**Smart Home Controller – Navigation Module**

**1. Project Setup**

* Ensure you have the latest version of Android Studio.
* Add dependencies in build.gradle (app-level):

kotlin

CopyEdit

implementation "androidx.navigation:navigation-fragment-ktx:2.7.1"

implementation "androidx.navigation:navigation-ui-ktx:2.7.1"

**2. Navigation Graph (res/navigation/nav\_graph.xml)**

Create a navigation graph for fragment navigation.

xml

CopyEdit

<?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"

app:startDestination="@id/mainFragment">

<fragment

android:id="@+id/mainFragment"

android:name="com.example.app.MainFragment"

android:label="Smart Devices">

<action

android:id="@+id/action\_mainFragment\_to\_settingsFragment"

app:destination="@id/settingsFragment" />

</fragment>

<fragment

android:id="@+id/settingsFragment"

android:name="com.example.app.SettingsFragment"

android:label="Settings" />

</navigation>

**3. MainActivity (MainActivity.kt)**

Set up the MainActivity with the Navigation Component.

kotlin

CopyEdit

package com.example.app

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import androidx.navigation.NavController

import androidx.navigation.fragment.NavHostFragment

import androidx.navigation.ui.setupActionBarWithNavController

class MainActivity : AppCompatActivity() {

private lateinit var navController: NavController

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val navHostFragment =

supportFragmentManager.findFragmentById(R.id.nav\_host\_fragment) as NavHostFragment

navController = navHostFragment.navController

// Setup ActionBar with NavController

setupActionBarWithNavController(navController)

}

override fun onSupportNavigateUp(): Boolean {

return navController.navigateUp() || super.onSupportNavigateUp()

}

}

**4. Layout for MainActivity (res/layout/activity\_main.xml)**

Define the layout with a NavHostFragment.

xml

CopyEdit

<androidx.constraintlayout.widget.ConstraintLayout 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:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<fragment

android:id="@+id/nav\_host\_fragment"

android:name="androidx.navigation.fragment.NavHostFragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:defaultNavHost="true"

app:navGraph="@navigation/nav\_graph" />

</androidx.constraintlayout.widget.ConstraintLayout>

**5. Main Fragment (MainFragment.kt)**

Define controls for smart devices.

kotlin

CopyEdit

package com.example.app

import android.os.Bundle

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import android.widget.Button

import android.widget.Switch

import androidx.fragment.app.Fragment

import androidx.navigation.fragment.findNavController

class MainFragment : Fragment() {

override fun onCreateView(

inflater: LayoutInflater, container: ViewGroup?,

savedInstanceState: Bundle?

): View? {

val view = inflater.inflate(R.layout.fragment\_main, container, false)

val lightSwitch = view.findViewById<Switch>(R.id.switch\_light)

val thermostatButton = view.findViewById<Button>(R.id.button\_thermostat)

val settingsButton = view.findViewById<Button>(R.id.button\_settings)

// Handle smart device controls

lightSwitch.setOnCheckedChangeListener { \_, isChecked ->

// Toggle light on/off

}

thermostatButton.setOnClickListener {

// Open thermostat settings

}

settingsButton.setOnClickListener {

// Navigate to SettingsFragment

findNavController().navigate(R.id.action\_mainFragment\_to\_settingsFragment)

}

return view

}

}

**Layout (res/layout/fragment\_main.xml):**

xml

CopyEdit

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<Switch

android:id="@+id/switch\_light"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Light Control" />

<Button

android:id="@+id/button\_thermostat"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Thermostat Settings" />

<Button

android:id="@+id/button\_settings"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Open Settings" />

</LinearLayout>

**6. Settings Fragment (SettingsFragment.kt)**

Create a fragment for device preferences.

kotlin

CopyEdit

package com.example.app

import android.os.Bundle

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import android.widget.EditText

import android.widget.Toast

import androidx.fragment.app.Fragment

class SettingsFragment : Fragment() {

override fun onCreateView(

inflater: LayoutInflater, container: ViewGroup?,

savedInstanceState: Bundle?

): View? {

val view = inflater.inflate(R.layout.fragment\_settings, container, false)

val deviceNameInput = view.findViewById<EditText>(R.id.input\_device\_name)

// Handle settings updates

view.findViewById<View>(R.id.button\_save\_settings).setOnClickListener {

val deviceName = deviceNameInput.text.toString()

if (deviceName.isNotBlank()) {

Toast.makeText(context, "Device name saved: $deviceName", Toast.LENGTH\_SHORT).show()

} else {

Toast.makeText(context, "Please enter a device name", Toast.LENGTH\_SHORT).show()

}

}

return view

}

}

**Layout (res/layout/fragment\_settings.xml):**

xml

CopyEdit

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:padding="16dp">

<EditText

android:id="@+id/input\_device\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Device Name" />

<Button

android:id="@+id/button\_save\_settings"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Save Settings" />

</LinearLayout>

**7. Run the App**

* Launch the app.
* Interact with smart device controls on the MainFragment.
* Navigate to the SettingsFragment to configure device preferences.

This design ensures a clean, modular structure using fragments and the Navigation Component.