

Mobile App Programming

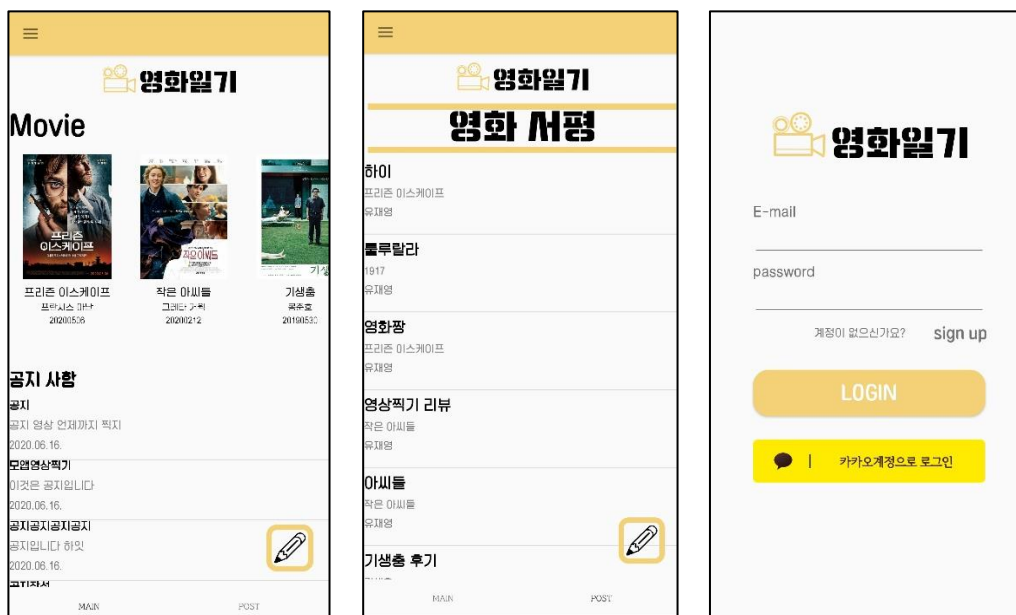
Personal Project Report – [영화일기]

2016311981 유재영

1. Overview

이번 모바일 앱 프로그래밍 프로젝트의 최종 주제는 영화에 대한 서평 및 리뷰를 적을 수 있는 어플리케이션입니다. 영화를 보는 것을 좋아하고 예전에는 서평을 블로그에 올리는 일을 했어서 이러한 앱이 있으면 유용하고 쓰고 싶다고 생각했습니다. 그래서 이번 기회를 통해 영화에 대한 감상을 글로 남길 수 있는 앱을 만들었습니다. Personal Assignment에서는 사진이 위주가 되는 SNS를 제작해보았기 때문에 적절한 주제가 될 것이라 판단했습니다. 어플리케이션은 총 6개의 Activities와 2개의 fragments 외 자바파일들로 이루어져 있고 모든 activity에 대해서 landscape가 가능하도록 제작되었습니다. Back-end 서버로는 수업시간에 배웠던 Firebase와 Storage를 이용했고, Open API로는 카카오 계정으로 연동하는 기능을 이용했습니다.

2. Application Design



어플리케이션의 메인 화면입니다. 기존 PA의 큰 틀과 비슷하지만 내부의 콘텐츠 구성과 디자인을 신경 써서 구현했습니다. 이미지를 drawable 폴더에 넣어 Image View를 이용해 로고와 글

쓸 수 있는 버튼을 꾸며주었습니다. 기본 색으로 #f4d177으로 정했고 이 색은 res/values의 colors.xml에 선언해 이용했습니다. 어플리케이션에 이용된 font들도 res/font 파일에 넣어서 font.xml에 선언해 이용했습니다.

3. Firebase Database, Storage

백엔드 서버로는 수업에 이용했던 Firebase console을 이용했습니다. 실시간 데이터베이스를 이용했으며, 큰 폴더로는 사용자의 일반 계정을 담아둔 "user_list", 카카오 계정으로 로그인 한 "user_list_kakao", 관리자의 공지 데이터를 저장하는 "announce", 영화 정보 담아둔 "movie", 서평을 모아둔 "Post"가 있습니다. 각각의 폴더에 각 개체에 대해 unique 키를 부여했습니다.(push) 예외적으로 영화에 대한 primary key는 영화의 제목으로 설정했는데, 실제로 같은 이름의 영화도 존재하므로 이 부분도 추후 unique 키를 부여하는 방식으로 수정되어야 합니다. 해당 프로젝트에서는 영화의 이름을 이용했습니다.

4. Open API – Kakao 계정 로그인

일반 계정 외에 카카오 계정으로 로그인이 가능하도록 Kakao Developers의 카카오 로그인 문서를 이용했습니다. 그러면서 일반 계정과 어떻게 구분을 하고 카카오 계정을 어떻게 Firebase에 저장할 지 고민했습니다. Firebase의 auth를 이용하는 방법이 있었지만 node.js 기반의 서버가 필요했고, 시간의 제약이 상당한 입장에서, 또 node.js에 대해 잘 알지 못하는 상태에서 다른 방법을 이용할 수밖에 없었습니다. Login Activity에서 Login 버튼을 눌러 로그인 하는 것과 카카오 로그인 버튼을 구분하고, 두 종류의 계정에 대해 database의 폴더를 구분했습니다. 일반 계정의 경우 sign up에서 넘어온 정보를 토대로 Sing up Activity에서 계정이 등록되어 로그인할 수 있도록 했고, 카카오 계정의 경우 Main Activity로 넘어가게 되면서 "user_list_kakao"에 해당 카카오 계정이 있는지 확인하고 없는 경우 사용자의 정보를 Firebase에 저장하도록 하였습니다. 따라서 카카오 계정의 Session에 대한 코드는 MainPageActivity.java에 존재하게 됩니다.



[일반 계정: user_list, 카카오 계정: user_list_kakao]

5. Implementation

- Signup Activity

The image shows two wireframe designs for the Signup Activity. The left design is a vertical form with the following elements: a title 'Sign Up' with a yellow icon, a label 'User ID(Email)' above a text input field, a label 'Password' above a text input field, a label 'Full Name' above a text input field, a label 'Birthday' above a text input field, and a yellow 'SIGN UP' button at the bottom. The right design is a horizontal form with the following elements: a title 'Sign Up' with a yellow icon, two input fields for 'User ID(Email)' and 'Full Name' in the top row, two input fields for 'Password' and 'Birthday' in the bottom row, and a yellow 'SIGN UP' button at the bottom center.

Signup Activity로 계정을 등록할 수 있습니다. 빈칸이 존재하거나 user ID(Email)이 이미 존재한다면 계정을 등록할 수 없습니다. FirebaseUserInfo.java의 객체를 이용해 Database에 계정을 등록합니다. 계정을 등록하게 되면 User ID(Email)에 입력했던 정보가 Login Page로 넘어가게 됩니다.

- Login Activity

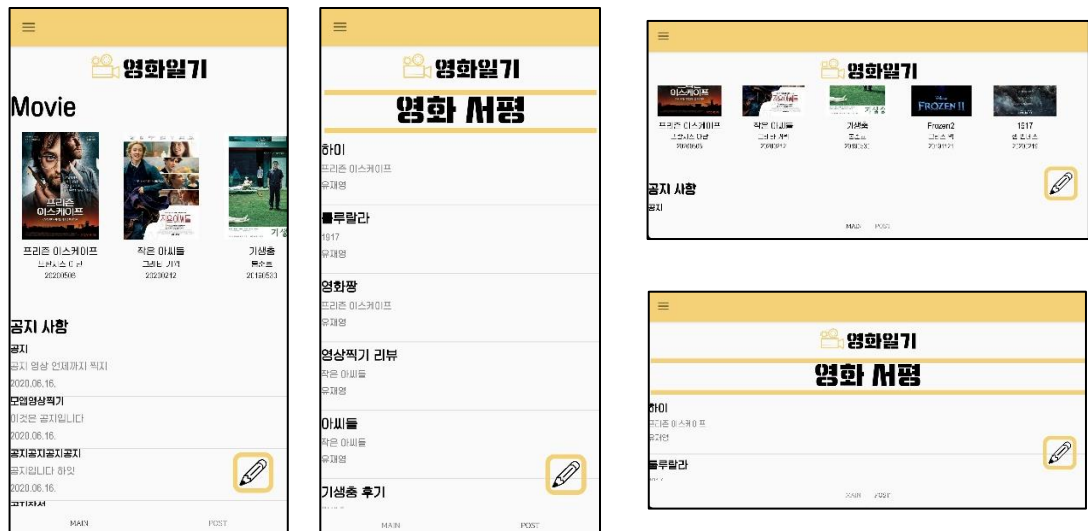
The image shows two wireframe designs for the Login Activity. The left design is a vertical form with the following elements: an app logo '영화일기' with a yellow icon, a label 'E-mail' above a text input field, a label 'password' above a text input field, a link '계정이 없으신가요? sign up', a yellow 'LOGIN' button, and a yellow button with a speech bubble icon and text '카카오계정으로 로그인'. The right design is a horizontal form with the following elements: an app logo '영화일기' with a yellow icon, two input fields for 'E-mail' and 'password' on the left, a link '계정이 없으신가요? sign up' on the right, a yellow 'LOGIN' button, and a yellow button with a speech bubble icon and text '카카오계정으로 로그인'.

Login Activity로 이메일(ID)가 존재하지 않으면 "등록된 계정이 없습니다.", 비밀번호가 틀렸을 경우 "비밀번호가 틀렸습니다."라는 Toast 메시지가 뜹니다. 계정을 입력하는 EditText의 input type은 textEmailAddress입니다.

- Admin Activity

Login 페이지에서 admin/admin을 입력하면 들어올 수 있는 Activity입니다. 해당 activity에서 Main Page의 공지사항에 등록되는 공지를 작성할 수 있습니다. 제목과 본문을 모두 작성해야 등록이 가능하며, 등록버튼을 누른다고 Activity가 이동하지 않고 뒤로가기 버튼을 눌러야 Login Page로 돌아가게 됩니다.

- MainPage Activity



Main Page Activity에는 2개의 Fragment(Main Fragment, Post Fragment)와 Navigation drawer, toolbar가 구현되어 있습니다. Tab을 앱의 하단 부분에 배치했습니다.

Main Fragment에는 두 종류의 View가 존재합니다. Movie의 정보(포스터, 영화 이름, 감독, 개봉날짜)를 보여주는 부분은 RecyclerView와 RecyclerView Adapter로 구현했고 가로로 스크롤이

됩니다. MovieInfo.java의 객체를 이용해 영화의 정보를 불러옵니다. MovieInfo.java의 객체를 이용해 정보를 불러오게 되고, Storage에는 영화이름.jpg의 포스터 이미지를 불러옵니다.

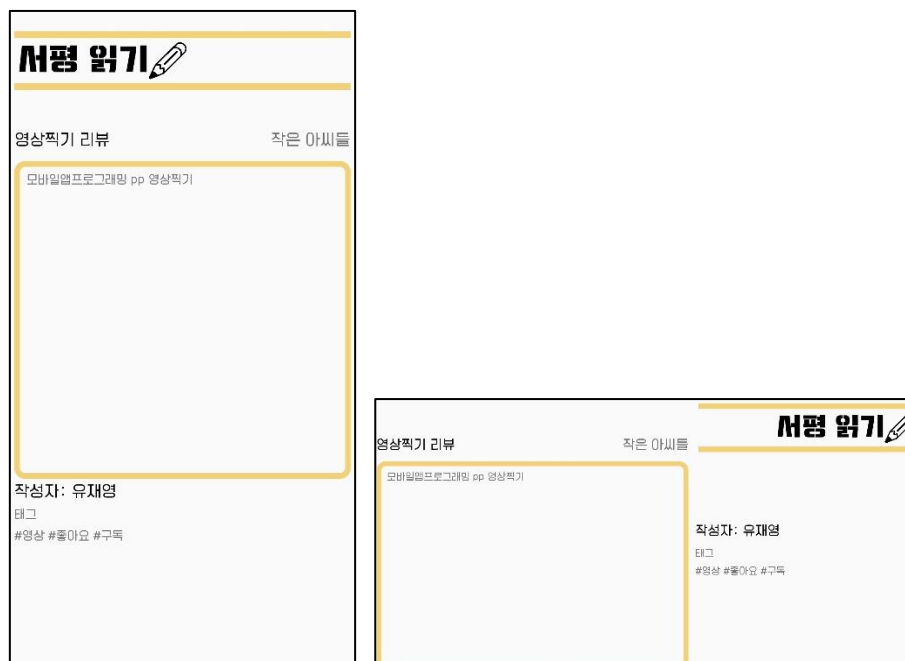
공지사항에 보여지는 부분은 Announce Activity에서 등록되었던 공지가 불러오며 ListView와 Base Adapter를 이용해 구현했습니다. 최신 공지인 경우 가장 상단에 뜨도록 구현되어 있습니다. Firebase에서 공지를 등록하거나 불러올 땐 AnnounceInfo.java의 객체를 이용합니다.

Main Fragment는 Activity들과 다르게 layout-land를 이용하지 않고 AndroidManifest.xml 파일에서 [android:configChanges="orientation|screenSize"]를 이용했습니다. Main Fragment의 공지사항의 경우 가로 모드일 때 화면이 보이지 않게 되는데 이 때 스크롤을 하면 볼 수 있습니다.

Post Fragment의 경우 작성된 서평이 ListView로 구현되어 보여집니다. 해당 ListView도 최근에 쓰인 게시글이 가장 상단에 위치하게 되며, 서평을 누르면 글에 대한 정보가 ReadPost Activity로 넘어가게 됩니다. PostContentInfo.java의 객체를 통해 데이터를 읽어옵니다. Post Fragment의 ListView에는 서평의 제목, 영화의 종류, 작성자가 보이게 됩니다.

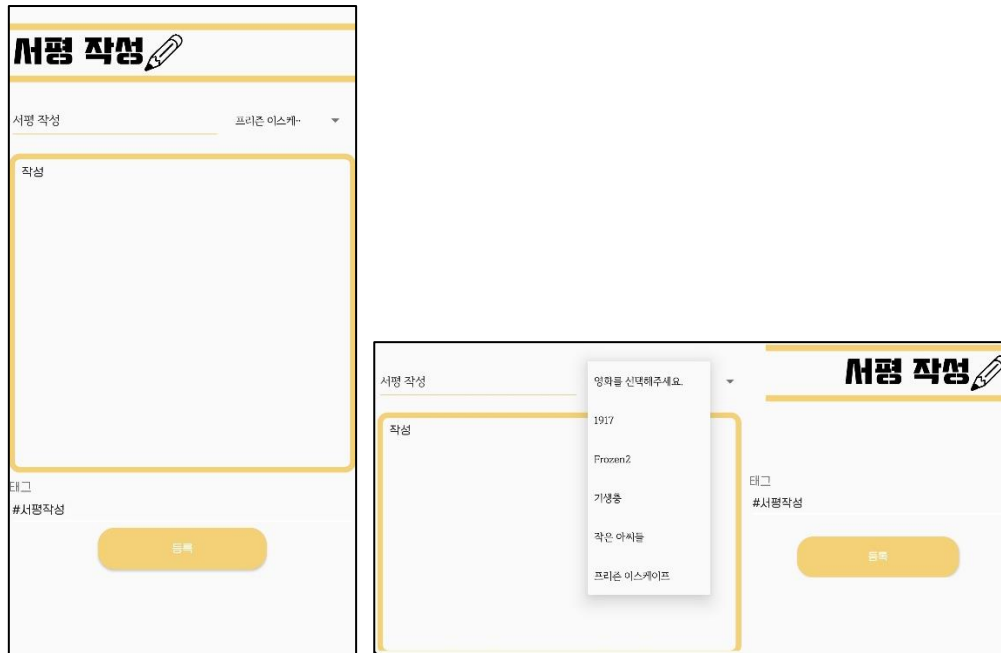
글쓰기 버튼(연필모양의 버튼)을 누를 때, tab정보가 intent에 포함되어 넘어가기 때문에 글을 작성하고 돌아올 때 이동 직전의 fragment로 돌아오게 됩니다.

- ReadPost Activity



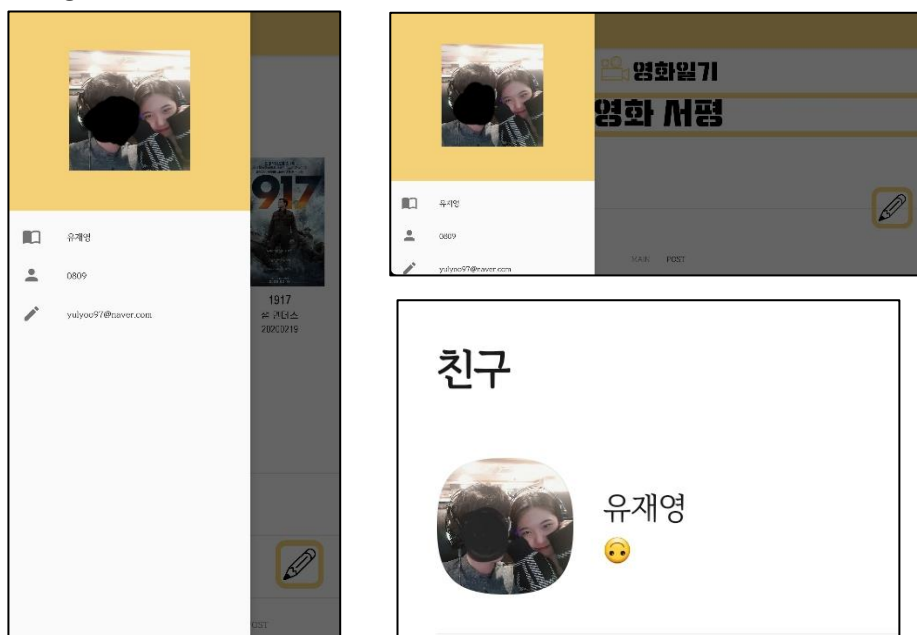
MainPage Activity의 Post Fragment에서 서평을 클릭할 때 뜨는 Activity입니다. 작성되어 있는 서평의 제목, 영화 종류, 내용, 작성자, tag의 정보가 보여집니다. 이 Activity에는 navigation drawer는 따로 없고 버튼도 따로 없습니다.

- WritePost Activity



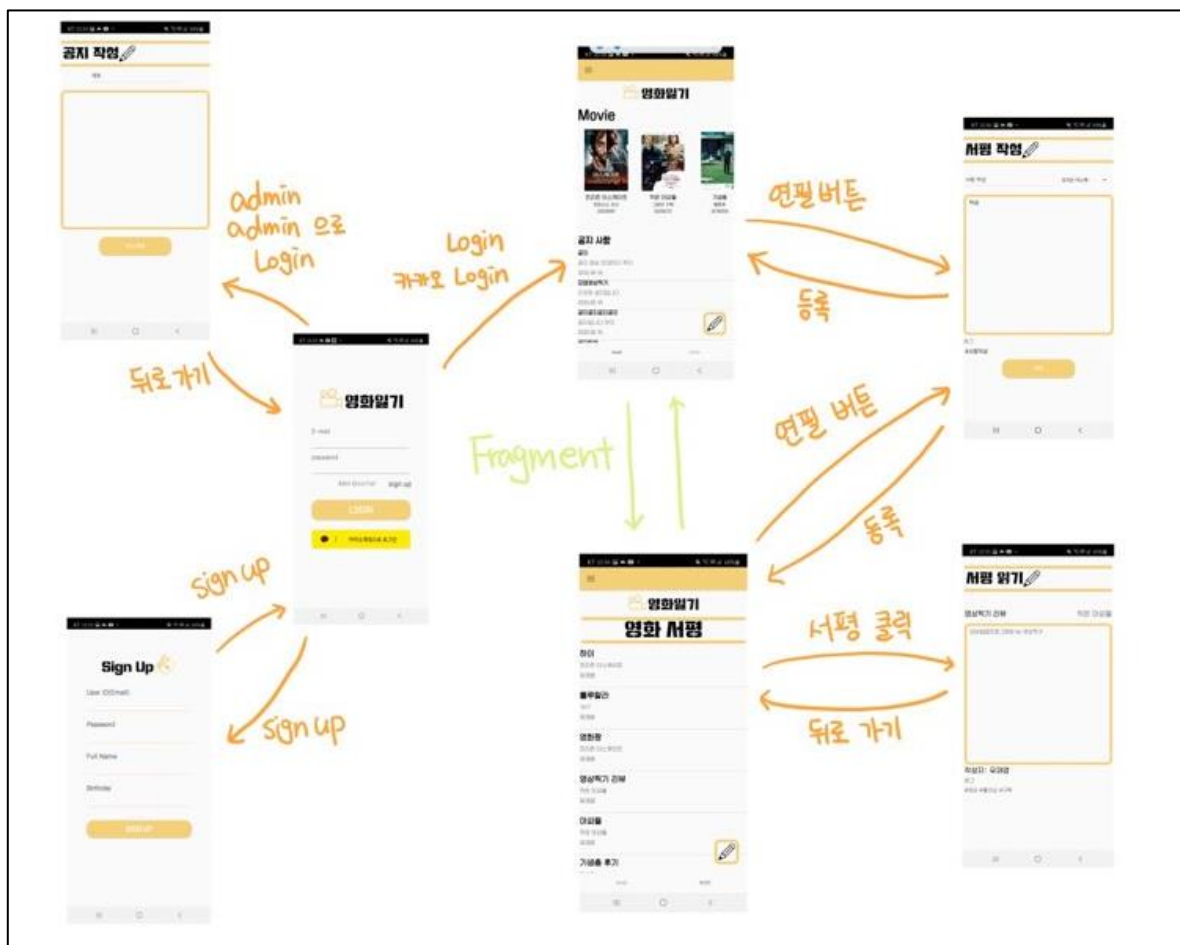
서평을 작성할 수 있는 Activity입니다. MainPage Activity에서 tab정보를 받아와서 작성이 끝나고 글이 등록되면 넘어오기 직전의 tab page로 돌아가게 됩니다. Spinner를 이용해 영화를 선택할 수 있도록 구현했고, 영화의 정보는 Firebase Database의 "movie"폴더에 저장되어 있는 정보를 가져와 띄우게 됩니다. 영화를 선택하지 않거나 제목 및 내용을 작성하지 않으면 게시글을 등록할 수 없습니다.

- Navigation View



Navigation View의 경우 Mainpage Activity와 WritePost Activity에 구현되어 있습니다. 가로모드에서도 보여지며, 해당 이미지의 예시는 카카오 계정으로 로그인 했을 때 뜨는 화면 입니다. 실제 카카오 계정의 이미지와 계정 이메일, 이름, 생일을 그대로 가져오게 됩니다. 일반 계정으로 등록했을 때는 Firebase에서 사용자의 이메일 계정과 비밀번호를 통해 사용자 정보를 가져오게 되며 등록된 profile image가 존재할 경우 Storage를 통해 이미지도 같이 뜨게 됩니다.

- 구동 방식



6. Challenge & Creativity

- Challenge

기존에 의도했던 도전은 List View에 새로운 게시물이 추가될 때 가장 위에 오도록 하는 것이었는데 생각보다 간단한 방법으로 해결되었습니다. Database에서 정보를 불러와 ArrayList<객체>에 저장할 때 index를 0으로 해서 추가해주면 해결이 되었습니다.

그래서 Challenge 라는 이름에 맞게 개인적으로 궁금했던 의문점에 대해 도전했습니다.

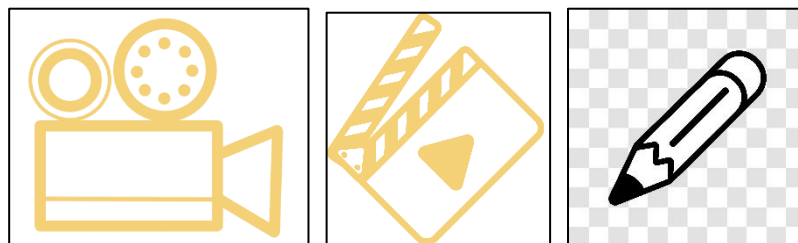
Fragment를 다루는 부분에 대해 지난 과제나 이번 Personal Project에서도 어려움을 꽤 겪었습니다. List View에서는 Image와 함께 item layout을 만들어 적용하는 것이 불가능했습니다. 그래서 Recycler View와 Recycler Adapter를 이용해 item을 View에 추가할 수 있었습니다. Main Fragment의 영화 정보는 Recycler View에 가로 스크롤을 적용해 구현할 수 있었습니다. 또한 한 Fragment에 다양한 종류의 View가 들어갈 수 있는지 궁금해 Main Fragment에 Recycler View와 List View를 동시에 적용될 수 있도록 구현해보았습니다. 각각의 View에 맞는 Adapter를 적용하여 해결할 수 있었습니다.

그리고 의도치 않게 해결하는데 애를 먹었던 도전이 있습니다. Activity들은 layout-land를 통해 landscape 화면에 대해 xml을 적용할 수 있었는데 Fragment는 가로모드가 될 때 Activity에 적용되지 않는 문제가 있었습니다. 이에 대한 해결방안으로 위에서 언급 했듯이, AndroidManifest.xml파일에서 [android:configChanges="orientation|screenSize"]를 이용했습니다. 가로모드로 인해 잘린 Main Fragment의 화면은 Scroll View를 적용하여 스크롤을 통해 접근이 가능하도록 했습니다.

- Creativity

이 어플리케이션의 큰 틀은 PA과제들에 맞춰져 있습니다. 이에 구별이 될 만한 기능이 추가되어야 했고 저는 Admin을 이용해 관리자가 공지를 추가할 수 있도록 구현했습니다. 관리자가 영화의 정보도 추가할 수 있도록 기능을 추가하려 했지만 시간이 부족하여 이 기능에 대해선 구현하지 못했습니다. 공지에 대한 내용을 추가할 수 있고 이에 대한 결과가 눈에 직접적으로 보인다는 점에서 Creativity에 대한 요소로 admin 기능을 구현했습니다.

또한 남들이 크게 신경 쓰지 않을 만한 부분에 조금 더 공들여 구현했습니다. 어플리케이션의 디자인은 현실에서 사용자가 앱을 설치하고 사용하는데 크게 영향을 미칩니다. UI/UX 디자인, 어플리케이션 자체의 디자인은 사실 컴퓨터공학과/소프트웨어학과 전공생의 입장에서 Back end에 비해 소홀히 할 수 있는 부분입니다. 저는 이런 부분에 직접 어플리케이션의 로고를 제작하고 필요한 이미지와 어울리는 폰트를 찾아 디자인에 신경 썼습니다.



7. Reference

- 1) Sondengs, [안드로이드 스튜디오] 이름 변경, 이름 한번에 바꾸기 단축키, 2017.05.24, <https://blog.naver.com/sondengs/221012753926>
- 2) Naver Software, 폰트, <https://software.naver.com/software/fontList.nhn?categoryId=I0000000>
- 3) Kanzler, [Android]레이아웃(LAYOUT)에 border 옵션 설정해주기, 2016.12.08, <https://kanzler.tistory.com/102>
- 4) PEACE, Android spinner and event listener example using arrayadapter, 2017, <https://mailmail.tistory.com/16>
- 5) 공수당&하루기출, 안드로이드 RecyclerView에서 firebase database 역순 정렬 방법, 2019.01.12, <https://hamzzibari.tistory.com/66>
- 6) 몽키펀치, RecyclerView로 가로 ListView 구현, 2018.10.16, <https://m.blog.naver.com/oaie/221378779560>
- 7) 알통몬, 안드로이드 RecyclerView 가로 스크롤, 수평 스크롤로 만드는 방법, 2017.09.15, <http://rain483.blog.me/221097597659>
- 8) 이것 저것 하는 블로그, (android) TabLayout 화면 꽉 채우도록 설정, 2019.09.21, <https://apt-info.github.io/%EA%B0%9C%EB%B0%9C/android-tablayout-fill/>
- 9) Hello World, Firebase Database 데이터 읽기, 3가지 방법, 2017, <https://stack07142.tistory.com/282>
- 10) DeveloperFury, [Android] EditText 텍스트 위에서부터 입력하기, 2019.09.09, <https://developer-fury.tistory.com/30>
- 11) 뉴월드, 안드로이드::리스트뷰 클릭이벤트 Intent, 2016.11.24, <https://robotluv1226.blog.me/220869831395>
- 12) 아라비안왕자, 안드로이드/Android ArrayAdapter, BaseAdapter를 이용한 ListView 구현, 2012.03.28, <https://arabiannight.tistory.com/entry/안드로이드Android-ArrayAdapter-BaseAdapter를-사용한-ListView-구현>
- 13) 금손, 안드로이드 이미지 URL로 표시하기, 2019.05.06, <https://jizard.tistory.com/179>
- 14) Kakao developers, <https://developers.kakao.com/docs/latest/ko/kakaologin/android>