**In Class Practice Test - 1**

Due Date: Week 6.

Purpose: The purpose of this practice test is to:

1. Help you develop an interactive Android application using Kotlin, Jetpack Compose, Material Design 3, and MVVM architecture on Android 14.

References: Textbook, ppt slides. This material provides the necessary information that you need to complete the exercises.

Be sure to read the following general instructions carefully:

- This practice test must be completed individually by all the students.

- You will have to **implement &** **demonstrate your solution in a scheduled lab session**.

**Exercise 1**

In this exercise you will write a simple MyNotes app.

|  |
| --- |
|  |

#### Step 1: Create a New Project

1. Open Android Studio and select **"New Project"**.
2. Name the project following this format: YourFullName\_COMP304\_001\_PracticeSession\_F24.
   * **Example:** JohnSmith\_COMP304\_001\_PracticeSession\_F24
3. Choose **Empty Activity** as the project template.
4. Set the **minSdkVersion** to 24 and the **targetSdkVersion** to 34 (Android 14).
5. Click **Finish**.

#### Step 2: Main Activity - Setting Up the UI

1. Rename MainActivity to YourFirstNameActivity.kt file, for example JohnActivity.kt.
2. Create a simple UI using Jetpack Compose:
   * Add a logo (Image) at the top.
   * Add an ImageButton below the logo, which navigates to the second activity on click.
   * Use a Column layout with vertical alignment for this UI structure.
3. Here is the code of JohnActivity:

|  |
| --- |
| 1. // JohnActivity.kt package com.example.johnsmith\_comp304\_001\_practicesession\_f24  import android.os.Bundle import androidx.activity.ComponentActivity import androidx.activity.compose.setContent import androidx.compose.foundation.Image import androidx.compose.foundation.layout.\* import androidx.compose.material3.\* import androidx.compose.runtime.Composable import androidx.compose.ui.Alignment import androidx.compose.ui.Modifier import androidx.compose.ui.res.painterResource import androidx.compose.ui.unit.dp import androidx.navigation.NavHostController import androidx.navigation.compose.NavHost import androidx.navigation.compose.composable import androidx.navigation.compose.rememberNavController import com.example.johnsmith\_comp304\_001\_practicesession\_f24.SmithActivityContent  class JohnActivity : ComponentActivity() {  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  *setContent* **{** MyNotesApp()  **}** } }  @Composable fun MyNotesApp() {  val navController = rememberNavController()  NavHost(navController = navController, startDestination = "johnActivity") **{** *composable*("johnActivity") **{** JohnActivityContent **{** navController.navigate("smithActivity")  **}  }** *composable*("smithActivity") **{** SmithActivityContent()  **}  }** }  @Composable fun JohnActivityContent(onNavigate: () -> Unit) {  Column(  modifier = Modifier  .*fillMaxSize*()  .*padding*(16.*dp*),  horizontalAlignment = Alignment.CenterHorizontally,  verticalArrangement = Arrangement.Center  ) **{** Image(  painter = painterResource(id = R.drawable.*logo*), // Replace with your logo image resource  contentDescription = "App Logo",  modifier = Modifier.*size*(128.*dp*)  )  Spacer(modifier = Modifier.*height*(16.*dp*))  Button(onClick = onNavigate) **{** Text("Go to MyNotes")  **}  }** } |

### Step 3: Creating the Second Activity (SmithActivity.kt)

1. **Add a New Kotlin File SmithActivity.kt**

|  |
| --- |
| 1. // SmithActivity.kt  package com.example.johnsmith\_comp304\_001\_practicesession\_f24   import androidx.compose.foundation.layout.\* import androidx.compose.foundation.lazy.LazyColumn import androidx.compose.foundation.text.BasicTextField import androidx.compose.material3.\* import androidx.compose.runtime.\* import androidx.compose.ui.Modifier import androidx.compose.ui.unit.dp import androidx.lifecycle.viewmodel.compose.viewModel  @Composable fun SmithActivityContent(notesViewModel: NotesViewModel = viewModel()) {  var title by remember **{** *mutableStateOf*("") **}** var content by remember **{** *mutableStateOf*("") **}** var priority by remember **{** *mutableStateOf*("Medium") **}** val notes = notesViewModel.notes   Column(modifier = Modifier.*padding*(16.*dp*)) **{** BasicTextField(  value = title,  onValueChange = **{** title = **it }**,  modifier = Modifier  .*fillMaxWidth*()  .*padding*(8.*dp*),  decorationBox = **{** innerTextField **->** Box(modifier = Modifier.*padding*(8.*dp*)) **{** if (title.*isEmpty*()) Text("Note Title")  innerTextField()  **}  }** )  BasicTextField(  value = content,  onValueChange = **{** content = **it }**,  modifier = Modifier  .*fillMaxWidth*()  .*padding*(8.*dp*),  decorationBox = **{** innerTextField **->** Box(modifier = Modifier.*padding*(8.*dp*)) **{** if (content.*isEmpty*()) Text("Note Content")  innerTextField()  **}  }** )  Spacer(modifier = Modifier.*height*(8.*dp*))  Text("Select Priority")  Row **{** RadioButton(selected = (priority == "High"), onClick = **{** priority = "High" **}**)  Text("High")  Spacer(modifier = Modifier.*width*(8.*dp*))  RadioButton(selected = (priority == "Medium"), onClick = **{** priority = "Medium" **}**)  Text("Medium")  Spacer(modifier = Modifier.*width*(8.*dp*))  RadioButton(selected = (priority == "Low"), onClick = **{** priority = "Low" **}**)  Text("Low")  **}** Spacer(modifier = Modifier.*height*(16.*dp*))  Button(onClick = **{** if (title.*isNotEmpty*() && content.*isNotEmpty*()) {  notesViewModel.addNote(Note(title, content, priority))  }  **}**) **{** Text("Add Note")  **}** Spacer(modifier = Modifier.*height*(16.*dp*))  LazyColumn **{** items(notes.size) **{** index **->** Text("${notes[index].title}: ${notes[index].content} [${notes[index].priority}]")  **}  }  }** } |

**Explanation**

* SmithActivityContent handles the form for creating notes.
* The form includes text fields for the title and content, a radio button group for priority, and a button to add a note.
* Notes are displayed using a LazyColumn.

### Step 4: Implementing the ViewModel (NotesViewModel.kt)

1. **Add a New Kotlin File NotesViewModel.kt**

|  |
| --- |
| 1. // NotesViewModel.kt  package com.example.johnsmith\_comp304\_001\_practicesession\_f24   import androidx.compose.runtime.mutableStateListOf import androidx.lifecycle.ViewModel  data class Note(  val title: String,  val content: String,  val priority: String )  class NotesViewModel : ViewModel() {  private val \_notes = *mutableStateListOf*<Note>()  val notes: List<Note> get() = \_notes   fun addNote(note: Note) {  \_notes.add(note)  } } |

**Explanation**

* NotesViewModel manages the list of notes using a mutable state list.
* addNote adds a new note to the list.

### Step 5: Adding Resources

1. **Add Image Resources**
   * Place a logo image (e.g., logo.png) and any other icons in the res/drawable directory.
   * Update the res/values/strings.xml for any string resources.

|  |
| --- |
|  |

1. **Modify the Manifest (AndroidManifest.xml)**
   * Ensure that the activities are declared correctly in the manifest.

|  |
| --- |
| 1. <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools">   <application  android:allowBackup="true"  android:dataExtractionRules="@xml/data\_extraction\_rules"  android:fullBackupContent="@xml/backup\_rules"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:roundIcon="@mipmap/ic\_launcher\_round"  android:supportsRtl="true"  android:theme="@style/Theme.JohnSmith\_COMP304\_001\_PracticeSession\_F24"  tools:targetApi="31">  <activity  android:name=".JohnActivity"  android:exported="true"  android:label="@string/app\_name"  android:theme="@style/Theme.JohnSmith\_COMP304\_001\_PracticeSession\_F24">  <intent-filter>  <action android:name="android.intent.action.MAIN" />   <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>  </manifest> |

### Step 6: Running the App

1. **Build and Run**
   * Ensure all dependencies are installed.

Build.gradle.kts (app):

|  |
| --- |
| plugins **{** alias(*libs*.*plugins*.*android*.*application*)  alias(*libs*.*plugins*.*jetbrains*.*kotlin*.*android*) **}** *android* **{** namespace = "com.example.johnsmith\_comp304\_001\_practicesession\_f24"  compileSdk = 34   defaultConfig **{** applicationId = "com.example.johnsmith\_comp304\_001\_practicesession\_f24"  minSdk = 24  targetSdk = 34  versionCode = 1  versionName = "1.0"   testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"  vectorDrawables **{** useSupportLibrary = true  **}  }** buildTypes **{** *release* **{** isMinifyEnabled = false  proguardFiles(  getDefaultProguardFile("proguard-android-optimize.txt"),  "proguard-rules.pro"  )  **}  }** compileOptions **{** sourceCompatibility = JavaVersion.*VERSION\_1\_8* targetCompatibility = JavaVersion.*VERSION\_1\_8* **}** *kotlinOptions* **{** jvmTarget = "1.8"  **}** buildFeatures **{** compose = true  **}** composeOptions **{** kotlinCompilerExtensionVersion = "1.5.1"  **}** packaging **{** resources **{** excludes += "/META-INF/{AL2.0,LGPL2.1}"  **}  } }** *dependencies* **{** *implementation*(*libs*.*androidx*.*core*.*ktx*)  *implementation*(*libs*.*androidx*.*lifecycle*.*runtime*.*ktx*)  *implementation*(*libs*.*androidx*.*activity*.*compose*)  *implementation*(platform(*libs*.*androidx*.*compose*.*bom*))  *implementation*(*libs*.*androidx*.*ui*)  *implementation*(*libs*.*androidx*.*ui*.*graphics*)  *implementation*(*libs*.*androidx*.*ui*.*tooling*.*preview*)  *implementation*(*libs*.*androidx*.*material3*)  *implementation*(*libs*.*androidx*.*navigation*.*runtime*.*ktx*)  *implementation*(*libs*.*androidx*.*navigation*.*compose*)  *testImplementation*(*libs*.*junit*)  *androidTestImplementation*(*libs*.*androidx*.*junit*)  *androidTestImplementation*(*libs*.*androidx*.*espresso*.*core*)  *androidTestImplementation*(platform(*libs*.*androidx*.*compose*.*bom*))  *androidTestImplementation*(*libs*.*androidx*.*ui*.*test*.*junit4*)  *debugImplementation*(*libs*.*androidx*.*ui*.*tooling*)  *debugImplementation*(*libs*.*androidx*.*ui*.*test*.*manifest*) **}** |

libs.versions.toml:

|  |
| --- |
| [versions] agp = "8.5.0" kotlin = "1.9.0" coreKtx = "1.13.1" junit = "4.13.2" junitVersion = "1.2.1" espressoCore = "3.6.1" lifecycleRuntimeKtx = "2.8.3" activityCompose = "1.9.0" composeBom = "2024.04.01" navigationRuntimeKtx = "2.8.2" navigationCompose = "2.8.2"  [libraries] androidx-core-ktx = { group = "androidx.core", name = "core-ktx", version.ref = "coreKtx" } junit = { group = "junit", name = "junit", version.ref = "junit" } androidx-junit = { group = "androidx.test.ext", name = "junit", version.ref = "junitVersion" } androidx-espresso-core = { group = "androidx.test.espresso", name = "espresso-core", version.ref = "espressoCore" } androidx-lifecycle-runtime-ktx = { group = "androidx.lifecycle", name = "lifecycle-runtime-ktx", version.ref = "lifecycleRuntimeKtx" } androidx-activity-compose = { group = "androidx.activity", name = "activity-compose", version.ref = "activityCompose" } androidx-compose-bom = { group = "androidx.compose", name = "compose-bom", version.ref = "composeBom" } androidx-ui = { group = "androidx.compose.ui", name = "ui" } androidx-ui-graphics = { group = "androidx.compose.ui", name = "ui-graphics" } androidx-ui-tooling = { group = "androidx.compose.ui", name = "ui-tooling" } androidx-ui-tooling-preview = { group = "androidx.compose.ui", name = "ui-tooling-preview" } androidx-ui-test-manifest = { group = "androidx.compose.ui", name = "ui-test-manifest" } androidx-ui-test-junit4 = { group = "androidx.compose.ui", name = "ui-test-junit4" } androidx-material3 = { group = "androidx.compose.material3", name = "material3" } androidx-navigation-runtime-ktx = { group = "androidx.navigation", name = "navigation-runtime-ktx", version.ref = "navigationRuntimeKtx" } androidx-navigation-compose = { group = "androidx.navigation", name = "navigation-compose", version.ref = "navigationCompose" }  [plugins] android-application = { id = "com.android.application", version.ref = "agp" } jetbrains-kotlin-android = { id = "org.jetbrains.kotlin.android", version.ref = "kotlin" } |

* + Test the app on an Android 14 emulator or physical device.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Easy eh!