BCS 371 Mobile Application Development I

Arthur Hoskey, Ph.D.
Farmingdale State College
Computer Systems Department

Today's Lecture

- Top app bar
- Floating action button

- A Scaffold helps assemble your app's structure according to Material Design guidelines.
- A TopAppBar should be displayed inside of a Scaffold.

```
Scaffold(
    topBar = { MyTopAppBar() },
    content = {
        Column(
            modifier = Modifier
                .padding(it)
                .fillMaxSize(),
        ) {
            Text(
                text = "Content of the page",
                fontSize = 30.sp


            )
        } // end - Column
    } // end - content
)
```

The topAppBar parameter of the Scaffold takes a composable function as a parameter (MyTopAppBar() function shown on the next slide)

The content parameter displays the main part of the screen. This example has a Column with one Text inside of it.

Scaffold

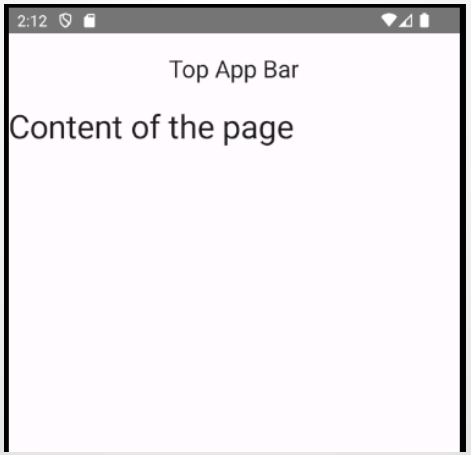
- Here is code for a function to create a TopAppBar with a title:

The TopAppBar is experimental in Material Design 3 so we must opt in

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun MyTopAppBar() {
    CenterAlignedTopAppBar (
        title = { Text(text = "Top App Bar") },
    )
}
```

TopAppBar

Title that appears inside the top app bar



TopAppBar

- Set colors on the top app bar.
- Only showing the top app bar code (not the function code).

```
CenterAlignedTopAppBar (
    title = { Text(text = "My App") },
    colors = topAppBarColors(
        containerColor = MaterialTheme.colorScheme.primaryContainer,
        titleContentColor = MaterialTheme.colorScheme.primary,
    ),
)
```

containerColor is the background color in the top app bar

titleContentColor is the text color

Note: You can set colors for other parts of the top app bar. Check the following link:
https://developer.android.com/reference/kotlin/androidx/compose/material3/TopAppBarColors

TopAppBar – colors

- Add a navigation icon to the top app bar.
- Typically used as a back button.
- Only showing the top app bar code (not the function code).

```
CenterAlignedTopAppBar (
    title = { Text(text = "My App") },
        navigationIcon = {
          IconButton(onClick = {
            // onClick event handler code goes here
          })
          {
            Icon(Icons.AutoMirrored.Filled.ArrowBack, "backIcon")
          } // end - IconButton
        } // end - navigationIcon
)
```

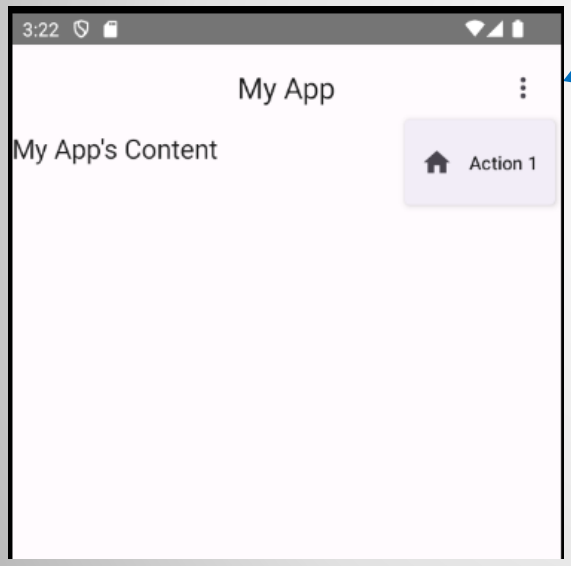
Add event handler code for when the navigation icon is pressed here

Navigation icon to display

TopAppBar – navigationIcon

TopAppBar - actions

- Actions provide access to the app's key actions.
- They appear on the right side of the top app bar.
- Press the vertical dots icon to display a menu of actions to choose from.
- Example code on next slide.



The action icon is three vertical dots

Actions should appear after pressing the action icon. There is only one action in this example. You can have as many as you want.

TopAppBar – actions

- Add actions to the top app bar.

```
var showMenu by remember { mutableStateOf(false) }
CenterAlignedTopAppBar (
    title = { Text(text = "My App") },
    actions = {
        IconButton(onClick = { showMenu = !showMenu }) {
            Icon(Icons.Default.MoreVert, contentDescription = null)
        }
        DropdownMenu(
            expanded = showMenu,
            onDismissRequest = { showMenu = false }
        ) {
            DropdownMenuItem(
                text= { Text("Action 1") },
                onClick =  {
                    // Action 1 click handler code goes here
                    showMenu = false
                },
                leadingIcon = { Icon(Icons.Filled.Home, contentDescription = null) })
        } // end - DropdownMenu
    } // end - actions
) // end - top app bar
```

showMenu is a Boolean variable for deciding if the menu should be displayed

Toggle showMenu when action icon pressed

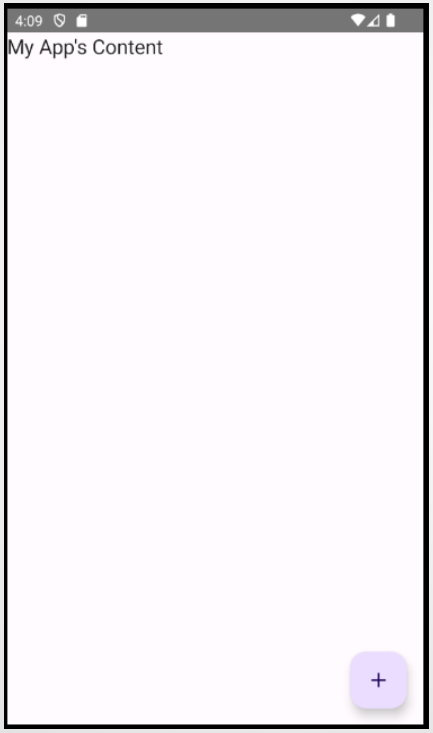
Menu icon. Three vertical dots are being used in this case

Drop down menu. It will cause menu items to appear below it if pressed. The showmenu variable decides if it should display the menu.

Add DropdownMenu items for each the action in the menu (only one in this example)

Floating Action Button (FAB)

- A high-emphasis button for performing a primary action in your app.
- It is located at the bottom right of the screen.
- For example:



Floating action for adding an item

Floating Action Button (FAB)

- A floating action button should be displayed in a Scaffold.

```
Scaffold(
    floatingActionButton = { MyFAB() },
    content = {
        Column(
            modifier = Modifier
                .padding(it)
                .fillMaxSize(),
        ) {
            Text(
                text = "My App's Content",
                fontSize = 20.sp
            )
        } // end - Column
    } // end - content
)
```

The floatingActionButton parameter of the Scaffold takes a composable function as a parameter (myFAB() function shown on the next slide)

- A floating action button should be displayed in a Scaffold.

```
@Composable
fun MyFAB() {

    FloatingActionButton(
        onClick = {
            // FAB click handler code goes here
        },
    ) {
        Icon(Icons.Filled.Add, "Floating action button.")
    }

}
```

Add event handler code for when the FAB icon is pressed here

FAB icon to display

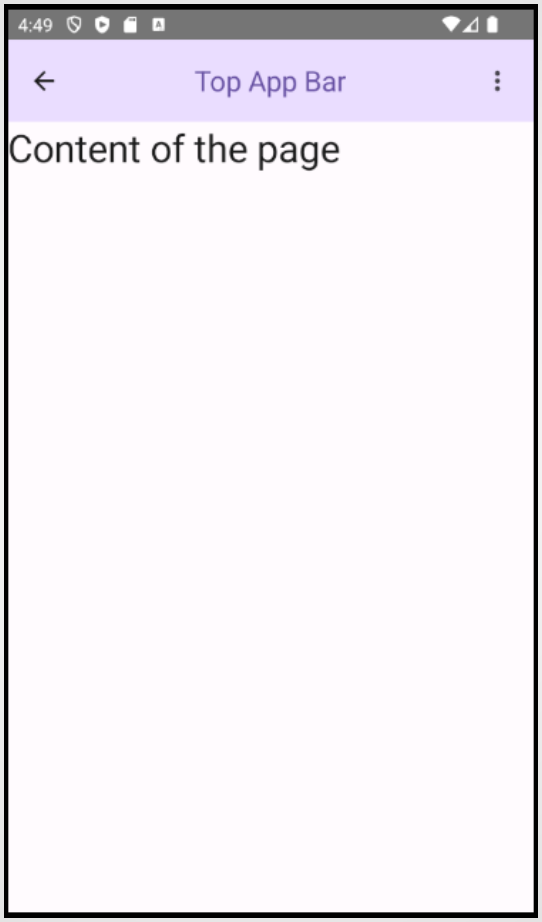
Floating Action Button (FAB)

- End of slides.

End of Slides

- Here is a sample TopAppBar:

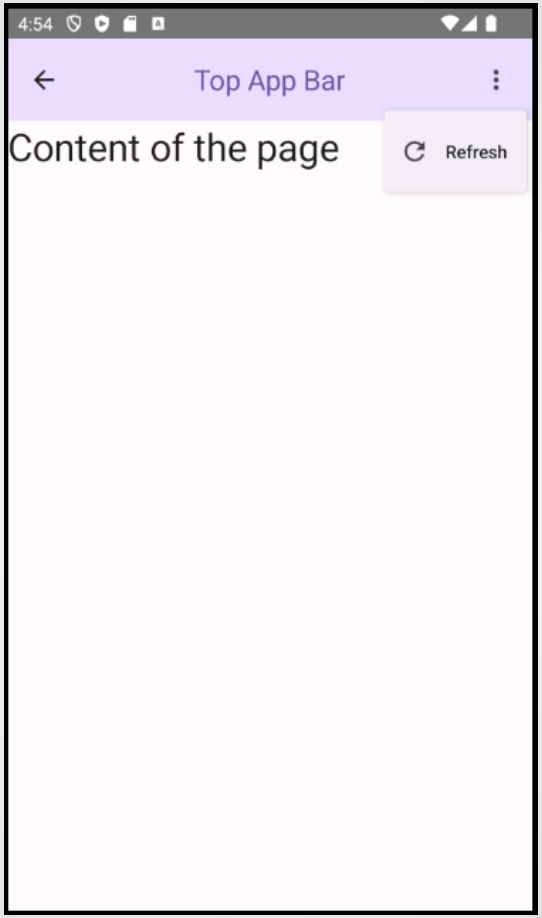
The arrow is the navigationIcon (generally used to go to the previous screen)



The three vertical dots are an action (you can add a few more actions here if you want)

TopAppBar

- Here is a sample TopAppBar after pressing action:



Pressing this three vertical dots shows a DropDownMenu appear (check code on next slide for how to make this happen)

There can be more than one DropDownMenuItem inside the menu

TopAppBar

```
var showMenu by remember { mutableStateOf(false) }
CenterAlignedTopAppBar (
    title = { Text(text = "Top App Bar") },
    colors = TopAppBarDefaults.topAppBarColors(
        containerColor = MaterialTheme.colorScheme.primaryContainer,
        titleContentColor = MaterialTheme.colorScheme.primary,
    ),
    navigationIcon = {
        IconButton(onClick = { // onClick event handler code goes here } ) {
            Icon(Icons.Filled.ArrowBack, "backIcon")
        }
    },
    actions = {
        IconButton(onClick = { showMenu = !showMenu }) {
            Icon(Icons.Default.MoreVert, contentDescription = null)
        }
        DropdownMenu(
            expanded = showMenu,
            onDismissRequest = { showMenu = false }
        ) {
            DropdownMenuItem(
                text= { Text("Refresh") },
                onClick =  { showMenu=false
                    // Other onClick code goes here },
                leadingIcon = { Icon(Icons.Filled.Refresh, contentDescription = null) })
        }
    }
)
```

The showMenu variable tells it whether to show or hide the menu

navigationIcon appears on left side of top app bar

IconButton to show/hide the menu (it has 3 vertical dots). When it's clicked it changes showMenu to the opposite value

DropdownMenu is the menu that appears

Add other DropdownMenuItems as necessary.

The DropDownMenuItem.onClick code can use a NavController to navigate to another screen.

Code to Define a TopAppBar

```
@Composable
fun mainScreen(navController: NavController) {

Scaffold(
        topBar = {
          // TopAppBar code goes here
        },

        content = {
           // Main screen content code goes here
        }
    )
}
```

Put TopAppBar code from previous slide here

Use TopAppBar in a Screen

- End of Slides

End of Slides