BCS 371 Lab – LazyColumn

Overview

Create a composable function that uses a LazyColumn to efficiently display a list of items.

Create a project

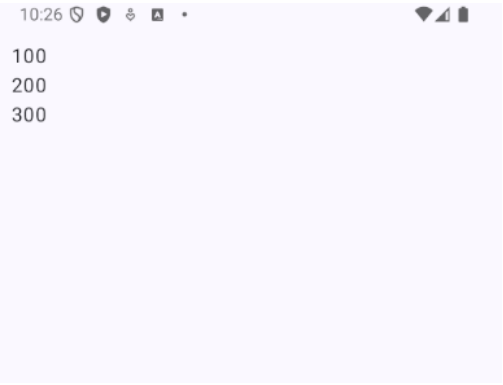
Create a new Android application in Android Studio. Choose the **Empty Activity** type to create an empty activity that uses Jetpack Compose.

Setup the Main Screen for Numbers

Create a Kotlin file name MainScreenNums.kt. Add a composable function named MainScreenNums. Here is the function header:

```
@Composable
fun MainScreenNums(modifier: Modifier = Modifier)
```

It should look like the following:



More specifications for this composable function:

- Declare a var and initialize a mutable state list of numbers. Use remember and mutableStateListOf to create the instance. Add instances of numbers to the list that contains the data shown above.
- Add a LazyColumn that will display each item in a Text composable.
- Update MainActivity to call MainScreenNums().

Run the App

Make sure all values in the list appear when the app runs.

Create Song Class

Create a class named Song. Add member properties for name (String) and artist (String).

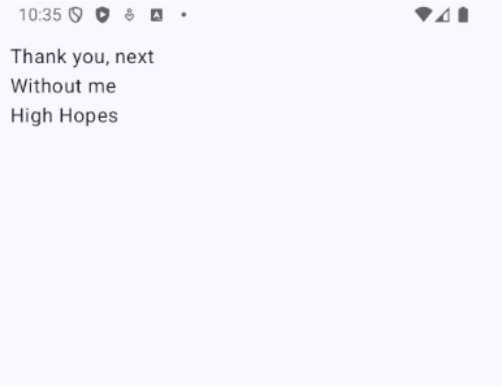
Note: Make sure to choose New|Kotlin Class/File menu option (not Java Class).

Setup the Main Screen for Songs

Create a Kotlin file name MainScreenSongs.kt. Add a composable function named MainScreenSongs. Here is the function header:

```
@Composable
fun MainScreenSongs(modifier: Modifier = Modifier)
```

It should look like the following:



More specifications for this composable function:

- Declare a var and initialize a mutable state list of songs. Use remember and mutableStateListOf to create the instance. Add instances of Song to the list that contains the data shown above.
- Add a LazyColumn that will display each song's name in a Text composable.
- Update MainActivity to call MainScreenSongs().

Run the App

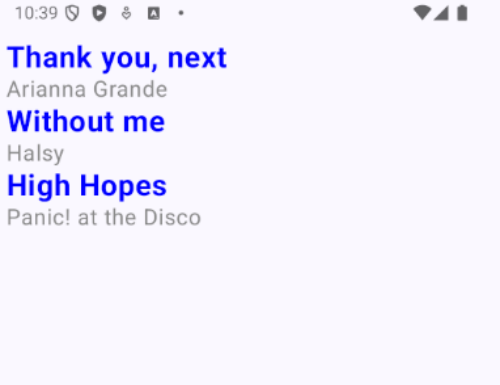
Make sure all values in the list appear when the app runs.

Update Main Screen for Songs

It should now show each song's name and artist in separate Text composables. Update the appearance as follows:

- Make the song's name appear as follows: color blue, bold, 24.sp font size.
- Make the song's artist appear as follows: color gray, 18.sp font size.

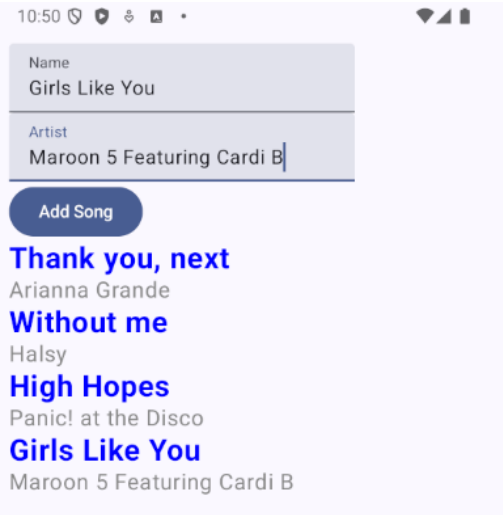
It should look like the following:



Add Song Data Entry Functionality to the App

Add song data entry functionality to the app. You will need to add two TextFields and one Button to the layout. When the user presses the button a new Song instance should be created and added to the underlying Song list (using the data in the TextFields). The LazyColumn should display the new song.

Here is a screenshot after the user types in new song data and presses the Add Song button:



Use a ViewModel to Store List Data

In this section we will update the app so that it uses a view model to hold song data (instead of a local variable in the composable function). Do the following:

- Add a view model to the app. Call it MainScreenViewModel.
- Move the song list declaration to the view model (instead of the composable function). You can declare it as follows:
  var songList = mutableStateListOf<Song>()

- Initialize the song list in the view model. Use the init block inside the view model class to do this.
- Create a new main screen composable function named MainScreenSongsViewModel. Copy the contents of MainScreenSongs and use that as a starting point. Here is the function header:
  @Composable
  fun MainScreenSongsViewModel(modifier: Modifier = Modifier)
- Add a method to the new model that adds a song to the song list. Here is the function header:
  fun addSong(name:String, artist: String)
- Update MainScreenSongsViewModel
  - It should use the song list declared in the view model instead of the local variable.
  - It should call the view model's addSong method to add a new song to the song list.
- Update MainActivity to call MainScreenSongsViewModel().

The screen should look the same (no changes to UI display).

Clear Name and Artist

The above version of the app does not clear the name and artist TextFields after the new song is added to the list. Add code to clear both TextFields after the new song is added.

Make the Items in the LazyColumn Handle Click Events

Display a toast with the clicked item's data. After clicking High Hopes, it should look like the following: