView);
50     mRecyclerView.setAdapter(mAdapter);
51     mRecyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
52     mRecyclerView.setItemAnimator(new DefaultItemAnimator());
53
54     CookieHandler.setDefault(new CookieManager());
55     mQueue = Volley.newRequestQueue(getContext());
56     mImageLoader = new ImageLoader(mQueue, new ImageLoader.ImageCache() {
57         private final LruCache<String, Bitmap> cache = new LruCache<String, Bitmap>(20);
58         @Override
59         public Bitmap getBitmap(String url) {
60             return cache.get(url);
61         }
62         @Override
63         public void putBitmap(String url, Bitmap bitmap) {
64             cache.put(url, bitmap);
65         }
66     });
67     requestMovie();
68     return rootView;
69 }
70 protected void requestMovie() {
71     String url = SERVER_URL + genre;
72     JsonObjectRequest request =
73         new JsonObjectRequest(Request.Method.GET, url, null,
74             new Response.Listener<JSONObject>() {
75                 @Override
76                 public void onResponse(JSONObject response) {
77                     Log.i("result", response.toString());
78                     mResult = response;
79                     parseJSON();
80                 }

```

```

81         },
82         new Response.ErrorListener() {
83             @Override
84             public void onErrorResponse(VolleyError error) {
85                 Toast.makeText(getContext(), error.getMessage(), Toast.LENGTH_LONG).show();
86                 Log.i("result", error.toString());
87             }
88         });
89     request.setTag(Queue.TAG);
90     mQueue.add(request);
91 }
92 protected void parseJSON() {
93     mArray.clear();
94     try {
95         JSONArray array = mResult.getJSONArray("list");
96         for (int i = 0; i < array.length(); i++) {
97             JSONObject item = (JSONObject) array.get(i);
98             int id = item.getInt("id");
99             String engName = item.getString("engname");
100             String korName = item.getString("korname");
101             String director = item.getString("director");
102             String year = item.getString("year");
103             String imageUrl = item.getString("imageUrl");
104             String metascore = item.getString("metascore");
105             String note = item.getString("note");
106             mArray.add(new Movie(id, engName, korName, director, year, imageUrl, metascore, note));
107         }
108     } catch (JSONException | NullPointerException e) {
109         Toast.makeText(getContext(), "Error " + e.toString(), Toast.LENGTH_LONG).show();
110         mResult = null;
111     }
112     mAdapter.notifyDataSetChanged();
113 }
114 public class MovieAdapter extends RecyclerView.Adapter<MovieAdapter.MovieView> {
115     public class MovieView extends RecyclerView.ViewHolder implements View.OnClickListener {
116         public TextView txtEngName;
117         public TextView txtKorName;
118         public TextView txtDirector;
119         public TextView txtYear;
120         public NetworkImageView imageUrl;
121         public String txtNote;
122         public TextView txtMetascore;
123         public MovieView(View root) {
124             super(root);
125             root.setOnClickListener(this);
126             txtEngName = (TextView) root.findViewById(R.id.engName);
127             txtKorName = (TextView) root.findViewById(R.id.korName);
128             txtYear = (TextView) root.findViewById(R.id.year);
129             txtDirector = (TextView) root.findViewById(R.id.dir);
130             txtMetascore = (TextView) root.findViewById(R.id.metaScore);
131             imageUrl = (NetworkImageView) root.findViewById(R.id.imageUrl);
132         }
133         @Override
134         public void onClick(View view) {
135             Intent result = new Intent(getContext(), Item.class);
136             String movie = txtEngName.getText().toString();
137             String note = txtNote;
138             String year = txtYear.getText().toString();
139             result.putExtra("movie", movie);
140             result.putExtra("year", year);
141             result.putExtra("note", note);
142             startActivity(result);
143         }
144     }
145     public MovieAdapter() { }
146     @Override
147     public MovieView onCreateViewHolder(ViewGroup parent, int viewType) {
148         View root = LayoutInflater.from(parent.getContext()).inflate(R.layout.movie_item, parent, false);
149         return new MovieView(root);

```

```

150     }
151     @Override
152     public void onBindViewHolder(MovieView holder, int position) {
153         Movie g = mArray.get(position);
154         holder.txtEngName.setText(g.getEngName());
155         holder.txtKorName.setText(g.getKorName());
156         holder.imageUrl.setImageUrl(SERVER_URL + g.getImageUrl(), mImageLoader);
157         holder.txtDirector.setText("감독: " + g.getDirector());
158         holder.txtYear.setText("개봉년도: " + g.getYear());
159         holder.txtMetascore.setText("메타스코어: " + g.getMetascore());
160         holder.txtNote = g.getNote();
161     }
162
163     @Override
164     public int getItemCount() {
165         return mArray.size();
166     }
167 }
168 }

```

item.java

Class: Item

- onCreate()
 - intent를 통해 전달받은 영화이름, 개봉년도, 줄거리를 activity_item의 해당항목들로 정보를 보여줌
 - 영화 포스터는 영화이름을 파일명으로 하여 NetworkImage뷰를 활용해 출력함.
- onClick()
 - 포스터 이미지 클릭 시 intent를 통해 사용자 웹브라우저로 영화제목을 구글검색하는 웹을 띄워줌.

```

1 public class Item extends AppCompatActivity {
2     protected ImageLoader mImageLoader = null;
3     protected RequestQueue mQueue = null;
4     public static final String SERVER_URL = "http://125.185.23.23/movie/";
5     @Override
6     protected void onCreate(Bundle savedInstanceState) {
7         super.onCreate(savedInstanceState);
8         setContentView(R.layout.activity_item);
9
10        Intent intent = getIntent();
11        TextView movie = (TextView)findViewById(R.id.movie);
12        movie.setText(intent.getStringExtra("movie"));
13        TextView year = (TextView)findViewById(R.id.date);
14        year.setText(intent.getStringExtra("year"));
15        TextView note = (TextView)findViewById(R.id.note);
16        note.setText(intent.getStringExtra("note"));
17
18        String temp = intent.getStringExtra("movie");
19        final String query = temp;
20        temp = temp+".jpg";
21        mQueue = Volley.newRequestQueue(this);
22        mImageLoader = new ImageLoader(mQueue,
23            new ImageLoader.ImageCache() {
24                private final LruCache<String, Bitmap>
25                    cache = new LruCache<String, Bitmap>(20);
26                @Override
27                public Bitmap getBitmap(String url) {
28                    return cache.get(url);
29                }
30                @Override
31                public void putBitmap(String url, Bitmap bitmap) {
32                    cache.put(url, bitmap);
33                }
34            });
35        NetworkImageView image = findViewById(R.id.img);

```

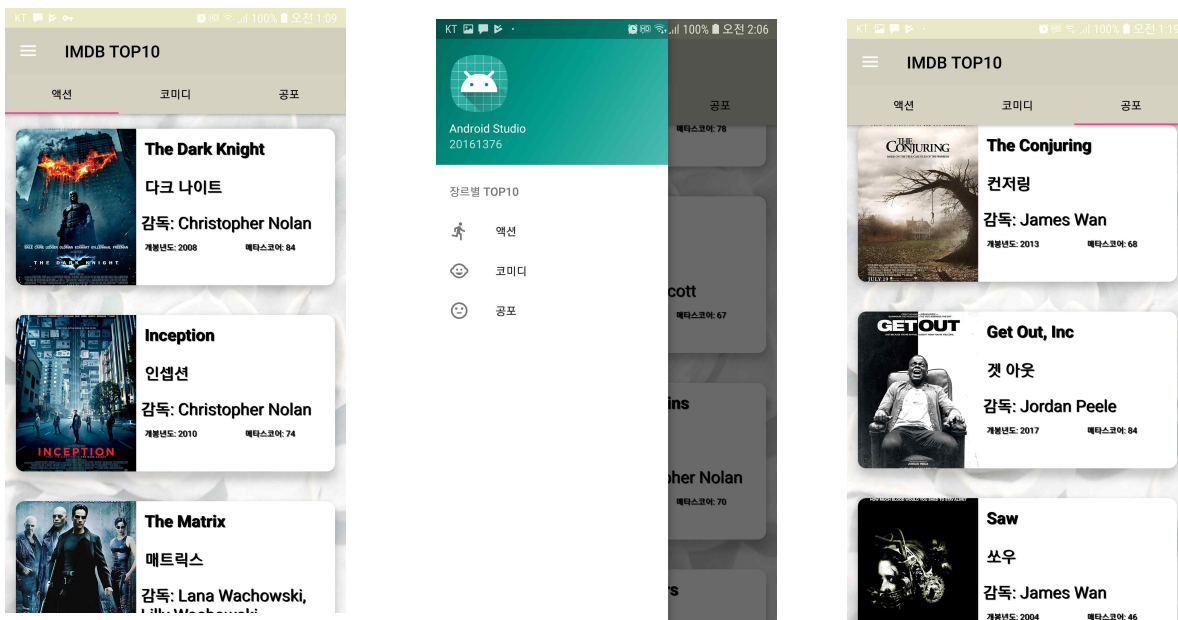
```

36         image.setImageUrl(SERVER_URL + temp, mImageLoader);
37
38         image.setOnClickListener(new View.OnClickListener(){
39             @Override
40             public void onClick(View v) {
41                 String link = "https://www.google.co.kr/search?q="+ query;
42                 Uri uri = Uri.parse(link);
43                 Intent intent = new Intent(Intent.ACTION_VIEW,uri);
44                 startActivity(intent);
45             }
46         });
47     }
48 }

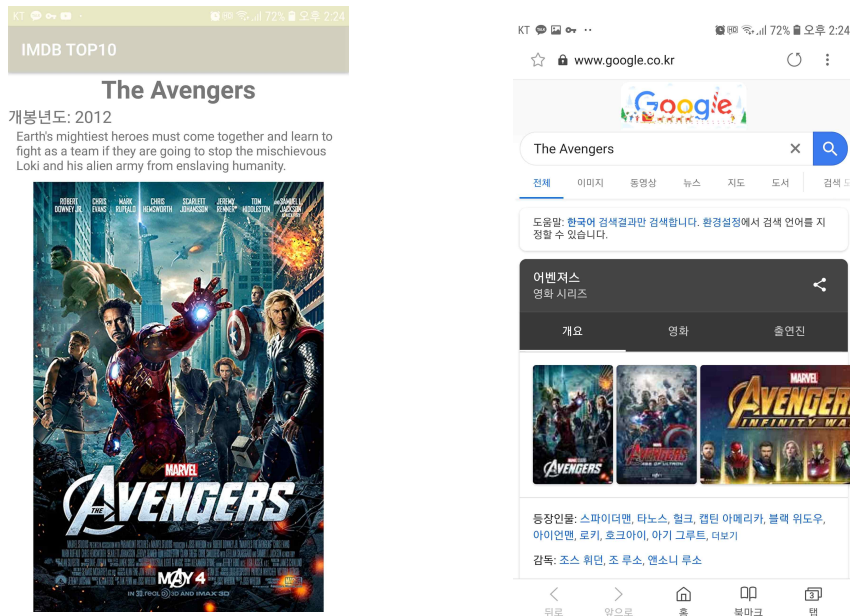
```

어플리케이션 사용법

사용자 초기화면에는 액션영화 장르에 대한 IMDB사이트의 유저 평점 순 Top10 항목을 보여줍니다. 사용자가 다른 탭을 클릭하거나 좌측 및 우측 슬라이드, 또한 좌측상단 𠂇 버튼을 클릭 시 다른 탭으로 넘어가는 슬라이드 메뉴 버튼을 생성하여 클릭 시 코미디 및 공포 영화에 대한 IMDB사이트의 유저 평점 순 Top10 항목을 보여줍니다. 각 항목들은 리스트 화 되어있어 화면슬라이드를 통해 10개의 항목을 보여줍니다.



사용자가 해당 리스트항목을 클릭하면 영화 제목, 개봉년도, 포스터와 추가적인 정보인 영화줄거리를 새로운 화면을 통해 보여줍니다. 또한 해당화면에서 영화 포스터를 클릭 시 영화제목을 토대로 구글검색을 한 사용자 인터넷브라우저 결과를 보여줍니다.



5. 결 론

현재 단순히 데이터베이스에 있는 내용들을 사용자에게 보여주는 측면에서는 구현이 잘되어있다. 현재까지 데이터가 많지 않아 수작업으로 데이터 삽입이 가능 했지만 추가적인 Top100, Top1000을 활용하려면 저번 과제의 크롤링을 활용해야한다.

추가적으로 사용자의 선택들 또한 새롭게 저장한후 데이터들이 누적된다면 '왓챠'어플리케이션처럼 사용자에게 더 기호에 맞는 추천을 가능하게 할 수 있을 것이다.

6. 자체 분석과 평가

문제 및 해결방법

- 어플리케이션을 설계시 탭뷰 부분에서 각 탭마다 동일한 필드의 테이블을 다른탭마다 구현이 목표였으며 해당 방식의 구현은 여러 Fragment를 생성하지않고 하나의 Fragment에서 생성자를 통해 php 파일이름의 String을 인자전달하여 구현을 계획하였다. 해당부분에서 컴파일 경고가 지속적으로 되어 추가적인 어노테이션을 삽입하여 구현함. [1]
- MovieFragment에서 item클래스로 memo항목을 textview를 통한 intent전달은 구현이 되지않아 해당 변수 ViewHolder는 String으로하여 MovieView에서 별도의 findViewById를 하지않고 데이터만 전달하는 방식으로 구현함.
- 또한 NetworkImageView를 Bitmap으로 바꾸어 putExtra를 하였는데 받는 클래스에서(Item.java) 지속적인 문제가 생겨 Item.java에서도 Volley를 통해 appace에있는 포스터이미지를 다시한번 접근하여 불러옴.

7. 부록

참고문헌 및 출처

[1] Avoid non-default constructors in fragments 해결방법 : <http://tony2012.tistory.com/27>

IMDB Movie, Horror (Sorted by Number of Votes Descending) :

https://www.imdb.com/search/title?title_type=movie&genres=horror&sort=num_votes,desc&explore=title_type,genres

금오공대 LMS 공개강좌, 안드로이드 심화 프로그래밍 (3) 탭뷰, (4) 슬라이드 메뉴 - 오병우

ui디자인 툴 : <https://app.moqups.com/unsaved/642e9156/edit/page/ad64222d5>