

# 안드로이드 기반의 SNS 앱 개발

팀 명 : 브로콜리  
지도 교수 : 이병천 교수님  
팀 장 : 이경호  
팀 원 : 김지환  
이지환

2021. 10.  
중부대학교 정보보호학과

# 목 차

## 1. 서론

1.1 연구 배경 .....	4
1.2 연구의 필요성 .....	4
1.3 연구 목표 .....	4
1.4 성과요약 .....	4

## 2. 관련 연구

2.1 안드로이드 .....	5
2.2 코틀린 .....	5 6
2.3 파이어베이스 .....	6
2.4 기존의 SNS 사례 .....	6

## 3. 본론

3.1 시스템 설계 .....	7
3.2 구현 환경 .....	7
3.3 구현 과정 설명 .....	7
3.3.1 로그인 화면 .....	7 8
3.3.2 파이어베이스 결과화면 .....	9
3.3.3 메인화면 .....	10
3.3.4 메인화면 파이어베이스 .....	11
3.3.5 그리드 화면 .....	11

3.3.6 Diary 작성 .....	12
3.3.7 알림 화면 .....	13
3.3.8 유저 정보 화면 .....	14 15 16
3.3.9 comment .....	17

#### **4. 결론**

4.1 결론 .....	18
4.2 기술적 성취 .....	18
4.2 기대 효과 .....	18

#### **5. 별첨**

5.1 소스 코드 .....	18
5.2 발표 자료 .....	19

# 1. 서론

## 1.1 연구 배경

안드로이드 코틀린으로 SNS를 만들어서 이용자가 단순한 사진 및 글을 쓰는 것이 아닌 본인한테 의미있는 사진으로 올리고 그와 함께 3~4줄 정도의 하루를 정리하여 일기를 쓰듯 sns를 개발하였습니다.

나의 일상과 공감 되는 사람들끼리 팔로우를 해서 같이 공감하면서 일상을 공유 해 보는 시스템의 앱 프로그램이다.

## 1.2 연구의 필요성

코로나로 인해 집에 있는 시간이 많은 요즘 20~30대 사람들에게 SNS 이용으로부터 찾아오는 우울증이 심하다고 한다. 이러한 우울증을 해결하기 위해 자신의 이야기를 그림과 함께 일기로 써보면서 하루를 정리하고 남들과 일상을 공유하면서 같이 소통 할 수 있는 앱이다.

## 1.3 연구 목표

사람들이 자신의 일상이 담긴 일기를 쓰면서 하루를 정리 할 수 있게 만들고, 다른 사람들이 자유롭게 댓글도 달고 좋아요로 공감도 하면서 많은 이용자들이 이용하고 다 같이 자신의 일상을 공유하고 고민거리를 같이 해결해 나가는 SNS가 되는 것이 목표이다.

## 1.4 성과 요약

현재 글을 올리고 거기에 대한 답변을 달아놓는 것까지 진행이 되었고 총 5개의 화면으로 구성이 되어있는데 각 화면마다 전체 글 보기, 글 쓰기, 유저정보, 사진만 보기로 구성이 되어있다.

자신이 올린 글에 대하여 댓글 혹은 좋아요가 늘린다면 사용자에게 알림이 울려 즉시 확인이 가능하다.

## 2. 관련 연구

### 2.1 안드로이드

우리가 사용한 프레임워크인 안드로이드는 구글에서 만든 스마트폰용 운영체제이다. 운영체제와 미들웨어, 사용자 인터페이스 어플리케이션, MMS 서비스 등을 하나로 묶어 서비스를 제공하며 다양한 어플리케이션을 만들어 설치하면 실행 될 수 있도록 구성된 어플리케이션 플랫폼이라고 볼 수 있다. 많은 사람들이 iOS(애플 운영체제)에 견주어 스마트폰과 태블릿으로 안드로이드 운영체제를 사용하면서 안드로이드는 세계 모바일 시장에서 가장 성공한 OS라는 평가를 받고 있다. 안드로이드는 리눅스를 기반으로 제작되었고 언어는 자바를 사용한다.

안드로이드는 2005년 구글이 앤디 루빈이라는 사람이 설립한 안드로이드라는 회사를 인수하면서 시작한다. 작은 소프트웨어 회사에 불과했던 안드로이드사는 구글이 인수하면서부터 운영체제 개발을 시작하면서 없어서는 안될 운영체제로 거듭나기 시작한다. 2007년 11월 안드로이드 플랫폼을 모바일 기기 운영체제로서 무료 공개한다고 발표하였다. 초창기 모바일 시장에서는 아이폰에 대한 대중의 반응은 뜨거웠기 때문에 안드로이드라는 OS를 보유하고 있는 구글은 큰 위협을 느꼈다. 그리하여 애플을 대항하는 반 애플 집단이 형성되는데 그 집단의 이름이 OHA이다. OHA는 리눅스 커널 2.6을 기반으로 한 그들의 첫 번째 모바일 기기 플랫폼 안드로이드를 발표하게 되었으며 여러 가지 안드로이드의 장점과 많은 회사의 협력 덕분에 현재의 독보적인 OS라는 위치까지 오르게 된다.

안드로이드는 대표적인 자바 기반 플랫폼이다. 따라서 전 세계의 많은 자바 개발자들이 안드로이드 앱 개발에 뛰어들 수 있었고, 안드로이드가 빠르게 성장 할 수 있는 계기가 되었다.

### 2.3 코틀린

코틀린은 JetBrains 라는 회사에서 만든 안드로이드 개발을 위한 프로그래밍 언어로 2016년 정식 발표 되었습니다. 코틀린은 기존에 안드로이드 어플리케이션 개발에 활용되는 JAVA언어를 보완하고 대체하기 위해 탄생하였습니다. 때문에 기존에 자바 언어로 동작하던 환경에서 코틀린 언어도 100%호환되는 것이 특징이다. 이는 기존에 자바로 개발하던 프로그램은 모두 코틀린으로 개발 할 수 있게 되었다.

코틀린의 장점에 있어서 문법이 간결하며 사용성이 좋습니다. 그렇기에 성능적인 부분에서 안정성 측면에서는 역사가 긴 자바의 장점이 독보적이지만, 코틀린은 기존 언어가 지닌 단점이나 한계를 보완하여 탄생했기 때문에 개발자 입장에서는 훨씬 사용성이 좋다고 할 수 있다. 코틀린은 문법이 간결하여 복잡한 구문 없이 비교적 직관적으로 코드를 작성 할 수 있다. 또한 표현력이 좋아 입문자 입장에서 빠르게 배울 수 있다는 장점이 있다. 두 번째로 기존에 존재하고 있는 라이브러리와 상호운용성이 좋다. 코틀린은 기존의 자바를 보완하기 위해 탄생 된 언어이기에 자바

와 함께 사용할 수 있고 또한 자바로 만들어진 프로젝트도 처음부터 코틀린으로 다시 개발 할 필요 없이 이후 생성되는 파일만 전부 코틀린으로 작성해도 전체 프로젝트가 정상적으로 작동하여 효율적으로 개발할 수 있다.

## 2.4 파이어베이스

파이어 베이스란 간단하게 말해 운영체제와 상관없이 앱을 개발하는 개발자라면 사용할 수 있는 개발 Tool이다.

파이어베이스는 2011년에 개발이 되었으며 2014년 구글에 인수된 모바일 및 애플리케이션 개발 플랫폼이며, 구글 애널리틱스와 구글 패브릭에서 제공하는 기능을 포함한 다양한 기능들을 제공한다.

또한 앱 개발 기간을 단축시켜주고 앱 개발 난이도를 낮춰줌으로써 개발 방식을 효율적으로 개선해준다.

개발 기능에 있어서 3가지 의 뛰어난 기능이 있다.

1. 인증이다. 개발 기능 중 인증 기능은 사용자가 로그인, 회원가입등을 할 때 보다 쉽고 편리하게 할 수 있도록 돕는 기능이다. 페이스북, 트위터, 깃을 사용할 필요 없이 파이어 베이스이 인증 기능을 이용한 SSO(통합인증)을 지원하는 서비스를 통해 사용자는 로그인을 활동을 쉽게 할 수 있다.

### 2. 실시간 데이터 베이스

파이어 베이스는 NOSql 클라우드 데이터베이스에 JSON 형태로 데이터를 저장하고 클라이언트에 실시간으로 동기화 한다.

## 2.5 기존의 SNS 사례

현재 안드로이드로 출시된 SNS 기반에 앱들은 많이 존재하고 있는 걸로 알고 있다. 대표적인 인스타그램, 트위터 등등 이외에도 여러 가지 채팅이 가능한 앱들이 존재하며 대부분의 많은 앱들은 사진과 함께 글을 쓰거나 동영상을 촬영해서 자신을 나타내고 있다. 하지만 저희 개발 플랫폼은 사진이 아닌 하루의 일상을 정리해 보면서 일기식으로 작성을 해서 남들과 공감되는 부분을 소통하고 같이 고민을 해결 해 볼 수 있는 소소한 SNS를 기반으로 프로젝트를 진행해 보았다.

### 3. 본론

#### 3.1 시스템 설계

안드로이드 스튜디오라는 프레임워크를 사용하고 사용 언어는 코틀린으로 하고 있다. 안드로이드 스튜디오로 구현을 함으로써 레이아웃과 여러 가지 Button, View, Widget 등으로 UI를 만들고 코틀린으로 백엔드와 기능을 부여한다.

#### 3.2 구현 환경

프론트엔드는XML 파일로 각 페이지마다 레이아웃을 구성한다. DB는 파이어베이스를 이용하여 이용자들의 정보를 저장하고 불러온다. 백엔드는 코틀린언어로 XML 파일과 바인딩을 시켜 레이아웃 아이템들의 기능을 만든다.

#### 3.3 구현 과정 설명

##### 1)로그인 화면



먼저 프로그램을 실행하면 로그인과 회원가입을 할 수 있는 액티비티로 이동을 합니다. 그리고 파이어베이스를 사용하여 이메일과 패스워드를 입력하면 자동으로 회원가입과 로그인이 동시에 진행이 되며 해당 이메일이 존재할 경우 로그인과 회원가입이 불가능하다.

The password is invalid or the user does not have a password.



Google  
로그인  
Google 계정 사용 자세히 알아보기

이메일 또는 휴대전화

[이메일을 잊으셨나요?](#)



facebook



project에 연결하려면 Facebook 계정에 로그인하세요

휴대폰 번호 또는 이메일 주소

Facebook 비밀번호

**로그인**

[비밀번호를 잊으셨나요?](#)

[계정 만들기](#)  
[나중에 하기](#)  
[고객 센터](#)

한국어  
Tiếng Việt  
ภาษาไทย  
Português (Brasil)

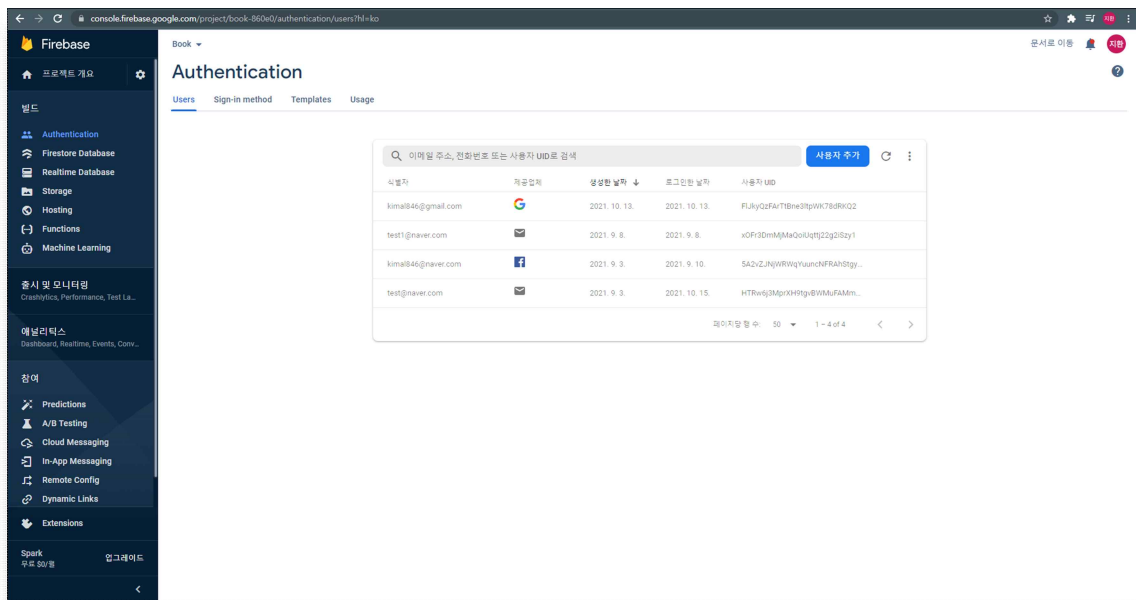
English (US)  
Bahasa Indonesia  
Español  
[+](#)

Facebook, Inc.

또한 OAuth를 사용하여 페이스북 소셜 로그인, 구글 소셜 로그인이 가능하다.



## 2)파이어베이스 결과화면



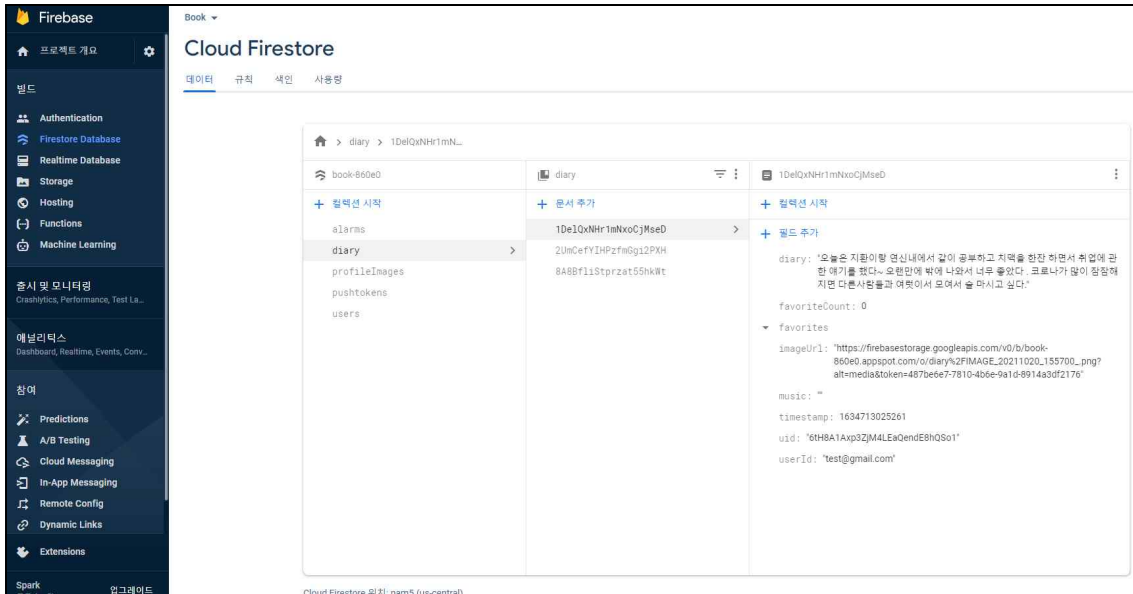
회원가입을 하면 파이어베이스 Authentication에 회원가입을 한 아이디 리스트가 올라온다.

### 3)메인화면



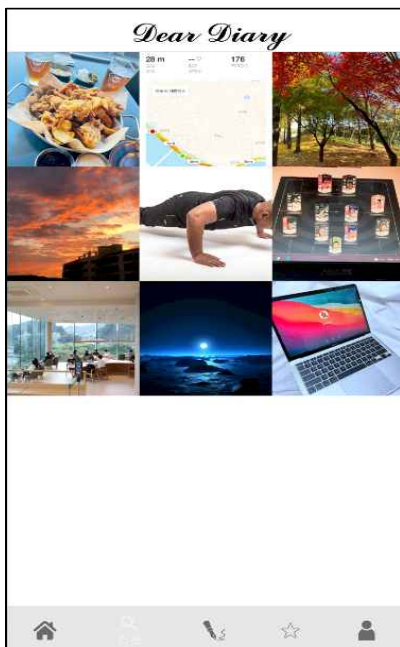
Diary 작성을 통해서 Diary를 업로드하면 결과 화면처럼 실시간으로 업로드가 됩니다. 별을 누르면 Likes 수가 올라가며 상대방에게 알림이 뜬다.

#### 4)메인화면 파이어베이스



Diary를 업로드 하면 diary라는 컬렉션에 추가가 되며 한 개의 Diary마다 내용, 좋아요 수, 좋아요를 누른 사람의 uid, timestamp, 올린 사람의 uid와 email이 Create 된다.

#### 5)그리드 화면 (최근에 쓴 일기들을 사진으로만 나타냄)



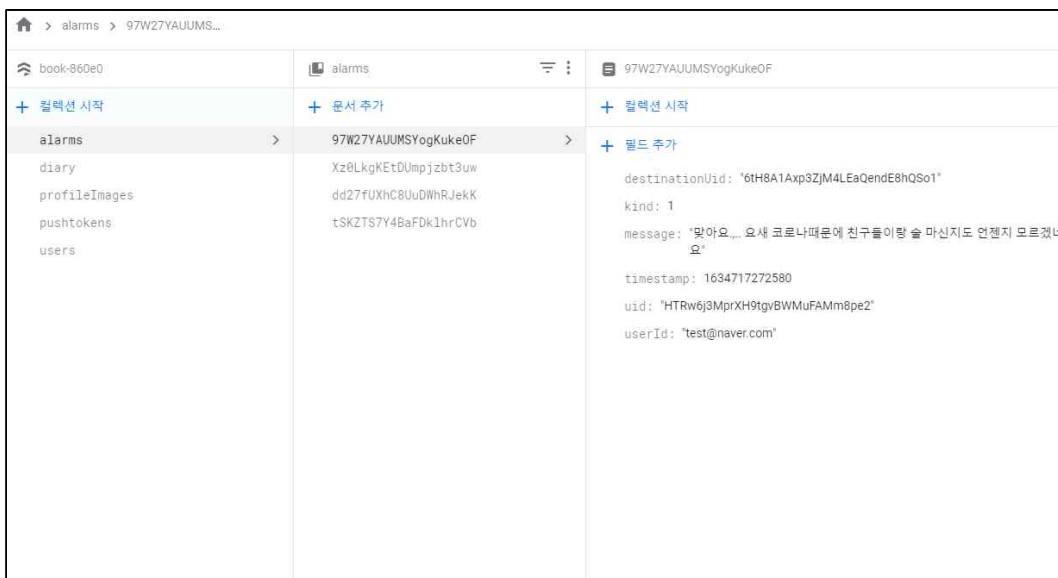
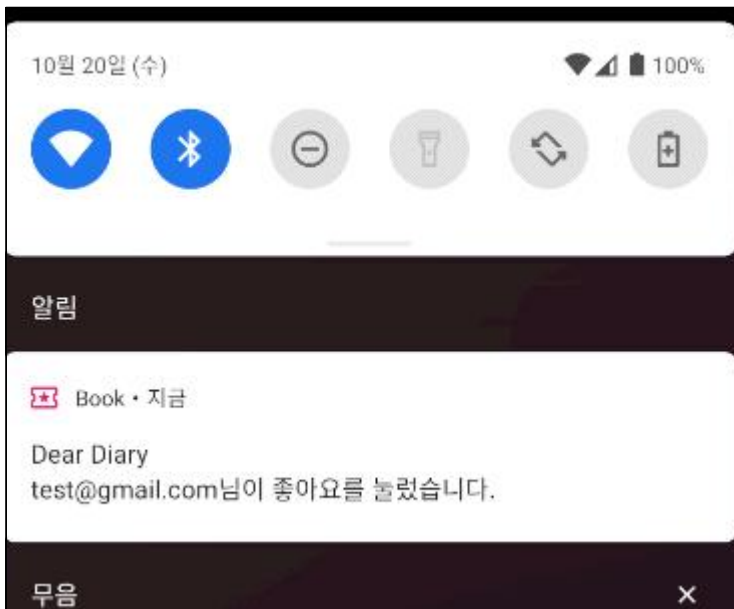
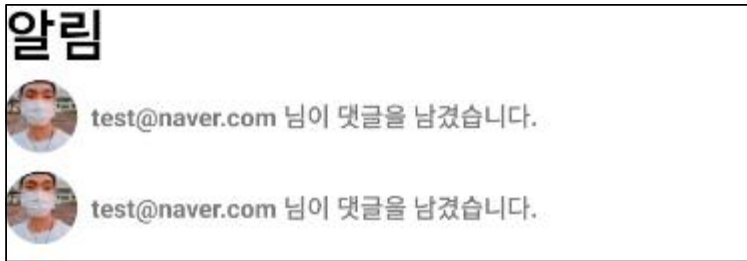
최근에 작성된 일기들의 이미지만 나온다.

## 6)Diary 작성



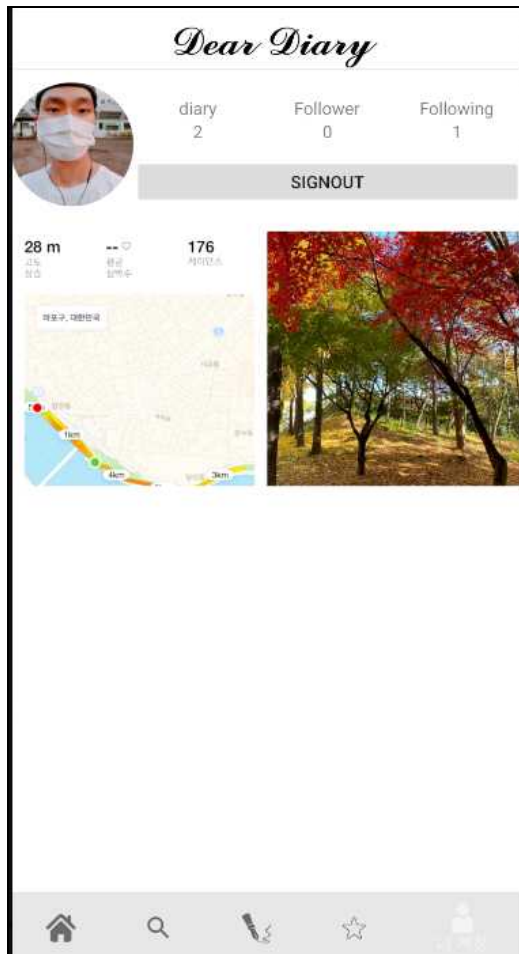
manifest에 android.permission.READ\_EXTERNAL\_STORAGE를 추가하여 사용자의 사진폴더를 열어 해당 사진을 누르면 사진이 추가가 되며 거기에 관한 일기 및 추천 음악을 작성한 후 아래에 있는 작성 완료를 누르면 데이터베이스에 올라가게 된다.

## 7)알람 화면



알람은 Fcm을 활용하여 해당 앱이 실행 중이 아닐지라도 알림이 뜨게 되고 bottom\_navigation 중 알림 창에서도 나타난다.

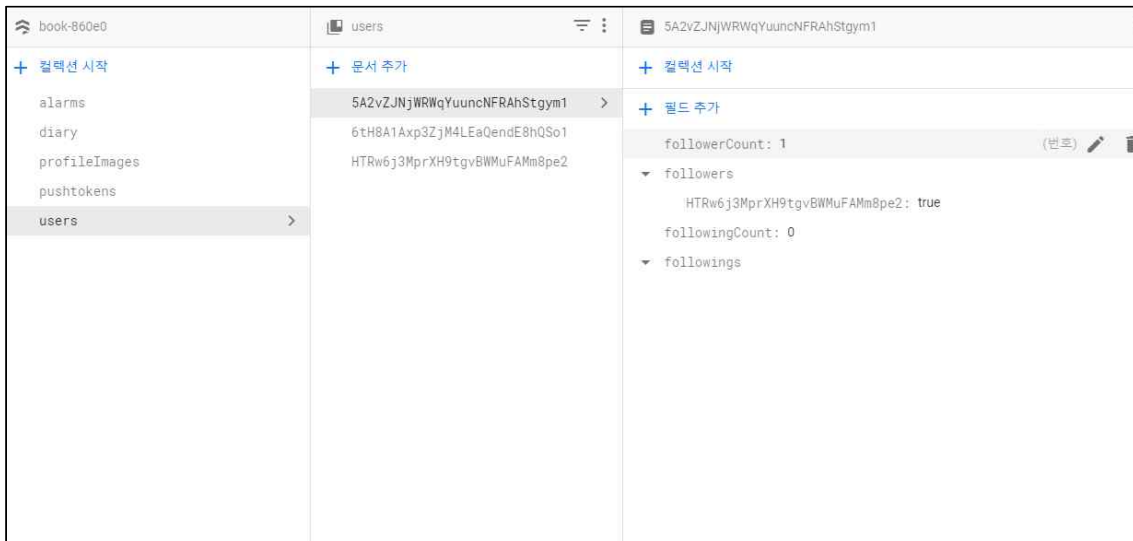
## 8)유저 정보 화면



자신의 계정일 경우 SIGNOUT이라는 화면이 뜨고 SIGNOUT을 누를 시 LoginActivity 화면으로 돌아간다.



메인화면에서 다른 사람의 이메일을 클릭 시 그 사람의 정보로 넘어가며 SIGNOUT 버튼이 팔로우버튼으로 바뀐다.



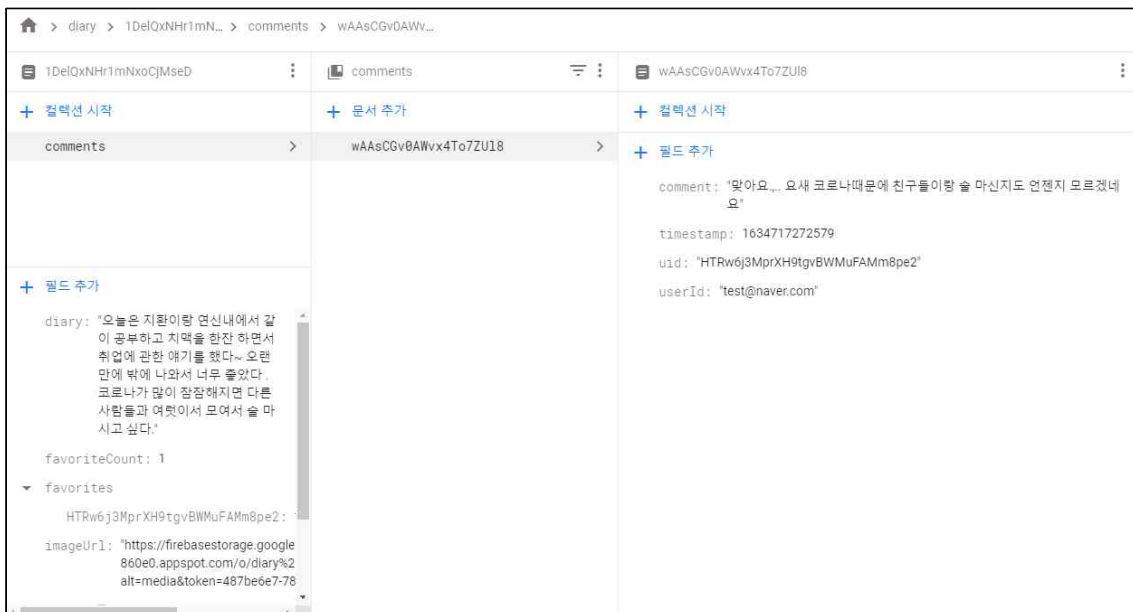
또한 팔로우버튼을 누르게 되면 Follow Cancel 버튼이 생기고 해당 버튼을 또 누르게 되면 FollowerCount 수가 줄어들게 된다.



## 9)Comment



아래 EditText에 입력을 하고 SEND를 누르게 되면 위처럼 실시간으로 올라가게 된다.



해당 diary에 comments 컬렉션에 추가가 된다.

## 4. 결론

### 4.1 결론

우리가 이번에 안드로이드로 개발한 SNS는 코로나로 인해 많은 사람들과 모이지 못하고 힘들어 하는 사람들에게 다른 SNS와 차별화를 두어 이용자에게 조금 더 친근감 있는 주제를 제공함으로써 편하고 솔직한 일상을 공유 할 수 있다.

### 4.2 기술적 성취

OAuth를 이용해서 누구나 쉽게 회원가입을 하고 이용 할 수 있게 개발 하였고, 처음 보는 앱이 아닌 어렸을 때부터 많이 쓰던 일기에 레이아웃을 최대한 활용에서 많은 이용자들에게 익숙함을 주려고 했다.

또한 알림 기능으로 이용해서 실시간으로 사람들과 소통하고 알림 서비스도 받을 수 있다.

### 4.3 기대효과

프로젝트 발표 이후 많은 이용자들이 사용 할 수 있도록 앱 스토어에 런칭 할 계획이고 또한 많은 이용자들이 사용 할 것을 대비해서 데이터베이스에 규모도 늘릴 것이다. 성공적으로 앱 스토어에 런칭이 되고 일반인들도 많이 이용하게 된다면 사업적인 측면과 상품화된 앱으로 발전하기 위해 다양한 기업에서 주최하는 해커톤에 참가하여 앱 시장에 큰 사업으로 발전 할 것이다.

## 5. 별첨

### 5.1 소스코드 (github 주소)

<https://github.com/kimjihw/Book>

## 5.2 발표 자료



# Dear Diary

브로콜리

이경호 91613856  
김지환 91609922  
이지환 91422166



# 목차

- 조원 편성
- 개발 목적
- 구성
- 추진 경과
- 개발 환경
- 개발 내용 및 시스템 운영
- 결론 및 기대효과



# [조원 편성]

레이아웃 개발 **이경호** (조장)

기능 구현 **김지환** (조원)

DB연동 **이지환** (조원)

# [주제 선정]

자존감 열풍, 완벽주의 시대가 만든 '문화적 최면'

이혜인 기자  
2021.10.08 16:15 일력 -

셀피  
월 스토 지음·이현경 옮김

서울신문·공공의장  
국민 2명 중 1명은  
37.4%는 지인이 사  
나타났다. 사회심리  
살과 현실의 '나' 불  
타인의 삶 또한 절

서울신문과 인터넷  
19세 이상 성인 10  
현대인의 삶을 돌아

999만 1천명

국민 절반 "SNS-현실 괴리감"... 빈곤세대 "디지털-현실 모두 불행"

f t y b

SNS 우울 - 스스로 '엄친아'와 비교당하는 고통

A 장두영 정신건강의학과 전문의 | © 일력 2021.09.20 08:00 | 822건

SNS가 꼭 정신건강에 나쁜 것만은 아닙니다. 짧은 시간에 누군가와 연결되어 위로  
움을 덜어줄 수도 있습니다. 그런데 내가 그 장점을 잘 살리고 있지 못하다면 잠시  
멀리할 수도 있습니다. 비유를 하자면, 파티에 초대되어 즐길 수 있다면 내게 좋은  
일이지만, 몸이 아파서 만남을 즐길 수 없는 상태라면 굳이 참석할 필요는 없습니다.

인간은 같은 시간을 쓰면서 살아갑니다. 아무리 비싼 음식이라도 점심을 두 번 먹는  
다고 두 배 즐겁진 않을 것입니다. 또 불편한 사람과 먹은 값비싼 음식이 사진에는  
멋지더라도 좋아하는 사람과 나는 저렴한 음식이 더 만족스러울 것입니다. 내가 어  
떤 가치를 추구하는 사람인지 나의 개성을 잘 알 수 있다면 SNS를 통해 내 욕망을  
자극하는 것으로부터 의욕을 얻을 수도 있습니다. 반대로 보기에는 멋지지만 나와는  
큰 관계가 없다는 것을 알게 될 때 스스로 비교하는 고통을 덜 느낄 수도 있습니다.

## [구성]



## [추진경과]

개발	기간	3월	4월	5월	6월	7월	8월	9월	10월
자료조사 및 연구		■	■						
로그인 구현 및 DB연동			■	■					
게시물 작성 구현 및 DB연동				■	■	■			
BottomNavigationView 구현 및 DB연동					■	■	■	■	
FcmPush 구현								■	
Layout 수정/보완 및 보고서 작성									■

## [개발 환경]



## [개발 내용 및 시스템 운영 1/9]

```
Login
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(binding.root)
    auth = FirebaseAuth.getInstance()

    binding.emailLoginButton.setOnClickListener {
        signInAndSignUp()
    }
    binding.facebookLoginButton.setOnClickListener {
        facebookLogin()
    }
    binding.googleSignInButton.setOnClickListener {
        googleLogin()
    }
}
```



## [개발 내용 및 시스템 운영2/9]

```

main & detail
inner class DetailViewRecyclerViewAdapter : RecyclerView.Adapter<RecyclerView.ViewHolder>() {
    var contentDTOS: ArrayList<ContentDTO> = arrayListOf()
    var contentUIdList: ArrayList<String> = arrayListOf()

    init {
        firestore?.collection("diary")?.orderBy("timestamp", Query.Direction.DESENDING)
            ?.addSnapshotListener { value, error ->
                contentDTOS.clear()
                contentUIdList.clear()
                if (value == null) return@addSnapshotListener
                for (snapshot in value!!.documents) {
                    var item = snapshot.toObject(ContentDTO::class.java)
                    contentDTOS.add(item!!)
                    contentUIdList.add(snapshot.id)
                }
                notifyDataSetChanged()
            }
    }
}
    
```



## [개발 내용 및 시스템 운영3/9]

```

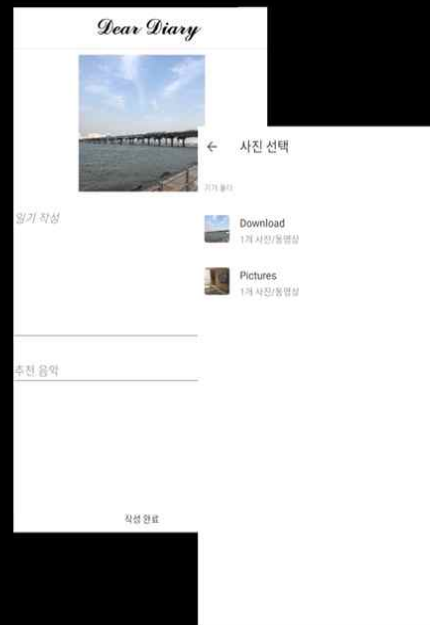
Diary 작성
class AddDiaryActivity : AppCompatActivity() {
    val binding by lazy{ActivityAddDiaryBinding.inflate(layoutInflater)}
    var PICK_IMAGE_FROM_ALBUM = 0
    var storage : FirebaseStorage? = null
    var photoUri : Uri? = null
    var auth : FirebaseAuth? = null
    var firestore : FirebaseFirestore? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(binding.root)

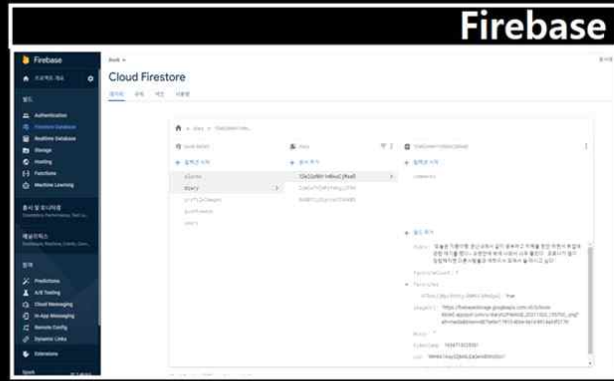
        storage = FirebaseStorage.getInstance()
        auth = FirebaseAuth.getInstance()
        firestore = FirebaseFirestore.getInstance()

        var photoPickerIntent = Intent(Intent.ACTION_PICK)
        photoPickerIntent.type = "image/*"
        startActivityForResult(photoPickerIntent, PICK_IMAGE_FROM_ALBUM)

        binding.addDiaryBtnUpload.setOnClickListener {
            contentUpload()
        }
    }
}
    
```



## [개발 내용 및 시스템 운영4/9]



## [개발 내용 및 시스템 운영5/9]

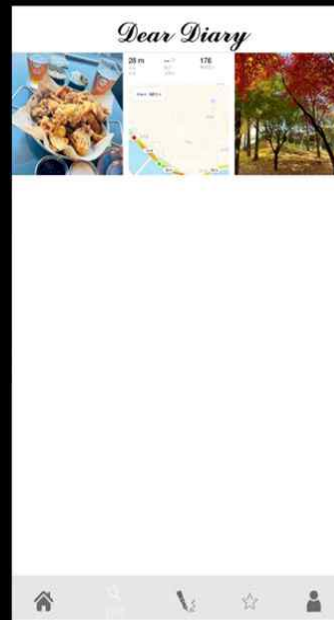
```

Grid

class GridFragment : Fragment() {
    var firestore : FirebaseFirestore? = null
    var fragmentView : View? = null
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        var fragmentView = LayoutInflater.from(activity).inflate(R.layout.fragment_grid, container, false)
        firestore = FirebaseFirestore.getInstance()
        fragmentView.gridFragment_recyclerView.adapter = UserFragmentRecyclerViewAdapter()
        fragmentView.gridFragment_recyclerView.layoutManager = GridLayoutManager(activity, 3)
        return fragmentView
    }
    @SuppressWarnings("NotifyDataSetChanged")
    inner class UserFragmentRecyclerViewAdapter : RecyclerView.Adapter<RecyclerView.ViewHolder>() {
        var contentDTOs: ArrayList<ContentDTO> = arrayListOf()

        init {
            firestore?.collection("diary").addSnapshotListener { querySnapshot, firebaseFirestore ->
                //Some times, This code return null of querySnapshot when it signout
                if (querySnapshot == null) return@addSnapshotListener

                //Set data
                for (snapshot in querySnapshot.documents) {
                    contentDTOs.add(snapshot.toObject<ContentDTO>(<class,java>!!))
                }
                notifyDataSetChanged()
            }
        }
    }
}
    
```





# [개발 내용 및 시스템 운영6/9]

```

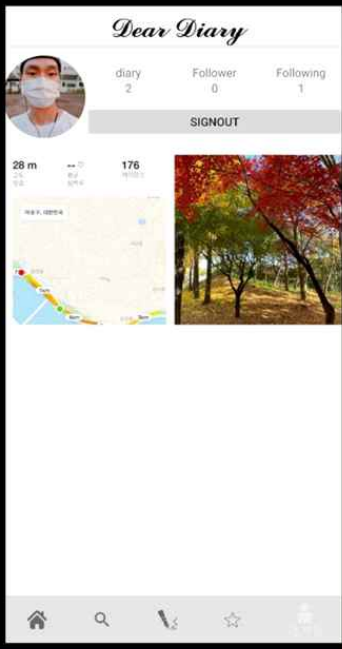
User
FragmentView {
    layoutInflater.inflate(R.layout.fragment_user, container, false)
    uid = arguments.getString("destinationId")
    if (restore != null) {
        restore.restoreInstanceState()
        auth = FirebaseFirestore.getInstance()
        currentUserId = auth.currentUser?.uid
    }

    if (uid == currentUser?.uid) {
        // 로그인
        findViewById<TextView>(R.id.button_signout).setOnClickListener {
            startActivity(Intent(activity, LoginActivity::class.java))
            auth.signOut()
        }
    } else {
        findViewById<TextView>(R.id.button_signout).text = "로그아웃"
        var mainActivity = (activity as MainActivity)
        mainActivity.toolbar.userName?.text = arguments.getString("userId")
        mainActivity.toolbar.btn_back.setOnClickListener {
            mainActivity.bottom_navigation.selectedItemId = R.id.action_home
        }
        mainActivity.toolbar.title_image?.visibility = View.GONE
        mainActivity.toolbar.userName?.visibility = View.VISIBLE
        mainActivity.toolbar.btn_back?.visibility = View.VISIBLE

        findViewById<TextView>(R.id.button_signout).setOnClickListener {
            requestFollow()
        }
    }
}

findViewById<TextView>(R.id.button_signout).setOnClickListener {
    startActivity(Intent(activity, LoginActivity::class.java))
}
}

```



# [개발 내용 및 시스템 운영7/9]

```

alarm(view)
fun sendAlarm(title: String, message: String) {
    val uid = FirebaseAuth.getInstance().currentUser?.uid

    FirebaseFirestore.getInstance().collection("users").document(uid).get()
        .addOnSuccessListener { taskSnapshot ->
            if (taskSnapshot.exists()) {
                val user = taskSnapshot.get() as Map<String, Any>
                val uid = user["uid"] as String

                FirebaseFirestore.getInstance().collection("users").document(uid).collection("alarms").document(title).set(
                    mapOf(
                        "title" to title,
                        "message" to message,
                        "timestamp" to Timestamp.now()
                    )
                )
            }
        }
}

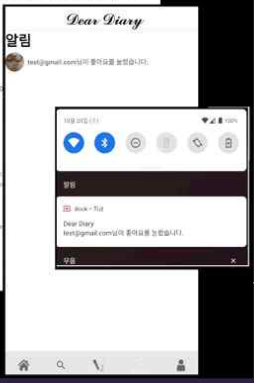
```

```

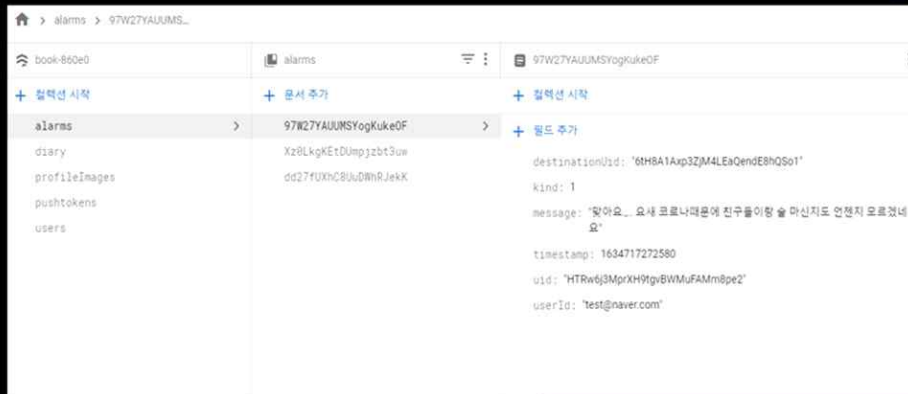
alarm(send)
fun sendAlarm(title: String, message: String) {
    FirebaseFirestore.getInstance().collection("users").document(uid).get()
        .addOnSuccessListener { taskSnapshot ->
            if (taskSnapshot.exists()) {
                val user = taskSnapshot.get() as Map<String, Any>
                val uid = user["uid"] as String

                FirebaseFirestore.getInstance().collection("users").document(uid).collection("alarms").document(title).set(
                    mapOf(
                        "title" to title,
                        "message" to message,
                        "timestamp" to Timestamp.now()
                    )
                )
            }
        }
}

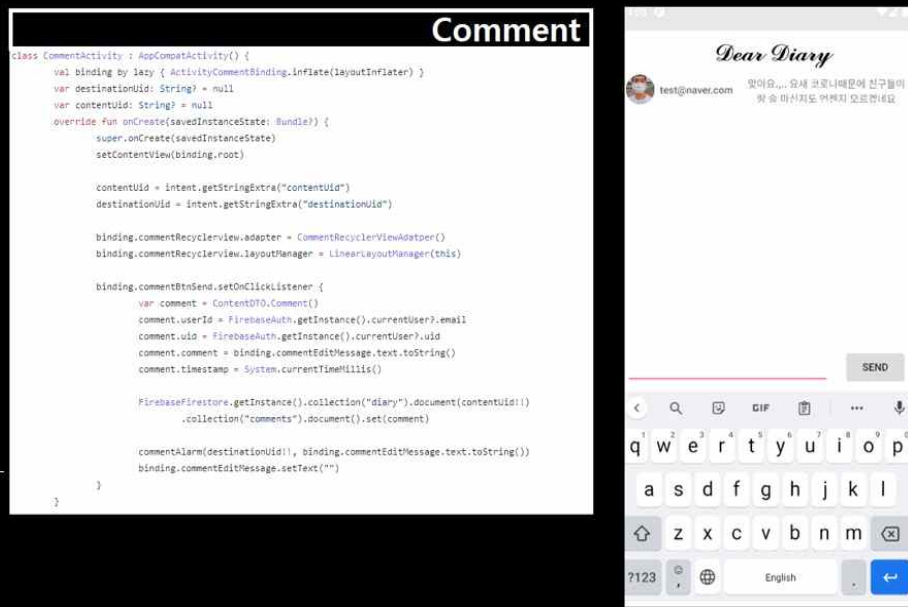
```



## [개발 내용 및 시스템 운영8/9]



## [개발 내용 및 시스템 운영9/9]



## [결론 및 기대효과]

### □ 결론

- Diary 형태의 SNS 구축
- 알림 기능을 사용한 실시간 나에 대한 정보 알림 (좋아요, 댓글, 팔로우)
- 다양한 소셜 로그인을 통한 빠른 회원가입 가능

### □ 기대 효과

- 다른 SNS와 차별점을 두어 이용자들에게 조금 더 친근감 있는 주제를 제공함으로써 편하고 솔직한 개인적인 일상을 공유할 수 있음
- 사회적으로 대두되는 "SNS우울증" 등 기존 SNS의 안 좋은 인식을 개선할 수 있음
- 게시물 작성 시 작성자의 추천 음악을 통하여 작성자의 기분을 좀 더 공감적인 유대감을 형성할 수 있음



감사합니다.