5COSC023W - MOBILE APPLICATION DEVELOPMENT
Lecture 5: Some Loose Ends, Dialogs and Creating Lists with Selectable Items

Dr Dimitris C. Dracopoulos

Sending Data from one Activity to Another

In the Sending Activity:

1. Create the Intent.
2. Set data or put extra data in the Intent.
3. Start the receiving (new activity) with `startActivity(intent)`.

In the Receiving Activity:

1. Get the Intent that created the Activity.
2. Retrieve the data or extras from the Intent.

Examples

```
// Setting data and extras
intent.setData(Uri.parse("http://www.google.co.uk"));
intent.setData(Uri.parse("tel:02079115000"));
intent.putExtra("Score", 56345);

// Retrieving data and extras
Uri url = intent.getData();
int score = intent.getIntExtra("score", 0);
```

A Full Example of 2 Communicating Activities

Activity 1:

```
package uk.ac.westminster.activitycommunicationcomposable

import android.content.Intent
import android.net.Uri
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.material3.Button
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.text.style.TextAlign

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContent {
            GUI()
        }
    }
}
```

A Full Example of 2 Communicating Activities (cont'd)

```
@Composable
fun GUI() {
    val context = LocalContext.current

    Button(onClick = {
        val i = Intent(context, Activity2::class.java)
        i.setData(Uri.parse("tel:02079115000"))
        i.putExtra("Score", 5200)
        context.startActivity(i)
    }) {
        Text("Send data", textAlign = TextAlign.Center,
             modifier = Modifier.fillMaxWidth())
    }
}
```

A Full Example of 2 Communicating Activities (cont'd)

Activity 2:

```
package uk.ac.westminster.activitycommunicationcomposable
import android.net.Uri
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.sp

class Activity2 : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContent {
            GUI2()
        }
    }

    // Note that GUI2 is defined inside the activity class here
    @Composable
    fun GUI2() {
        // extract the data sent by Activity1 from the intent
        var uri: Uri? = intent.data
        var score = intent.getIntExtra("Score", 0)

        // display the extracted data
        Text("Received: \n $uri \n Score: $score", fontSize = 24.sp,
             modifier = Modifier.fillMaxSize(), textAlign = TextAlign.Center)
    }
}
```

How to Create a Dialog (Popup Window)

Different ways but mostly the following can be used:

- `Dialog`
- `AlertDialog`

An Example of Creating a Dialog

```
package uk.ac.westminster.alertdialogcomposableexample

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.Column
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.Info
import androidx.compose.material3.AlertDialog
import androidx.compose.material3.Button
import androidx.compose.material3.Icon
import androidx.compose.material3.Text
import androidx.compose.material3.TextButton
import androidx.compose.runtime.Composable
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            GUI()
        }
    }
}
```

An Example of Creating a Dialog (cont'd)

```
// An example of an alert dialog, pop up window
@Composable
fun GUI() {
    // is the pop up window open?
    var openDialog by remember { mutableStateOf(false) }


    Column {
        Text("My app is great")
        Button(onClick = { openDialog = true }) { Text("Open dialog") }
    }
    if (openDialog) {
        AlertDialog(
            icon = {
               Icon(Icons.Default.Info, contentDescription = "Example Icon")
            },
            title = {
                Text(text = "Title")
            },
            text = {
                Text(text = "dialogText")
            },
            // what happens when you click outside the dialog
            onDismissRequest = {
                openDialog = false
            },
            confirmButton = {
                TextButton(
                    onClick = {
                        openDialog = false
                    }
                ) {
                    Text("Confirm")
                }
            },
```

An Example of Creating a Dialog (cont'd)

```
            dismissButton = {
                TextButton(
                    onClick = {
                        openDialog = false
                    }
                ) {
                    Text("Dismiss")
                }
            }
        )
    }
}
```

How to Create a List with Selectable Items

For displaying many entries in a list visually, use `LazyColumn` instead of `Column`.

- It is more efficient
- Otherwise (depending on how many items the app attempts to display), your application might freeze or crash.

An Example of a List with Selectable Items

```
package com.example.listwithselectableitemscomposable

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.runtime.Composable
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.selection.selectable
import androidx.compose.material3.Text
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            GUI()
        }
    }
}
```

An Example of a List with Selectable Items (cont'd)

```
@Composable
fun GUI() {
    var selected by remember{ mutableStateOf("") }

    // Create an empty list
    var myList = mutableListOf<String>()

    // populate the list with some items
    for (i in 1..100)
        myList.add("Item $i")

    Column(modifier = Modifier.fillMaxSize(),
           horizontalAlignment = Alignment.CenterHorizontally) {
        Text("Menu", fontSize = 36.sp, fontWeight = FontWeight.Bold)
        LazyColumn(Modifier.height(500.dp)) {
            for (i in myList)
                item {
                    Row(
                        modifier = Modifier
                            .fillMaxWidth()
                            .clickable(onClick = {
                                selected = i
                            })
                    )
                    {
                        Text(text = i, fontSize = 24.sp)
                    }
                }
        }

        Text(text = "This is what was just selected: $selected",
                     modifier = Modifier.padding(top = 20.dp))
    }
}
```