**Common UI Patterns in Jetpack Compose**

Jetpack Compose offers a set of modern UI components for Android, making it easier to implement standard UI patterns like **Toast**, **Snackbar**, **BottomSheet**, and **Navigation**. These are essential UI elements that enhance user interaction in mobile applications. Here's a detailed explanation of these patterns and their usage in Jetpack Compose:

**1. Toast**

A **Toast** is a small message that appears briefly at the bottom of the screen to provide feedback to the user.

**How to Implement a Toast in Jetpack Compose:**

Jetpack Compose does not have a native Toast composable, but you can easily use the Android Toast inside composables.

**Example:**

import android.widget.Toast

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.Button

import androidx.compose.material3.Text

import androidx.compose.runtime.\*

import androidx.compose.ui.platform.LocalContext

import androidx.compose.ui.unit.dp

import androidx.compose.runtime.Composable

@Composable

fun ToastDemo() {

val context = LocalContext.current

Column(

modifier = Modifier.fillMaxSize(),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

Button(onClick = {

Toast.makeText(context, "This is a Toast message", Toast.LENGTH\_SHORT).show()

}) {

Text(text = "Show Toast")

}

}

}

**2. Snackbar**

A **Snackbar** is similar to a Toast but offers more flexibility, including actions (such as buttons) that users can interact with. It’s commonly used to show short messages or alerts in response to user actions.

**How to Implement a Snackbar in Jetpack Compose:**

Jetpack Compose provides a Snackbar composable as part of the Material library.

**Example:**

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.\*

import androidx.compose.runtime.\*

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.unit.dp

@Composable

fun SnackbarDemo() {

val snackbarHostState = remember { SnackbarHostState() }

val scope = rememberCoroutineScope()

Column(

modifier = Modifier.fillMaxSize(),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

Button(onClick = {

scope.launch {

snackbarHostState.showSnackbar(

message = "This is a Snackbar",

actionLabel = "Undo"

)

}

}) {

Text("Show Snackbar")

}

Spacer(modifier = Modifier.height(20.dp))

SnackbarHost(hostState = snackbarHostState)

}

}

**Key Points:**

* You can provide an **action** like "Undo" to a Snackbar.
* It appears for a set duration and can be dismissed by the user.

**3. BottomSheet**

A **BottomSheet** is a modal or persistent UI component that slides up from the bottom of the screen. It’s commonly used to show additional information or actions without leaving the current context.

Jetpack Compose does not have a native BottomSheet composable yet, but you can use

**ModalBottomSheet** from **Accompanist** or use the **BottomSheetScaffold** provided by Material.

**Example: Using BottomSheetScaffold:**

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.\*

import androidx.compose.runtime.\*

import androidx.compose.ui.Modifier

import androidx.compose.ui.unit.dp

@Composable

fun BottomSheetDemo() {

var sheetState = rememberBottomSheetScaffoldState()

BottomSheetScaffold(

scaffoldState = sheetState,

sheetContent = {

Column(

modifier = Modifier.fillMaxWidth().padding(16.dp),

horizontalAlignment = Alignment.CenterHorizontally

) {

Text("This is the bottom sheet")

Spacer(modifier = Modifier.height(20.dp))

Button(onClick = { /\* Handle action \*/ }) {

Text("Action")

}

}

},

sheetPeekHeight = 0.dp

) {

Column(

modifier = Modifier.fillMaxSize(),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

Button(onClick = {

// Code to show the Bottom Sheet

scope.launch {

sheetState.bottomSheetState.expand()

}

}) {

Text("Show Bottom Sheet")

}

}

}

}

**Key Points:**

* **BottomSheetScaffold** is used to handle the layout for the sheet and the rest of the screen.
* **ModalBottomSheet** can also be used for more temporary, modal-like sheets.

**4. Navigation in Jetpack Compose**

**Navigation** is the process of managing transitions between screens in your app. Jetpack Compose has its own navigation library to handle navigation declaratively using composables.

**How to Implement Navigation in Jetpack Compose:**

1. Add the **Jetpack Compose Navigation** dependency:

dependencies {

implementation "androidx.navigation:navigation-compose:2.5.0"

}

1. Define your composable destinations and set up the navigation graph.

**Example:**

import androidx.compose.foundation.layout.\*

import androidx.compose.material3.\*

import androidx.compose.runtime.\*

import androidx.compose.ui.Modifier

import androidx.navigation.NavHostController

import androidx.navigation.compose.\*

@Composable

fun NavigationDemo() {

val navController = rememberNavController()

NavHost(navController = navController, startDestination = "home") {

composable("home") { HomeScreen(navController) }

composable("details") { DetailsScreen() }

}

}

@Composable

fun HomeScreen(navController: NavHostController) {

Column(

modifier = Modifier.fillMaxSize(),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

Text("Home Screen")

Spacer(modifier = Modifier.height(16.dp))

Button(onClick = {

navController.navigate("details")

}) {

Text("Go to Details")

}

}

}

@Composable

fun DetailsScreen() {

Column(

modifier = Modifier.fillMaxSize(),

verticalArrangement = Arrangement.Center,

horizontalAlignment = Alignment.CenterHorizontally

) {

Text("Details Screen")

}

}

**Key Points:**

* **NavHost**: Defines the navigation graph with routes and screens.
* **NavController**: Handles navigation between screens.
* **composable**: Represents individual screens (destinations).

**Summary**

* **Toast**: A brief message that provides feedback, implemented via Android’s Toast API.
* **Snackbar**: A composable to show brief messages with optional actions.
* **BottomSheet**: A UI component that slides up from the bottom of the screen, created using BottomSheetScaffold.
* **Navigation**: A system to manage app screens, using the NavHost and NavController in Jetpack Compose.