4

Building App Navigation

Activity 4.01 – building primary and secondary app navigation with bottom navigation

Solution

Perform the following steps to solve the problem:

1. Open Android Studio and select **New Project** on the Android welcome screen. Select **Empty Activity** and call it `My Sports`.
2. Create the `Screens.kt` file as you did in *Exercise 4.2 – creating an app with a navigation drawer* and add `ContentScreen`.
3. Create a `Routes.kt` file and add the destinations and bottom navigation:

```
@Serializable
sealed class Destination(val label: String) {

    @Serializable
    data object Home: Destination("Home")

    @Serializable
    data object Calendar: Destination("Calendar")

    @Serializable
    data object Profile: Destination("Profile")
    @Serializable
    data object MySports: Destination("Profile")

    @Serializable
    data class MySportItem(val name: String): Destination(name)
}

sealed class BottomNavigation(
    val label: String,
    val selectedIcon: ImageVector,
    val unselectedIcon: ImageVector,
    val route: Destination
) {
    data object Home : BottomNavigation(
        "Home",
        Icons.Filled.Home,
        Icons.Outlined.Home,
        Destination.Home
    )
    data object Calendar : BottomNavigation(
        "Calendar",
        Icons.Filled.DateRange,
        Icons.Outlined.DateRange,
        Destination.Calendar
    )
    data object Profile : BottomNavigation(
        "Profile",
        Icons.Filled.Person,
        Icons.Outlined.Person,
        Destination.Profile
    )
    data object MySports : BottomNavigation(
        "My Sports",
        Icons.Filled.Star,
        Icons.Outlined.Star,
        Destination.MySports
    )
}
```

This file is very similar to the corresponding `Routes.kt` file in the bottom navigation exercise. The `Destination` `sealed` class has all the destinations that are going to be used to create `NavGraph`, and the `BottomNavigation` sealed class has the navigation items that will be displayed in `bottomBar`.

4. Next, add a `SportButton` composable to the `Screens.kt` file:

```
@Composable
fun SportButton(navController: NavHostController, name: String) {
    OutlinedButton(
        onClick = {
            navController.navigate(Destination.MySportItem(name))},
        modifier = Modifier
            .fillMaxWidth()
            .padding(horizontal = 20.dp),
        shape = RoundedCornerShape(12.dp),
        border = ButtonDefaults.outlinedButtonBorder,
        colors = ButtonDefaults.run {
            outlinedButtonColors(
                containerColor = Color.LightGray,
                contentColor = Color.Black
            )
        },
        elevation = ButtonDefaults.buttonElevation(defaultElevation
            = 2.dp)
    ) {
        Text(
            text = name,
            fontSize = 24.sp,
            fontWeight = FontWeight.Bold,
            modifier = Modifier.padding(vertical = 8.dp)
        )
    }
}
```

The `SportButton` composable has the `NavHostController` and `name` parameters, which are used to navigate to the `MySportItem` destination, passing in the name of the individual sport. The button displays as a simple outlined button.

5. Next, add a `SportsScreen` composable to `Screens.kt`:

```
@Composable
fun SportsScreen(navController: NavHostController) {
    Column(
        verticalArrangement = Arrangement.Top,
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier.padding(16.dp)
    ) {
        SportButton(navController, "Football")
        Spacer(modifier = Modifier.height(12.dp))
        SportButton(navController, "Hockey")
        Spacer(modifier = Modifier.height(12.dp))
        SportButton(navController, "Basketball")
    }
}
```

This composable will be used on the **My Sports** tab to display the three sports buttons. They are spaced evenly from the top of the screen.

6. Next, create a `NavHost` composable using all the destination classes in the `Routes.kt` file:

```
@Composable
fun NavigationHost(navController: NavHostController, modifier:
    Modifier = Modifier) {
    NavHost(
        navController = navController,
        startDestination = Destination.Home,
        modifier = modifier
    ) {
        composable<Home> {
            ContentScreen("Home")
        }
        composable<Profile> {
            ContentScreen("Calendar")
        }
```

```
            composable<Calendar> {
                ContentScreen("Profile")
            }
            composable<MySports> {
                SportsScreen(navController)
            }
            composable<MySportItem> { navBackStackEntry ->
                val appRoute = navBackStackEntry.toRoute<MySportItem>()
                ContentScreen(appRoute.label)
            }
        }
    }
```

7. Create a `BottomNavigationBar` composable to link the bottom navigation items to the destinations and ensure that import 'androidx.navigation.NavDestination.Companion.hasRoute' has been added to the imports list:

```
fun BottomNavigationBar(navController: NavHostController) {
    val navBackStackEntry =
        navController.currentBackStackEntryAsState()
    val currentDestination = navBackStackEntry.value?.destination

    val items = listOf(
        BottomNavigation.Home,
        BottomNavigation.Calendar,
        BottomNavigation.Profile,
        BottomNavigation.MySports,
    )

    NavigationBar(
        containerColor = Color.White,
        contentColor = Color.Black
    ) {
        items.forEach { item ->
```

```
                val isSelected =
                    currentDestination?.hasRoute(item.route::class) ==
                        true

                NavigationBarItem(
                    selected = isSelected,
                    icon = {
                        Icon(
                            imageVector = if (isSelected)
                                item.selectedIcon else
                                    item.unselectedIcon,
                            contentDescription = item.label
                        )
                    },
                    label = { Text(item.label) },
                    onClick = {
                        navController.navigate(item.route) {
                            launchSingleTop = true
                            restoreState = true
                            popUpTo(navController.graph.
startDestinationId) {
                                saveState = true
                            }
                        }
                    }
                )
            }
        }
}
```

8. Create a `MainApp` composable that adds the content for `topBar`, `bottomBar`, and `content`:

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun MainScreen() {
    val navController = rememberNavController()
    Scaffold(
        topBar = {
            CenterAlignedTopAppBar(
                title = { Text("My Sports") },
                modifier = Modifier.statusBarsPadding(),
                colors = TopAppBarDefaults.
centerAlignedTopAppBarColors(
                    containerColor = MaterialTheme.colorScheme.
surfaceContainer
                )
            )
        },
        bottomBar = { BottomNavigationBar(navController) }
    ) { paddingValues ->
        NavigationHost(navController, modifier =
            Modifier.padding(paddingValues))
    }
}
```

9. Finally, update the `onCreate` function with the `MainApp` composable:

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    setContent {
        MySportsTheme {
            MainScreen()
        }
    }
}
```

The display should be similar to the following screen:

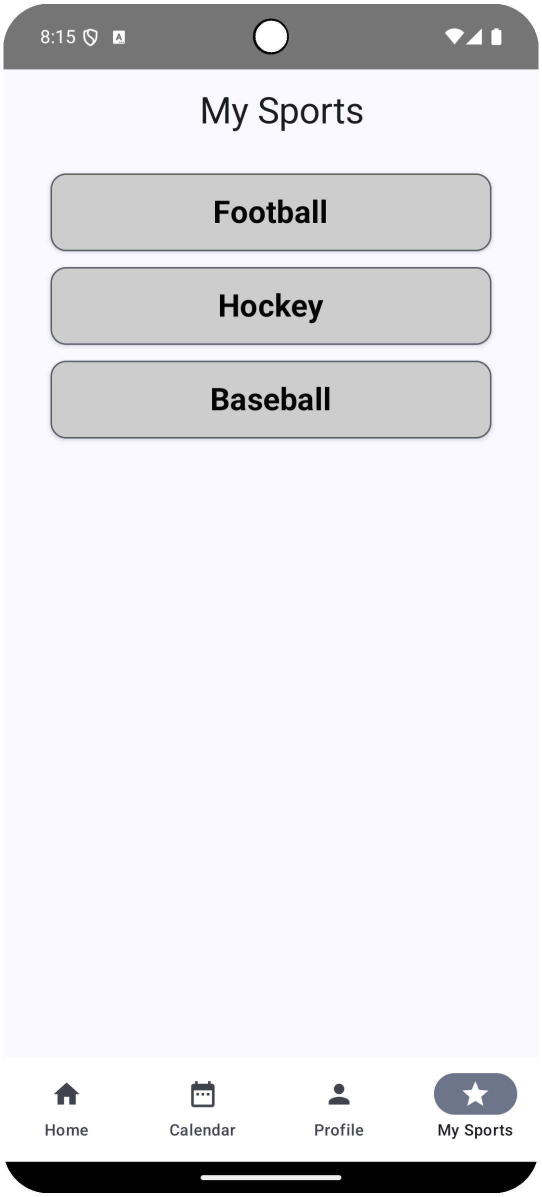

Figure 4.10: The final display