

3

Developing the UI with Jetpack Compose

Activity 3.01 – Creating a business metrics dashboard 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 Dashboard.
2. In MainActivity, add a column containing a Text composable with Business Dashboard as the header of the page:

```
Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
    Column(
        modifier = Modifier
            .fillMaxSize().padding(innerPadding),
        verticalArrangement = Arrangement.Top,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            "Business Dashboard",
            fontWeight = FontWeight.Bold,
            fontSize = 22.sp,
            textAlign = TextAlign.Center,
            modifier = Modifier.fillMaxWidth().padding(bottom = 16.dp)
```

```

    )
  }
}

```

3. Create four states for the four headline metrics above the MainActivity class header:

```

const val TOTAL_SALES = "Total Sales"
const val ACTIVE_USERS = "Active Users"
const val CONVERSION_RATE = "Conversion Rate"
const val REVENUE_GROWTH = "Revenue Growth"

```

4. In setContent, create a MutableState object with the remember function, which will hold the value of the selected metric initialized to "None":

```

var selectedItem by remember {mutableStateOf("None")}

```

5. Create a Screens.kt file and add a DashboardTile composable that will be reused to display a heading. It should take a title text and a click handler for the action when the tile is clicked:

```

@Composable
fun DashboardTile(title: String, onClick: () -> Unit)
{
    Surface(
        shape = RoundedCornerShape(12.dp),
        shadowElevation = 4.dp,
        modifier = Modifier
            .size(150.dp)
            .clickable { onClick() }
    ) {
        Column(
            verticalArrangement = Arrangement.Center,
            horizontalAlignment =
                Alignment.CenterHorizontally,
            modifier = Modifier.padding(8.dp)
        ) {
            Text(
                text = title,
                fontSize = 16.sp,
                fontWeight = FontWeight.Bold
            )
        }
    }
}

```

```

    )
  }
}
}

```

6. Create a `DetailedStatistics` composable that takes in the selected metric and evaluates the value before displaying one of the key metrics in detail. This can be a few lines of text:

```

@Composable
fun DetailedStatistics(selectedItem: String) {
    when (selectedItem) {
        TOTAL_SALES -> {
            Column(
                horizontalAlignment =
                    Alignment.CenterHorizontally
            ) {
                Text(
                    text = "Total Sales Breakdown",
                    fontWeight = FontWeight.Bold,
                    fontSize = 18.sp
                )
                Spacer(
                    modifier = Modifier
                        .height(8.dp)
                )
                Text(
                    text = "Online Sales: $8,000",
                    fontSize = 16.sp
                )
                Text(
                    text = "In-store Sales: $4,000",
                    fontSize = 16.sp
                )
                Text(
                    text =
                        "Year-over-Year Growth: +10%",
                    fontSize = 16.sp
                )
            }
        }
    }
}

```

```
}

ACTIVE_USERS -> {
    Column(
        horizontalAlignment =
            Alignment.CenterHorizontally
    ) {
        Text(
            text = "Active Users Breakdown",
            fontWeight = FontWeight.Bold,
            fontSize = 18.sp
        )
        Spacer(
            modifier = Modifier
                .height(8.dp)
        )
        Text(
            text =
                "Daily Active Users: 1,500",
            fontSize = 16.sp
        )
        Text(
            text =
                "Monthly Active Users: 2,400",
            fontSize = 16.sp
        )
        Text(
            text = "User Retention: 75%",
            fontSize = 16.sp
        )
    }
}

CONVERSION_RATE -> {
    Column(
        horizontalAlignment =
            Alignment.CenterHorizontally
    ) {
```

```
        Text(
            text = "Conversion Rate Details",
            fontWeight = FontWeight.Bold,
            fontSize = 18.sp
        )
        Spacer(
            modifier = Modifier
                .height(8.dp)
        )
        Text(
            text = "Website Conversion: 4.8%",
            fontSize = 16.sp
        )
        Text(
            text = "App Conversion: 6.2%",
            fontSize = 16.sp
        )
        Text(
            text = "Overall Conversion: 5.4%",
            fontSize = 16.sp
        )
    }
}

REVENUE_GROWTH -> {
    Column(
        horizontalAlignment =
            Alignment.CenterHorizontally
    ) {
        Text(
            text = "Revenue Growth Overview",
            fontWeight = FontWeight.Bold,
            fontSize = 18.sp
        )
        Spacer(
            modifier = Modifier
                .height(8.dp)
        )
    }
}
```

```

        Text(
            text = "Last Month: +7.8%",
            fontSize = 16.sp
        )
        Text(
            text = "Last Quarter: +8.2%",
            fontSize = 16.sp
        )
        Text(
            text = "Year-to-Date: +15.5%",
            fontSize = 16.sp
        )
    }
}

else -> {
    Text(
        text =
            "Select an item to view details",
        fontSize = 16.sp
    )
}
}
}

```

7. Create the main body of the dashboard, which will display the four dashboard tiles with a key metric headline title and a click handler that updates the selected metric to one of the four states. Add the Row composables below the Text composable title:

```

Row(
    horizontalArrangement = Arrangement.SpaceEvenly,
    modifier = Modifier.fillMaxWidth()
) {
    DashboardTile(TOTAL_SALES) { selectedItem =
        TOTAL_SALES }
    DashboardTile(ACTIVE_USERS) {
        selectedItem = ACTIVE_USERS
    }
}

```

```
}

Spacer(modifier = Modifier.height(14.dp))

Row(
    horizontalArrangement = Arrangement.SpaceEvenly,
    modifier = Modifier.fillMaxWidth()
) {
    DashboardTile(CONVERSION_RATE) {
        selectedItem = CONVERSION_RATE
    }
    DashboardTile(REVENUE_GROWTH) {
        selectedItem = REVENUE_GROWTH
    }
}
```

8. Add a panel to display the business metric:

```
Surface(
    shape = RoundedCornerShape(12.dp),
    color = Color.LightGray,
    modifier = Modifier
        .fillMaxWidth()
        .padding(34.dp)
        .height(150.dp)
) {
    Box(contentAlignment = Alignment.Center) {
        DetailedStatistics(
            selectedItem = selectedItem
        )
    }
}
```

9. Verify that each of the four tiles populates the company details for each of the corresponding four states.

10. To show which company metric is being displayed, change the background color of the selected `DashboardTile` composable by passing in the `selectedItem` String and comparing it with the dashboard title:

```
@Composable
fun DashboardTile(
    title: String,
    selectedItem: String,
    onClick: () -> Unit
) {
    Surface(
        shape = RoundedCornerShape(12.dp),
        shadowElevation = 4.dp,
        color = if (selectedItem == title)
            Color(0xFFBBDEFB) else Color.White,
        modifier = Modifier
            .size(150.dp)
            .clickable { onClick() }
    ) {
        Column(
            verticalArrangement = Arrangement.Center,
            horizontalAlignment =
                Alignment.CenterHorizontally,
            modifier = Modifier.padding(8.dp)
        ) {
            Text(
                text = title,
                fontSize = 16.sp,
                fontWeight = FontWeight.Bold
            )
        }
    }
}
```

11. Change the places where `DashboardTile` is used to add `selectedItem`:

```
DashboardTile(TOTAL_SALES, selectedItem) {
    selectedItem =
        TOTAL_SALES
```



```
}  
DashboardTile(ACTIVE_USERS, selectedItem) {  
    selectedItem = ACTIVE_USERS  
}  
DashboardTile(CONVERSION_RATE, selectedItem) {  
    selectedItem = CONVERSION_RATE  
}  
DashboardTile(REVENUE_GROWTH, selectedItem) {  
    selectedItem = REVENUE_GROWTH  
}
```

The display should then be like the following screen:

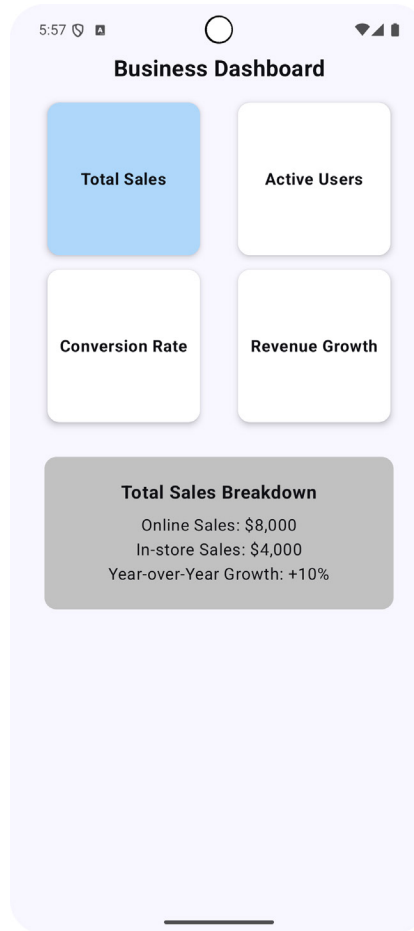


Figure 3.27 – Business Dashboard with selected metric highlighted

