

ASSIGNMENT 3

Rohan Makhija

20BCE1846

Youtube App

MainActivity.kt –

```
package com.youtubeAPP_20BCE1846.views.activities
import androidx.appcompat.app.AppCompatActivity import
android.os.Bundle import android.view.View import
androidx.navigation.fragment.NavHostFragment import
androidx.navigation.ui.setupWithNavController import
com.breens.youtubecclone.R import
com.breens.youtubecclone.databinding.ActivityMainBinding
```

```
class MainActivity : AppCompatActivity() {
```

```
    private lateinit var binding: ActivityMainBinding
```

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

```
    private fun setupNav() {
```

```
        val navHostFragment =
```

```
supportFragmentManager.findFragmentById(R.id.fragmentContainerView) as NavHostFragment
val navController = navHostFragment.navController
```

```
//set nav to automatically assign destinations from menu
```

```
binding.bottomNavigationView.setupWithNavController(navController)
```

```
navController.addOnDestinationChangedListener { _, destination, _ ->
```

```
    when (destination.id) {
```

```
        R.id.fragmentHomeScreen -> showBottomNav()
```

```
        else -> hideBottomNav()
```

```
    }
```

```
}
```

```
}
```

```
private fun showBottomNav() {
```

```
binding.bottomNavigationView.visibility = View.VISIBLE
```

```
}
```

```
private fun hideBottomNav() {
```

```
binding.bottomNavigationView.visibility = View.GONE
```

```
}
```

```
}
```

AndroidManifest.xml –

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.breens.youtubecclone">
```

```
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.YoutubeClone">
        <activity
            android:name=".views.activities.MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

```
</manifest>
```

Home.kt –

```
package com.breens.youtubecclone.views.fragments

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import com.breens.youtubecclone.R
import com.breens.youtubecclone.databinding.FragmentHomeScreenBinding
```

```

class FragmentHomeScreen: Fragment(R.layout.fragment_home_screen) {
    private var _binding: FragmentHomeScreenBinding? = null    private val
    binding get() = _binding!!

    override fun onCreateView(
        inflater: LayoutInflater,    container:
        ViewGroup?,    savedInstanceState:
        Bundle?
    ): View {
        _binding = FragmentHomeScreenBinding.inflate(inflater, container, false)
        hideActionBar()    return binding.root
    }

    private fun hideActionBar() {
        (activity as AppCompatActivity).supportActionBar?.hide()
    }

    override fun onDestroyView() {
        super.onDestroyView()
        _binding = null
    }
}

```

Flash.kt –

```

package com.breens.youtubeclone.views.fragments

```

```

import android.os.Bundle import android.view.LayoutInflater import
android.view.View import android.view.ViewGroup import
androidx.appcompat.app.AppCompatActivity import

```

```
androidx.fragment.app.Fragment import com.breens.youtubecclone.R import
com.breens.youtubecclone.databinding.FragmentHomeScreenBinding
```

```
class FragmentHomeScreen: Fragment(R.layout.fragment_home_screen) {
    private var _binding: FragmentHomeScreenBinding? = null    private val
    binding get() = _binding!!
```

```
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        _binding = FragmentHomeScreenBinding.inflate(inflater, container, false)
        hideActionBar()    return binding.root
    }
```

```
    private fun hideActionBar() {
        (activity as AppCompatActivity).supportActionBar?.hide()
    }
```

```
    override fun onDestroyView() {
        super.onDestroyView()
        _binding = null
    }
}
```

Navigation.xml –

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<navigation xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/nav_graph" app:startDestination="@id/splashScreen">

    <fragment android:id="@+id/splashScreen"
android:name="com.breens.youtubeclone.views.fragments.FragmentSplash"
android:label="fragment_splash_screen"
tools:layout="@layout/fragment_splash_screen" >
        <action
            android:id="@+id/action_splashScreen_to_fragmentHomeScreen"
            app:destination="@id/fragmentHomeScreen"
            app:popUpTo="@id/nav_graph"/>
    </fragment>
    <fragment android:id="@+id/fragmentHomeScreen"
android:name="com.breens.youtubeclone.views.fragments.FragmentHomeScreen"
android:label="FragmentHomeScreen"
tools:layout="@layout/fragment_home_screen"/>
</navigation>

```

Test.kt –

```
package com.breens.youtubeclone
```

```
import androidx.test.platform.app.InstrumentationRegistry import
androidx.test.ext.junit.runners.AndroidJUnit4
```

```
import org.junit.Test import
org.junit.runner.RunWith
```

```
import org.junit.Assert.*
```

```
@RunWith(AndroidJUnit4::class) class
ExampleInstrumentedTest {
    @Test fun
useAppContext() {
    // Context of the app under test.    val appContext =
InstrumentationRegistry.getInstrumentation().targetContext
assertEquals("com.breens.youtubeclone", appContext.packageName)
    }
}
```