

2주차 개발일지

자료조사

LifeCycle과 Flag



다양한 플래그들 조사하면 과제에 도움 될 것 (ex. 아래와 같은 상황에는 어떤 일이 발생할까요?)

핸드폰이 갑자기 꺼졌을때 전원버튼눌렸을때 전화가왔을 때 알람이 울렸을때 어떤 생명주기가 호출되는걸까?

LifeCycle

SharedPreference

▷ SharedPreference이란

키-값 데이터 저장 | Android 개발자 | Android Developers

저장하려는 키-값 컬렉션이 비교적 작은 경우 SharedPreferences API를 사용해야 합니다. SharedPreferences 객체는 키-값 쌍이 포함된 파일을 가리키며 키-값 쌍을 읽고 쓸 수 있는 간단한 메서드를 제공합니다. 각 SharedPreferences 파일은 프레임워크에서 관리하며 비공개이거나 공유될 수 있습니다. 이 페이지에서는 SharedPreferences API를 사용하여

<https://developer.android.com/training/data-storage/shared-preferences?hl=ko>



작은 양의 데이터를 저잘하고 싶을 때 key-value 형태로 저장하는 SharedPreferences API를 사용한다. 각 SharedPreferences은 데이터를 저장하는 파일을 만들고 그 안의 데이터를 편집한다.

SharedPreferences 객체는 키-값 쌍이 포함된 파일을 가리키며 키-값 쌍을 읽고 쓸 수 있는 간단한 메서드를 제공한다. 각 SharedPreferences 파일은 프레임워크에서 관리하며 비공개이거나 공유될 수 있습니다. 각 SharedPreferences 파일 허용된 이용자에게만 접근되는 파일이므로 자주 값을 바꿀 시 속도가 느려질 수 있다.

▷ SharedPreference 사용하기

[Android] #3-1. SharedPreferences - Practice

컴퓨터 공학과를 졸업하고, 프리랜서로 활동하고 있는 두 선배가 말한다! 첫 영상이라 부족한 점이 많습니다. 많은 피드백 부탁드려요! #컴퓨터공학과 #개발 #안드로이드 #iOS #서버 #소마 #대외활동 #대학생

https://www.youtube.com/watch?v=16YldlqH7_o&list=PLKi37YIxs-7vY_UFtpQR5wGoQC-rhUFY&index=3



- SharedPreferences 객체를 생성해야 한다.
- 데이터에 접근할 때는 Editor를 사용해야 하고 마지막에 apply를 해야 저장이 된다.
- 키 값을 입력하고 .put으로 저장할 수 있고 .get으로 값을 빼 수 있다.

//저장

```
private lateinit var sharedPerferences : SharedPerferences
//(생성하고 싶은 파일 이름, 생성 모드)
sharedPerferences = getSharedPerferences(test, MODE_PRIVATE)
val editor : SharedPerferences.Editor = sharedPerferences.edit()
editor.putInt("age", age)
editor.putString("name", name)
editor.apply()
```

```
//읽기

private lateinit var sharedPerferences : SharedPerferences
sharedPerferences = getSharedPerferences(test, MODE_PRIVATE)
val age = sharedPerferences.getInt("age", age)
val name = sharedPerferences.getString("name", name)
```

BottomNavigationView로 fragment 전환하기

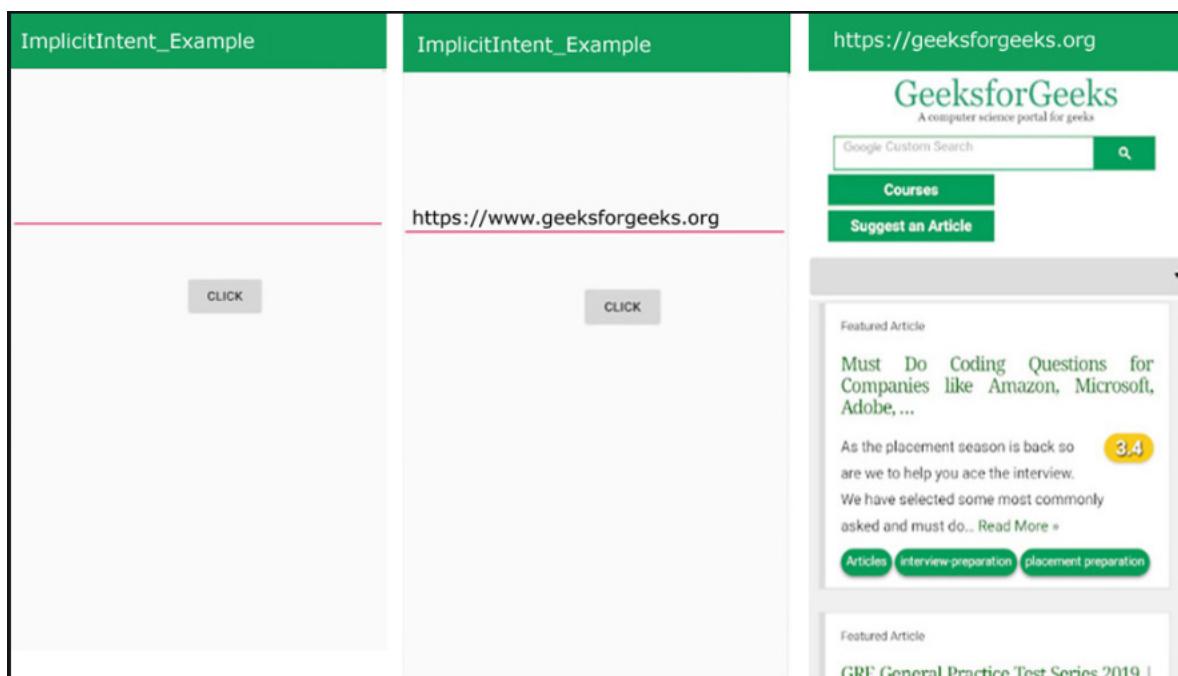
Fragment

Implicit Intent

▷ Implicit Intent이란

Intent는 메시징 객체로, 다른 앱 구성 요소로부터 작업을 요청하는 데 사용할 수 있습니다.

Implicit Intent는 Intent를 받을 명확한 대상의 이름이 정해져 있지 않은 곳에 쓴다.



구현



IBK 기업은행 : onCreate, onStart, onResume, onPause, onStop, onRestart
네이버 블로그 : onCreate, onStart, onResume, onStop, onDestroy

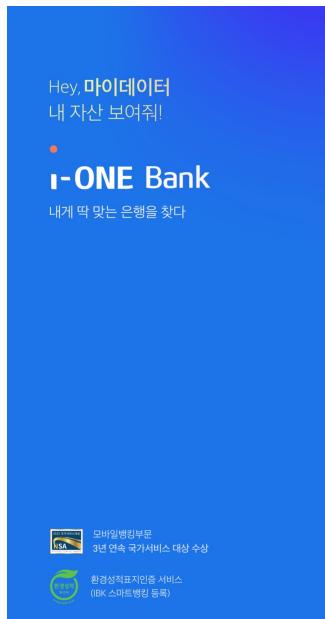
- onCreate
- onStart → 기업은행 화면 보호 복구

- onResume** → 기업은행 scroll position listener & scroll position 복구
- onPause** → 기업은행 화면 보호
- onStop** → 기업은행 scroll position 저장
- onRestart** → 기업은행 광고 넘기기
- onDestroy** → 네이버 블로그 onDestroy에서 자동저장
- SharedPref** → 네이버 블로그 임시저장
- no App bar** → 네이버 블로그 내 정보창
- bottom nav bar with fragment** → 기업은행
- implicit intent** → 네이버 지도

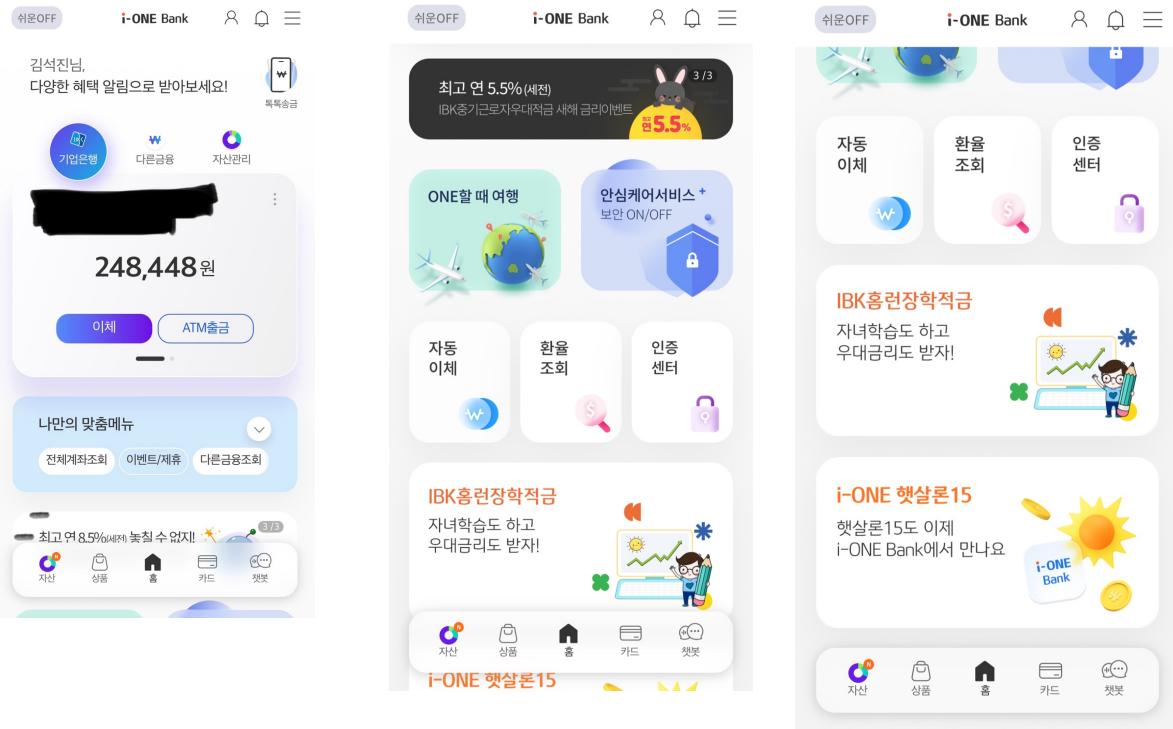
IBK 기업은행

▼ 🌟 기업은행 기존 앱 디자인

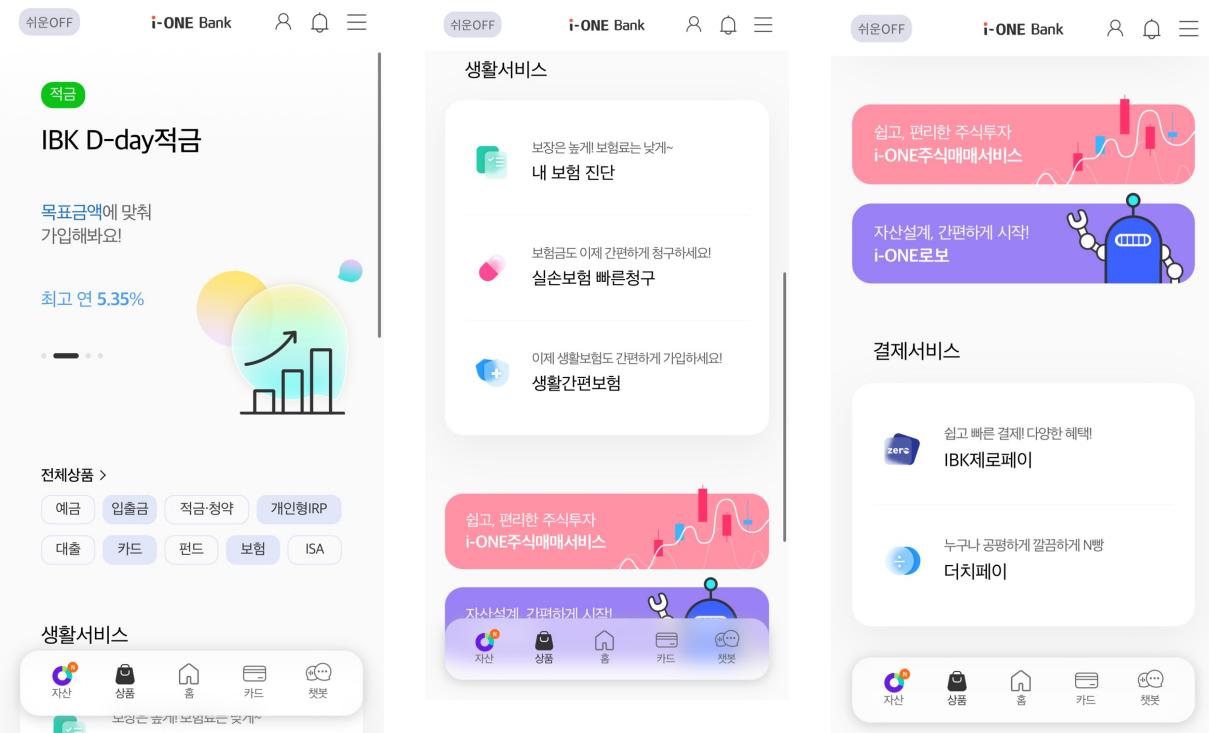
1. 로그인 화면



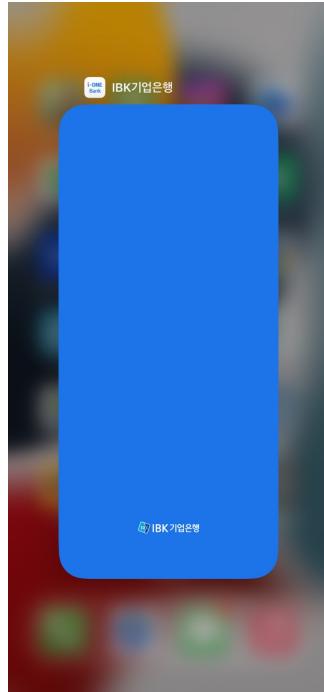
2. 흠 fragment



3. 상품 fragment



4. 개인 정보 보호 화면



▼ ☀ 기업은행 디자인 분석

▷ image scaling

ImageView ScaleType in Android with Example - GeeksforGeeks

ScaleType is used for uniformly scaling the image bounds to the ImageView. Android ImageView provides various types of ScaleType for different configurations. Now, we will look at each of these ScaleType in detail. For exploring these scale types we will use the GeeksforGeeks logo as our image resource also set the [🔗 https://www.geeksforgeeks.org/imageview-scaletype-in-android-with-example/](https://www.geeksforgeeks.org/imageview-scaletype-in-android-with-example/)



▷ bottom navigation 배너 달기

```
var badge = bottomNavigation.getOrCreateBadge(menuItemId)
badge.isVisible = true
// An icon only badge will be displayed unless a number is set:
badge.number = 99
```

▷ CardView스러운 ImageView

등근 모서리와 elevation을 편하게 설정하기 위해서는 cardview안에 넣는게 편하다. imageView에서 elevation은 자유롭지 않다.

```
app:cardBackgroundColor="@color/white"
app:cardCornerRadius="20dp"
android:elevation="3dp"
```

▼ ☀ 기업은행 기능 분석

- 최초로 앱이 시작되는 onCreate부분에 애니메이션을 넣어야 한다.
- pattern lock activity에서 이미 있는 pattern과 비교해서 같은 때만 앱에 진입해야 한다.

- 애니메이션 activity → pattern lock activity → main activity
- bottom navigation으로 fragment 전환하기
- fragment 전환 후에도 fragment 위치와, fragment의 scroll position을 기억해야한다.
- activity의 포커스가 풀릴 때 개인정보 보호용 패란화면이 나타나야 한다.
- viewPager를 이용한 광고 앱 재진입시 전환

▼ ☀ 기업은행 기능 구현

▷ 시작 animation

Create an Animated Welcome Screen for your Android App

This video show you how to create an animated welcome activity for your android application. Like my Facebook page : <https://www.facebook.com/codeglympse> Subscribe My YouTube channel : <http://www.youtube.com/ticoontechologies>

 <https://www.youtube.com/watch?v=QD9nhyWX-gs>

Animated
Welcome Screen For
You Android App

- res > anim 디렉토리를 만들고 animation 추가한다.

translate animation android xml - Google Search

 https://www.google.com/search?q=translate+animation+android+xml&sxsrf=AJOqlzXcwsruEbiZzwLlgKMML2CYbS4esg:1675256451533&source=lnms&tbo=vid&sa=X&ved=2ahUKEwi3583NsPT8AhWS1GEKhcU3BA4Q_AUoAneCAEQBA&biw=1707&bih=793&dpr=1.13#fpstate=ive&vld=cid:9f187995,vid:8fB917XM6_Q

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:fromYDelta="0"
        android:toYDelta="50"
        android:duration="3000"/>
</set>
```

- AnimationActivity 클래스에서 어떤 view에 animation을 적용할지 설정한다.

```
val animationBounceUp = AnimationUtils.loadAnimation(this, R.anim.bounce_up)

binding.bounceImg.startAnimation(animationBounceUp)

val toLockPage = Intent(this, LockActivity::class.java)

animationBounceUp.setAnimationListener(object : Animation.AnimationListener {
    override fun onAnimationRepeat(animation: Animation?) {
    }

    override fun onAnimationEnd(animation: Animation?) {
        startActivity(toLockPage)
        finish()
    }

    override fun onAnimationStart(animation: Animation?) {
    }
})
```

▷ 개인정보 보호 화면 구현

보호 화면 imageView를 elevation="5dp"(가장 상단에 위치하게) 배치하고 visibility를 invisible로 설정하고 원하는 lifecycle callback 메소드에서 visibility를 변화한다.

- Application.ActivityLifecycleCallbacks 연결법

How To Detect Android App Foreground Status

In this article, we're going to learn how to detect if the app is in the foreground or background. Along with that, we'll also learn how to detect the visible activity to the user. In the old days, detecting whether the app is in the foreground or not was one of the common...

 <https://medium.com/the-devops-corner/how-to-detect-android-app-foreground-status-c9443ddef260>



- focus 읽었을 때 화면 바꾸기

```
override fun onWindowFocusChanged(hasFocus: Boolean) {
    super.onWindowFocusChanged(hasFocus)
    if(hasFocus)
        binding.protectionImg.visibility= View.INVISIBLE
    else
        binding.protectionImg.visibility= View.VISIBLE
}
```

- back버튼, home버튼 눌렸을 시 화면 바꾸고 복구

```
override fun onPause() {
    super.onPause()
    binding.protectionImg.visibility= View.VISIBLE
    Log.d(TAG, "onPause: ")

}

override fun onStop() {
    super.onStop()
    binding.protectionImg.visibility= View.VISIBLE
    Log.d(TAG, "onStop: ")

}
```

```
override fun onStart() {
    super.onStart()
    binding.protectionImg.visibility= View.INVISIBLE

    Log.d(TAG, "onStart: ")

}

override fun onResume() {
    super.onResume()
    binding.protectionImg.visibility= View.INVISIBLE

    Log.d(TAG, "onResume: ")

}
```

▷fragment의 scroll 상태 저장

※ 가장 중요한 것은 log.d를 이용해서 어느 lifecycle에서 값이 저장되고 있는지 계속 확인하는 것이다.

※fragment 사이 전환 시에는 onStop까지만 호출되는데 home버튼으로 나가게 되면 onSaveInstanceState까지 불린다.

1. ScrollView의 scroll 위치 값 가져오기 그리고 global variable에 scroll값 저장하기

```
binding.root.setOnScrollChangeListener{v, scrollX, scrollY, oldScrollX, oldScrollY->
    scrollPosition = scrollY
}
```

2. onSaveInstanceState를 override에서 bundle에 저장하기

```

override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
    outState.putInt(SCROLL_POSITION, scrollPosition)
}

```

3. onViewStateRestored를 override해서 bundle에서 과거 scroll값 global variable에 전달

```

override fun onViewStateRestored(savedInstanceState: Bundle?) {
    super.onViewStateRestored(savedInstanceState)
    if(savedInstanceState != null) {
        scrollPosition = savedInstanceState.getInt(SCROLL_POSITION, -1)
    }
}

```

4. onResume에서 scroll 위치 조절하기

```

override fun onResume() {
    super.onResume()
    binding.root.post(Runnable{binding.root.scrollTo(0, scrollPosition)})
    binding.root.setOnScrollChangeListener{v, scrollX, scrollY, oldScrollX, oldScrollY->
        scrollPosition = scrollY
    }
    Log.d(TAG, "onResume: ")
    Log.d(TAG, scrollPosition.toString())
}

```

▼ 여러 실패한 방법들

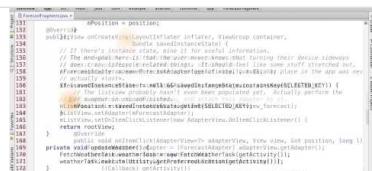
1. onSavedInstanceState를 이용해 fragment의 scrollY값을 저장하려고 했지만 log값 확인 결과 onPause에서 scrollY값을 가져오지 못한다.

→ onPause에서는 값을 가져올 만큼 시간이 없다.

Restore Scroll Position on Rotation - Developing Android Apps

This video is part of an online course, Developing Android Apps. Check out the course here:
<https://www.udacity.com/course/ud853>.

 <https://www.youtube.com/watch?v=tWtaDNJs48o>



2. onStop에서 scrollY값을 가져온다. sharedPref를 사용한다.

→ fragment에서는 sharedPreferences 접근 안된다. activity에서만 getSharedPreferences를 쓸 수 있다.

[Android] #3-1. SharedPreferences - Practice

컴퓨터 공학과를 졸업하고, 프리랜서로 활동하고 있는 두 선배가 말한다! 첫 영상이라 부족한 점이 많습니다. 많은 피드백 부탁드려요! #컴퓨터공학과 #개발 #안드로이드 #iOS #서버 #소마 #대외활동 #대학생

 https://www.youtube.com/watch?v=16YldlqH7_o&list=PLK37Ylx5-7vY_UFtpQR5wGoQC-rhUFY&index=3



▷ViewPager2로 광고, 상단 이미지 연결하기 그리고 상태 저장하기

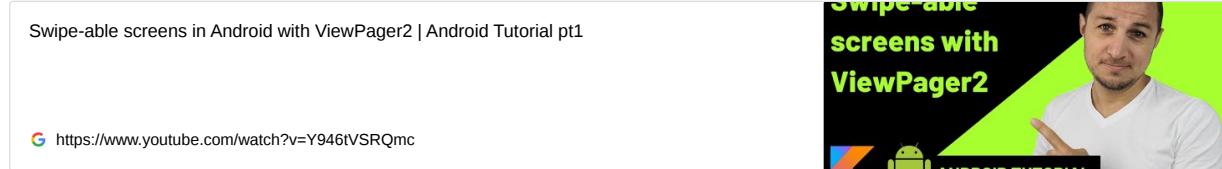
ViewPager2를 사용하여 탭으로 스와이프 뷰 만들기 | Android 개발자 | Android Developers

스와이프 뷰를 사용하면 손가락의 가로 동작이나 스와이프로 탭과 같은 동위 화면 간을 탐색할 수 있습니다. 이러한 탭 세트를 가로 페이징 이라고도 합니다. 이 주제에서는 탭 간 전환을 위해 스와이프 뷰로 탭 레이아웃을 만드는 방법과 탭 대신 제목 스트립을 표시하는 방법을 설명합니다. AndroidX의 ViewPager2 위젯을 사용하여 스와이프 뷰를 만들

 <https://developer.android.com/guide/navigation/navigation-swipe-view-2?hl=ko>



ViewPager2는 swipe view를 만들 수 있는 방법으로 같은 공간에서 가로로 여러 사진 같은 view를 스와이프로 넘길 수 있게 한다.
ViewPager2는 tabLayout과 연동해서 어떤 페이지를 보고 있는지 확인 가능하다.
ViewPager2는 RecyclerView와는 다르게 중간에 멈출 수 없다.
ViewPager2는 ViewPager의 개선된 버전이다.



ViewPager2 마지막에서 다시 처음으로

??

ViewPager2에서 onRestart시 페이지 넘기기

```
private fun moveViewPager(binding: FragmentProductBinding){  
    val adapter = ProductAdapter(PageLists.introSlides)  
    viewPager2 = binding.viewPager  
    viewPager2.adapter = adapter  
    var past = viewPager2?.currentItem  
    if(past != null) {  
        when(past){  
            0 -> viewPager2?.setCurrentItem(1)  
            1 -> viewPager2?.setCurrentItem(2)  
            2 -> viewPager2?.setCurrentItem(3)  
            3 -> viewPager2?.setCurrentItem(0)  
        }  
    }  
}
```

▷여러 이미지 clickable로 만들기

▷로그인 진행 시 틀린 횟수 앱 삭제 후에도 저장

onStop에서 값을 저장 한다.

▷로그인 화면 (sharedPref, no appBar)

GitHub - aritararoy/PatternLockView: An easy-to-use, customizable and Material Design ready Pattern Lock view for Android

An easy-to-use, customizable, Material Design ready Pattern Lock view for Android. This library allows you to implement pattern locking mechanism in your app easily and quickly. It is very easy to use and there are plenty of customization options available to change the functionality and look-and-feel of this view to match your needs.

G <https://github.com/aritararoy/PatternLockView>

aritararoy/
PatternLock

An easy-to-use, customizable and
ready Pattern Lock view for Andro

All 2 Contributors 19 Issues 3k Stars

JitPack | Publish JVM and Android libraries

JitPack makes it easy to release your Java or Android library. Publish straight from GitHub or Bitbucket.

G <https://jitpack.io/#aritararoy/PatternLockView>



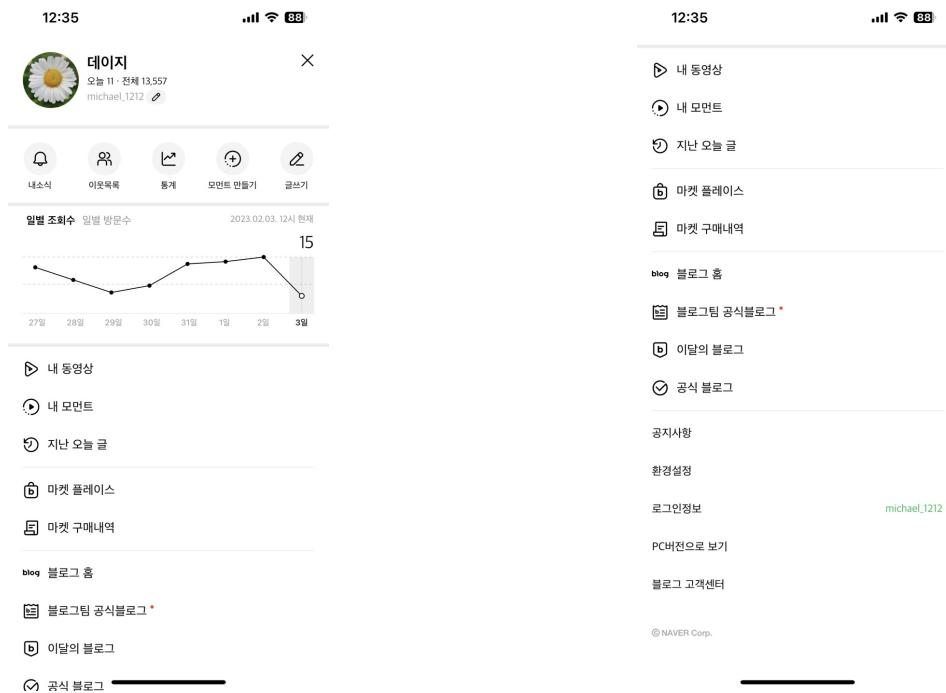
gradle maven 사용하기

intent? 네이버 지도

onDestroy sharedPref? 네이버 블로그

■ 네이버 블로그 ▼ ■ 네이버블로그 기존 앱 디자인

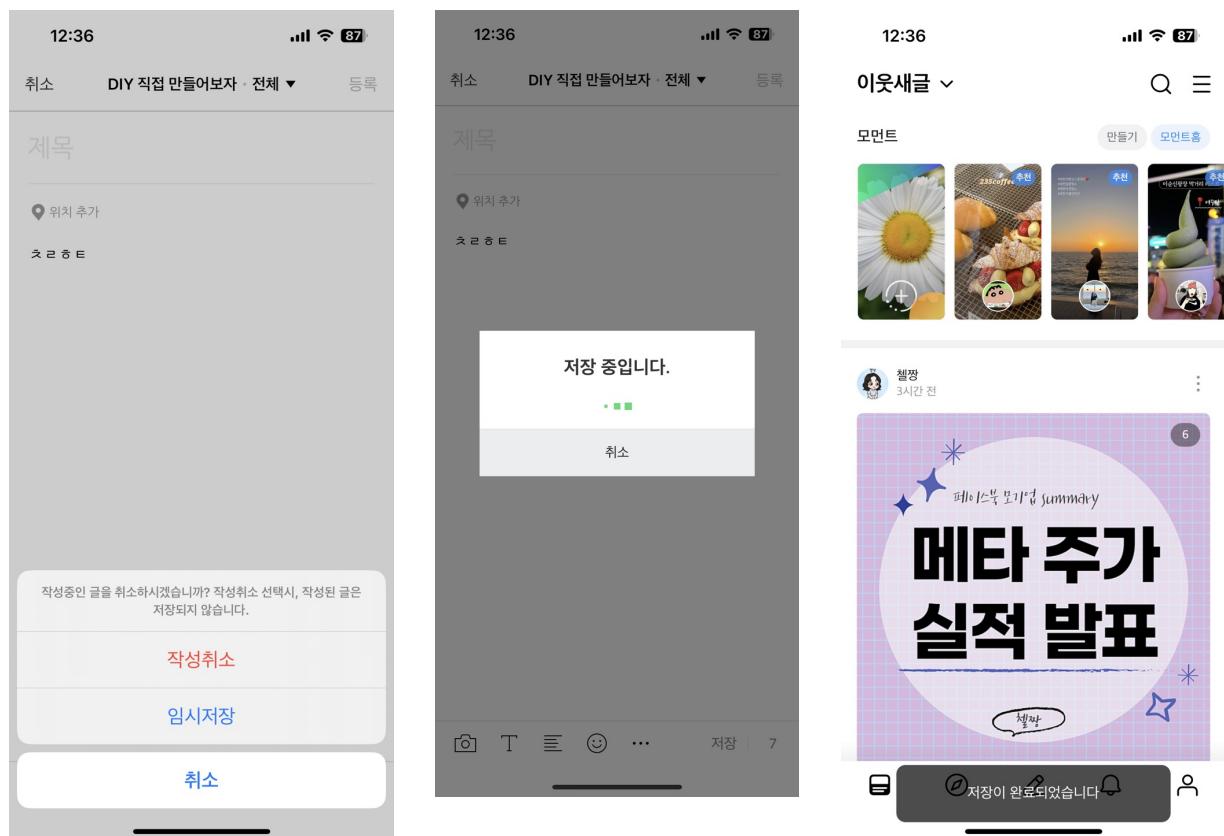
1. 내 프로필 화면



2. 포스팅 화면



3. dialog와 toast



▼ 네이버블로그 기능 분석

- 제목 editText를 눌렀을 때 오른쪽에 이미지를 추가할 수 있는 button이 나타나야 한다.
- 상단 좌측 최소 버튼을 눌렀을 때 임시저장을 할 것인지 물어봐야 한다.
- onStop 이후 다시 진입해도 쓰던 글을 유지해야 한다.
- onDestroy 이후 다시 WriteActivity로 진입 시 자동저장된 글에서 이어 쓸 것인지 물어야 한다.
- 임시저장 후 WriteActivity가 Destroy된 후 임시저장 완료 toast message를 띄워야 한다.

▼ 네이버블로그 기능 구현

▷editText 눌렀을 때 이미지 추가 버튼 등장

```
binding.editText.setOnFocusChangeListener{view, hasFocus->
    if(hasFocus)
        binding.imageViewAddPhoto.visibility= View.VISIBLE
    else
        binding.imageViewAddPhoto.visibility= View.INVISIBLE
}
```

▷SharedPref를 이용한 임시저장

resume에 있는 버튼을 연결해서 current(몇번 째 저장인지)이름으로 sharedpref 파일을 만들어 데이터를 저장한다. 그리고 count값을 1더하고 view에 최신화한다.

```
override fun onResume() {
    super.onResume()
    Log.d(TAG, "onResume: ")
    binding.textView15.setOnClickListener{
        saveData(current)
        current = current + 1
        saveCount(current)
        updateCount(current)
    }
}
```

임시저장 개수를 저장하고 불러오는 메소드를 만들고 임시저장 개수를 받아서 그 개수를 이름으로 제목과 본문을 저장하는 saveData(int) 메소드를 만든다.

```
private fun saveCount(num: Int){
    val sharedpreferences = getSharedPreferences(SAVED_NUM, MODE_PRIVATE)
    val editor = sharedpreferences.edit()
    editor.putInt(COUNT, num)
    editor.apply()
}

private fun getCount(): Int{
    val sharedpreferences = getSharedPreferences(SAVED_NUM, MODE_PRIVATE)
    val past = sharedpreferences.getInt(COUNT, -1)
    if(past == -1)
        return 0
    return past
}

private fun saveData(pageNum: Int) {
    val sharedpreferences = getSharedPreferences(pageNum.toString(), MODE_PRIVATE)
    val editor = sharedpreferences.edit()
    editor.putString(TITLE, binding.editText.getText().toString())
    editor.putString(DETAIL, binding.editTextDetail.getText().toString())
    editor.apply()
    savingDialogAndToast()
}
```

view에 임시저장 개수를 최신화한다.

```
private fun updateCount(num: Int) {
    binding.textView17.text = num.toString()
}
```

onStart()에서 count값을 최신화해서 view

```
override fun onStart() {
    super.onStart()
    Log.d(TAG, "onStart: ")
    current = getCount()
    updateCount(current)
}
```

▷자동 임시저장 구현

onDestroy가 callback되면 autoSave메소드를 불러서 제목과 본문을 자동 저장한다.

```
override fun onDestroy() {
    super.onDestroy()
    Log.d(TAG, "onDestroy: ")
    if(title != null && detail != null){
        autoSave()
        Log.d(TAG, "onDestroy: saved")
    } else {
        //deleteAutoSave()
    }
}

fun autoSave(){
    val sharedpreferences = getSharedPreferences(AUTO, MODE_PRIVATE)
    val editor = sharedpreferences.edit()
    editor.putString(TITLE, binding.editText.getText().toString())
    editor.putString(DETAIL, binding.editTextDetail.getText().toString())
    editor.apply()
}
```

mainActivity에서 글쓰기 activity로 넘어가려고 버튼을 눌렀을 때 만약 autosave된 정보가 있으면 다일로그를 띄워서 intent값에 autosave true값을 전달한다.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)

    binding.roundBtn5.setOnClickListener{
        //TODO dialog
        val intent = Intent(this, WriteActivity::class.java)
        if(detectAutoSave()){
            showCustomDialog()
        } else {
            startActivity(intent)
            finish()
        }
    }
}

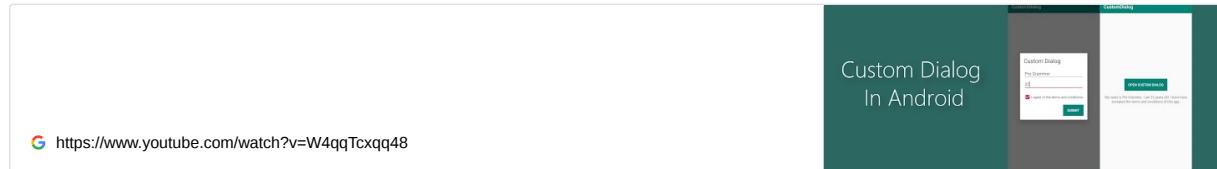
private fun detectAutoSave() : Boolean{
    val sharedpreferences = getSharedPreferences(AUTO, MODE_PRIVATE)
    val title = sharedpreferences.getString(TITLE, "error")
    val detail = sharedpreferences.getString(DETAIL, "error")
    if(title == "error" && detail == "error"){
        return false
    }
}
```

```
        return true  
    }
```

글쓰기 activity에서 onCreate시에 autosave값이 true이면 uploadAutoSave 메소드를 불러와 제목과 본문에 자동저장되어 있는 text를 넣는다.

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    binding = ActivityWriteBinding.inflate(layoutInflater)  
    setContentView(binding.root)  
    Log.d(TAG, "onCreate: ")  
  
    AutoSaveFlag = intent.getBooleanExtra(AUTO, false)  
    if(AutoSaveFlag){  
        uploadAutoSave()  
    }  
  
}  
  
fun uploadAutoSave(){  
    val sharedpreferences = getSharedPreferences(AUTO, MODE_PRIVATE)  
    title = sharedpreferences.getString(TITLE, "")  
    detail = sharedpreferences.getString(DETAIL, "")  
    binding.editText.setText(title)  
    binding.editTextDetail.setText(detail)  
}
```

▷ cutom dialog



```
//Function to display the custom dialog.  
fun showCustomDialog() {  
    val dialog = Dialog(this)  
    //We have added a title in the custom layout. So let's disable the default title.  
    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE)  
    //The user will be able to cancel the dialog by clicking anywhere outside the dialog.  
    dialog.setCancelable(true)  
    //Mention the name of the layout of your custom dialog.  
    dialogBinding = CustomDialogBinding.inflate(layoutInflater)  
    dialog.setContentView(dialogBinding.root)  
  
    dialogBinding.btnClose.setOnClickListener{  
        val intent = Intent(this, WriteActivity::class.java)  
        if(detectAutoSave()){  
            intent.putExtra(AUTO, true)  
        }  
        startActivity(intent)  
        dialog.dismiss()  
        finish()  
    }  
    dialogBinding.btn1.setOnClickListener{  
        dialog.dismiss()  
    }  
    dialog.show()  
}
```

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e5cf8bcd-eb67-49fb-9c8e-4621935e2aca/Untitled.mp4>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e4c6946e-00aa-41fb-b4dc-3e8206a5e870/Untitled.mp4>