

## SOPT 25th APP JAM

- 프로젝트명 : Crecker
- 기간 : 2019. 12. 23 ~ 2020. 1. 4
- 파트 : Android
- 기능 : Splash , Retrofit2, Glide, 중첩 프래그먼트, MPchart, SharedReference, TabLayout, ViewPager, 자동배너 등등
- 맡은 부분 : 법률 페이지, 뉴스 페이지, 마이페이지 리포트, 스플래시, 로그인 • 회원가입 수정, 전체적인 Front디자인 수정
- GitHub : [https://github.com/Team-Crecker/Crecker\\_Android.git](https://github.com/Team-Crecker/Crecker_Android.git)
- 간단 후기 : <https://kangmin1012.tistory.com/13>

SOPT에서 진행하는 APP JAM에 안드로이드 개발 파트원으로 참여하여 2주동안 하나의 앱을 만드는 경험을 했다. '크리에이터라면 누구나! 오픈형 MCN 서비스'라는 타이틀을 들고 나온 'Crecker' 팀에 참가해 개발을 시작했다. 실질적인 안드로이드 개발은 이번이 처음이었기 때문에 굉장히 떨리는 경험이었다. 전체적으로 기본적인 기능을 이용해서 구현에 힘썼다.

로그인 화면에서는 e-mail 형식의 아이디와 패스워드를 입력하는 형식으로 구성 되어있다. 앱잼에서는 테스트용 아이디를 이용해서 시연했기 때문에, 아이디에 맞는 패스워드를 입력하지 못했을 경우 로그인이 불가능한 작업만 간단하게만 들어냈다.

아이디와 패스워드를 입력 후 로그인을 누르게 되면, Retrofit으로 서버에 이메일과 패스워드를 보낸다. Response로 성공 또는 실패를 받는데 실패를 받게 되면 아이디 또는 비밀번호를 확인하라는 토스트 메시지를 출력하고, 성공하게 되면 정상적으로 홈 화면으로 들어가게 된다. 이 때 서버로부터 토큰 값을 받는데, 이 토큰 값은 로그인 후 클라이언트가 하는 모든 작업에서 통신할 때 쓰인다. SharedReference로 토큰 값을 저장하고, 통신할 때 Header 부분에 토큰 값을 넣어주면서 통신한다. 시간 관계상 Refresh Token은 무시했다.



로그인 화면

## 1. Home



홈 화면 모습은 회원가입 때 선택하는 관심 카테고리를 통해 보여주는 광고가 달라진다. 전체적인 스크롤 뷰 안에 RecyclerView가 있기 때문에 이중스크롤 방지를 위해 NestedScrollView를 이용하였다. 하단 탭 아이콘은 CustomView를 이용하여 만들었고, '홈 탭', '광고 탭', '법률 탭', '뉴스 탭', 마이페이지 탭' 총 5가지로 구성 되어있다. 광고를 클릭하게 되면 상세 페이지로 넘어가게 된다. RecyclerView 같은 경우는 ItemDercoration을 통해 Item 끼리의 여백을 주어 좀 더 깔끔한 뷰를 구현했다.

홈 화면

## 2. 법률 페이지



법률 페이지 메인



LAW 페이지 메인



아이템 정렬



질문에 대한 답변을 받았을 경우



질문 작성 화면



준비 중 EmptyView

법률 페이지는 Law, Branding, Video 카테고리로 나뉘어져 있으며, 앱잼 기간동에는 Law 페이지만 구현 했다. 법률 페이지 홈은 필독 사항과 법률 전문가의 프로필이 전체적으로 보여진다. Floating Action Button으로 질문 작성이 가능하다.

Law 화면 상단은 ViewPager를 통해 배너 형식으로 구현하였고, 상담내역은 RecyclerView를 이용해 상담내역들을 리스트 형식으로 출력한다. 상담내역 아이템은 서버에서 가져오는데 답변완료 상태, 비밀글 상태 등을 통해 총 4가지 형식으로 화면에 뿌려준다. 이 화면에서도 질문작성이 가능하게 Floating Action Button이 화면상에 존재한다.

아이템의 정렬은 조회순, 답변순, 등록순으로 정렬할 수 있으며, Default값은 답변순이다.

버튼을 누르게 되면 BottomSheet로 정렬 기준이 올라오며

질문화면에서 카테고리가 존재하는데, 이는 RadioButton으로 구현하여 한가지만 선택할 수 있게 했으며 카테고리 선택, 제목 및 내용 작성을 하지 않을 경우 질문이 작성되지 않는다. 비밀글 선택여부는 CheckBox를 통해 설정할 수 있다.



상담 신청 화면. 질문에 대한 답변을 받았을 경우에만 작성 가능

본인이 작성한 질문이 전문가의 답변이 완료 되었을 경우, 더 자세한 상담을 받고 싶으면 상담을 신청할 수 있다. 이름, 날짜, 시간, 상담 받고 싶은 내용을 입력하면 신청할 수 있다. 날짜는 `DateDialog`로 시간은 `TimeDialog`를 이용하여 각각 시간을 받는다. `TimeDialog`같은 경우는 `Spinner`형식이 `Deplicated`되었기 때문에 따로 라이브러리를 받아서 구현하였다. 날짜와 시간 정보를 입력 받을 때 실제로 받는 값이 아니라 가공된 모습으로 보여줘야 하기 때문에, `Formatting`을 이용해 `View`에 보이는 것처럼 사용자가 보기 편하게 바뀌서 화면에 보여준다.



DateDialog



TimeDialog



선택 후 화면에 보여질 때

### 3. News 페이지

News 화면은 크리에이터들이 관심 가질 만한 최신 트렌드나 크리에이터 소식, 정부 지원 사업 및 크리에이터 공모전 등 다양한 정보들을 모아둔 곳으로, 프래그먼트 안에 TabLayout과 ViewPager를 사용했으며 중첩 프래그먼트로 구현하였다. 중첩 프래그먼트를 사용할 때 SupportFragmentManager가 아닌 ChildFragmentManager를 이용해야 중첩 프래그먼트를 쓸 수 있다. 메인 화면의 자동으로 넘어가는 배너는 'BannerViewPager' 라이브러리를 사용했고, 배너를 클릭 시 ViewHolder에서 클릭 이벤트를 처리해 해당 글에 대한 상세 페이지로 넘어가게 구현하였다.



News페이지 메인

메인 화면은 '인기 지원 활동'과 '최신 지원 활동이' 존재한다.

인기 지원 활동 같은 경우는 LinearLayoutManager속성 값으로 HORIZONTAL을 주어 가로 형태로 나오도록 설정했고, 최신 지원 활동은 GridLayoutManager를 사용했다. 이 과정에서 ItemDecoration을 통해 아이템 마다 여백을 설정해줘서 보기 편하게 만들었다.

또, 이중스크롤 방지를 위해서 최신 지원 활동 RecyclerView 속성에 'nestedScrollEnabled'값을 'false'를 주었다. 이렇게 하지 않고 상위 ScrollView를 NestedScrollView로 설정해도 같은 효과를 낼 수 있다.

각 아이템을 누를 때 서버에서 받아온 Idx 값을 Intent 객체에 담아 상세페이지로 이동하고, 받아온 Idx 값을 서버에 보내 요청하면 해당 아이템에 관한 상세 내용을 받아 화면에 출력한다.

Daily 탭은 크리에이터들이 관심가질 만한 최신 트렌드나 기타 뉴스들을 모아놓은 곳이다. 뉴스 아이템을 누르면 인사이트 느낌으로 글을 읽을 수 있다. 매거진 형식으로 만들어서 디자인적으로 깔끔하고 단순히 화면 보여주는 부분이기 때문에 크게 어려움 없이 구현할 수 있었다

현재는 더미데이터들로 구성되어 있지만 실제 스타트업을 하게 된다면, 크롤링을 이용하거나 기자들이 작성할 수 있도록 할 예정이라고 한다.

Support 탭은 크리에이터 공모전이나 정부 지원 사업 등을 모아둔 곳이다. 이 곳은 Daily탭과는 다르게 GridLayout을 이용해 2줄로 아이템을 나열하고 있으며,

아이템 클릭 시 상세 페이지로 이동할 수 있다.

Daily 상세 페이지와 다른 점은 스크랩 기능과 신청하기 기능이 있다는 점이다. 스크랩 기능과 신청하기 기능은 스크롤되지 않고 화면에 고정되어 있으며, 하트 버튼을 누르면 스크랩이 되고, 스크랩이 된 상태에서 한 번 더 누르면 스크랩이 해제된다. 스크랩 유무에 따라 그림이 바뀌고, 버튼을 누르면 스크랩 상태를 서버에 보내 서버 내 해당 아이템의 스크랩 유무 데이터를 수정해준다.

신청하기 버튼은 해당 URL로 이동시켜주는 역할을 한다. 처음에는 실제로 URL로 이동시켰는데, 더미데이터 속에는 URL이 존재하지 않았기 때문에 간단하게 토스트 메시지를 띄워서 신청이 완료되었다는 문구를 출력해 주었다.



## DailyNews에 대한 상세 페이지 화면

## 4. 마이페이지 Report

마이페이지 Report 부분은 크리에이터가 광고가 매칭되고 영상을 올린 후 해당 영상들의 조회 수, 좋아요, 광고비, 평균 ER값을 보여준다. 조회 수는 그래프로도 표현해주며, 하단에는 내 광고 영상 중 가장 조회 수가 높은 카테고리 3등까지 보여준다.

조회수, 좋아요, 광고비는 단위마다 쉼표(,)를 찍어야 하기 때문에 서버에서 받은 값을 formatting 하여 변경된 형식으로 뷰에 보여준다.

Report는 토탈 리포트와 개별 리포트로 나뉘어져 있다. 토탈 리포트는 사용자의 모든 광고 관련 정보를 보여주게 되고, 개별 리포트는 사용자가 진행한 광고 아이템 하나하나에 대한 정보를 보여준다.



Report – 토탈 리포트

그래프는 MPChart 라이브러리를 이용했다. 꺾은선 그래프를 채용했고, 서버로부터 5개의 y축 값을 받는다. 여기서 y축 값은 조회 수가 되고, x축은 시간대가 된다. x축 값은 같은 경우는 그래프를 표현할 때 고정으로 0,1,2,3,4를 주기 때문에 받을 필요가 없어서 y축 값인 조회 수만 받는다. MPChart의 속성을 이용해서 Circle은 보여주지 않고 순수 그래프로 표현했으며, 선 아래쪽 부분은 채워주는 속성을 주어 좀 더 예쁜 그래프를 구현해 보았다. x축 라벨은 시간을 5등분하여 각각 Label을 붙여 보여주고, 마지막에 animation을 통해 y축으로 이동 애니메이션을 주어 그래프가 치솟는 느낌을 주었다.

개별 리포트는 내가 진행한 광고들에 대한 정보를 볼 수 있다. 첫 화면은 RecyclerView를 통해 내가 진행한 광고들의 개별적인 아이템을 볼 수 있다. 각 아이템들에 대한 조회수와 좋아요 수를 서버에서 받아 각각의 아이템마다 적용시켰다. 이 역시 해당 아이템을 클릭하게 되면 자세한 내용을 볼 수 있게 화면 이동을 하게 된다.



Report – 개별 리포트



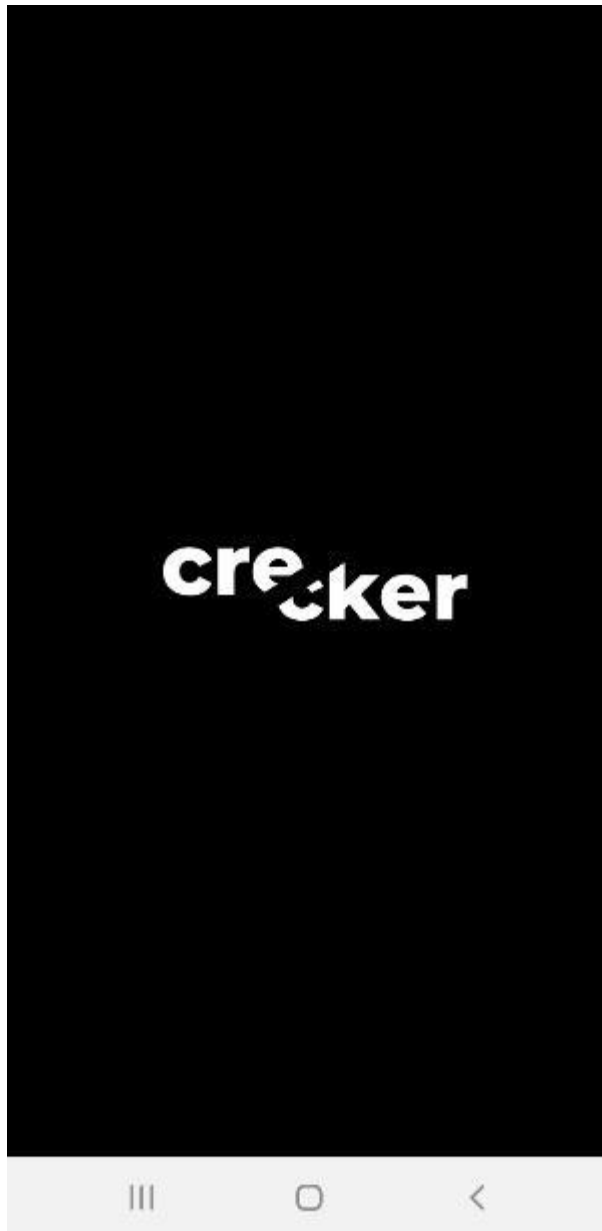
광고 상세 리포트 – 해당 부분은 제대로 보여주지 못 했다.

리포트 부분에서 아쉬웠던 점은 개별 아이템에 대한 상세리포트 화면에서 데이터를 제대로 출력 해주지 못 한다는 점이다. 로그를 찍어보니 들어오는 값이 null값이었고, 서버와의 통신은 정상적이었지만 서버에게 보내주는 광고의 idx 값이 서버 쪽에서 확인이 안됐다. 앱잼 기간 동안 이 부분에 대해서 여러가지 시도를 해보았지만 결국 마지막까지 오류는 찾지 못 했다. 그래서 처음 xml에서 설정한 text 값으로 보여주고 있다. 추가적으로 develop을 하게 된다면 우선적으로 해당 부분의 오류를 찾는 부분부터 시작할 것 같다.



## 5. 스플래시

스플래시 화면은 디자인 팀에서 lottie를 이용해 만들어 주었다. 우리는 이를 적용하기 위해 lottie 라이브러리를 추가하였고, xml에 LottieAnimationView를 선언 한 후 코드에서 startLoading()함수를 만들어 그 안에 핸들러 객체를 생성하였다. 마지막으로 onBackPressed()를 오버라이드해서 스플래시 화면 진행 도중 뒤로가기 키를 누를 수 없게 설정해 주었다.



## 6. 기타 수정

전반적인 뷰의 수정과 로그인 회원가입을 조금 수정했다. 크게 수정한 것은 아니고, 자잘한 뷰의 여백 재설정이나, 글꼴이 제대로 적용되지 않은 부분들을 적용시켜주었다.

회원가입 부분은 총 4단계로 진행이 되는데, 각각의 단계는 프래그먼트로 구성 되어있다.

'다음' 버튼을 누르게 되면 액티비티에서 프래그먼트를 교체하게 되고, 마지막 단계에서 '확인' 버튼을 누르면 회원가입 정보를 서버로 보내준다.

각 단계에서 사용자가 입력한 정보들을 마지막 한번에 서버로 보내줘야 했기 때문에, object 객체를 만들어 그 곳에 정보들을 하나씩 입력 받고 서버와 통신을 할 때, 해당 object 객체를 보내주었다.

처음에는 전부 액티비티로 구성하여 intent 객체를 이용해서 계속해서 값을 이동시키는 방법을 생각했으나, 뭔가 비효율적인 느낌을 받아 object 객체를 이용하게 되었다.

그 때 당시에는, 시간도 촉박 했었고(이걸 생각할 때가 제출 6시간 전이었다.) 더 고민할 틈이 없었기 때문에 바로 적용시키게 되었다. 현재로 와서는 조금 더 효율적인 방법이 무엇이 있을까 생각할 수 있는 부분이다.

```
Object SignData {  
    var email : String = ""  
    var password : String = ""  
    var phone : String = ""  
    var address : String = ""  
    var realName : String = ""  
    var youName : String = ""  
    var youAddress : String = ""  
    var interest : List<String> = listOf()  
    var notRegUrl : String = ""  
}
```

회원가입 정보를 담는 Object

## 7. 후기

SOPT 첫 활동이자 첫 앱잼이었다. 안드로이드 개발자로서 통신관련 부분이나, 뷰를 보여줄 때 어떤 방법으로 보여줘야 더 효율적인가를 생각해 볼 수 있는 소중한 경험이었다. 개인적으로는



Crecker 로고

이번 앱잼에서 결과물을 만들었다는 점보다, 앞으로 내가 안드로이드 개발을 할 때 필요한 것이 무엇인지 확실하게 알 수 있는 가이드라인이 되었다는 점이 더 크게 다가왔다. 이번 앱잼에서는 그냥 액티비티에 모든 코드를 쓰는 단순한 방법을 사용했다. 디자인 패턴은 일절 모르는 상태였기 때문이다. 그러다 안드로이드 멘토들과 이야기하면서, 디자인 패턴의 사용이나 코드의 최적화 부분에 관해서 많은 노력이 필요하다고 느끼게 되었다. 이미 진행된 부분이 어느 정도 있었기 때문에 적용은 시키지 못 했지만, (공부도 안돼 있었다.) 앞으로 개발할 때 MVVM 패턴을 적용시켜서 어플을 만들어보는 하나의 목표가 뚜렷하게 생기게

되었다.

개발적인 실력 상승 부분에서 한 걸음 걸을 수 있게 됨과 동시에, 다른 파트원들과 2주 동안 하나의 목표를 향해 같이 힘냈던 경험은 내게 있어서 엄청난 재산이 되었다. 같은 화면을 보고 있을 때 기획자와 디자이너의 생각은 어떤 지 알 수 있었고, 서버파트원들과의 대화에서 통신을 할 때 어떤 방식으로 통신을 주고 받아야 하는지 얘기를 많이 해봐야 한다는 걸 경험했다. 내가 효율적일 것 같다고 생각 한 통신이 서버 입장에서서는 그렇지 않을 수 있다는 것과, 그와 반대로 서버에서 주는 데이터



**Crecker 시연 부스**

방식이 내 입장에서서는 비효율적일 수도 있

다는 점을 서로 간 활발한 이야기를 통해서 알 수 있었기 때문이다.

안드로이드 파트원끼리의 협업도 좋은 경험이었다. 누가 어느 정도 진행 됐는지 끊임없는 이야기를 통해 서로의 진행상황을 공유하고, Git을 이용한 협업을 통해 각자가 만든 기능이 하나로 통합되었을 때의 쾌감은 이루 말할 수 없었다. Git을 사용하는데 있어서 조금 더 자신감이 생기게 되었고, 협업이 어떤 식으로 이루어지는지에 대해서 배울 수 있었다.

이번 앱잼에서는 안드로이드 파트 리드개발자가 없었다. 그것과 비스무리한 역할을 맡기는 했지만 그 부분은 Git을 전반적으로 관리하는 부분이었고, 실질적으로 개발할 때는 3명의 파트원 모두가 서로 의견을 내면서 모르는 것들은 서로 물어보고 알려주는 식으로 진행했다. 누구 하나 특출나게 잘하거나 못하는 사람이 없었기 때문에, 협업을 하면서 서로가 서로를 도와주는 과정에서 나오는 시너지가 상당했다고 생각한다.

이번 앱잼은 전반적으로 내 부족한 부분이 무엇이었는지 깨닫게 해주었고, 사람과의 협업을 진행하는데 있어서 나는 어떤 식으로 행동해야 하는지 배울 수 있었다.

다음 목표로 MVVM패턴을 적용시키기 위해 필요한 DataBinding, LiveData들을 학습하고 조그마한 개인 프로젝트를 통해 익숙해진 후, Sopt 26기 때 앱잼에서 실제로 MVVM패턴을 이용해 어플을 만드는 것을 목표로 잡았다. 너무나도 값진 경험이었고, 이 경험을 살려 조금 더 성장할 수 있는 내가 되었으면 좋겠다.