BCS 371 Mobile Application Development I

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

Today’s Lecture

- Intents

Navigating to Screens in Older Android UI Development

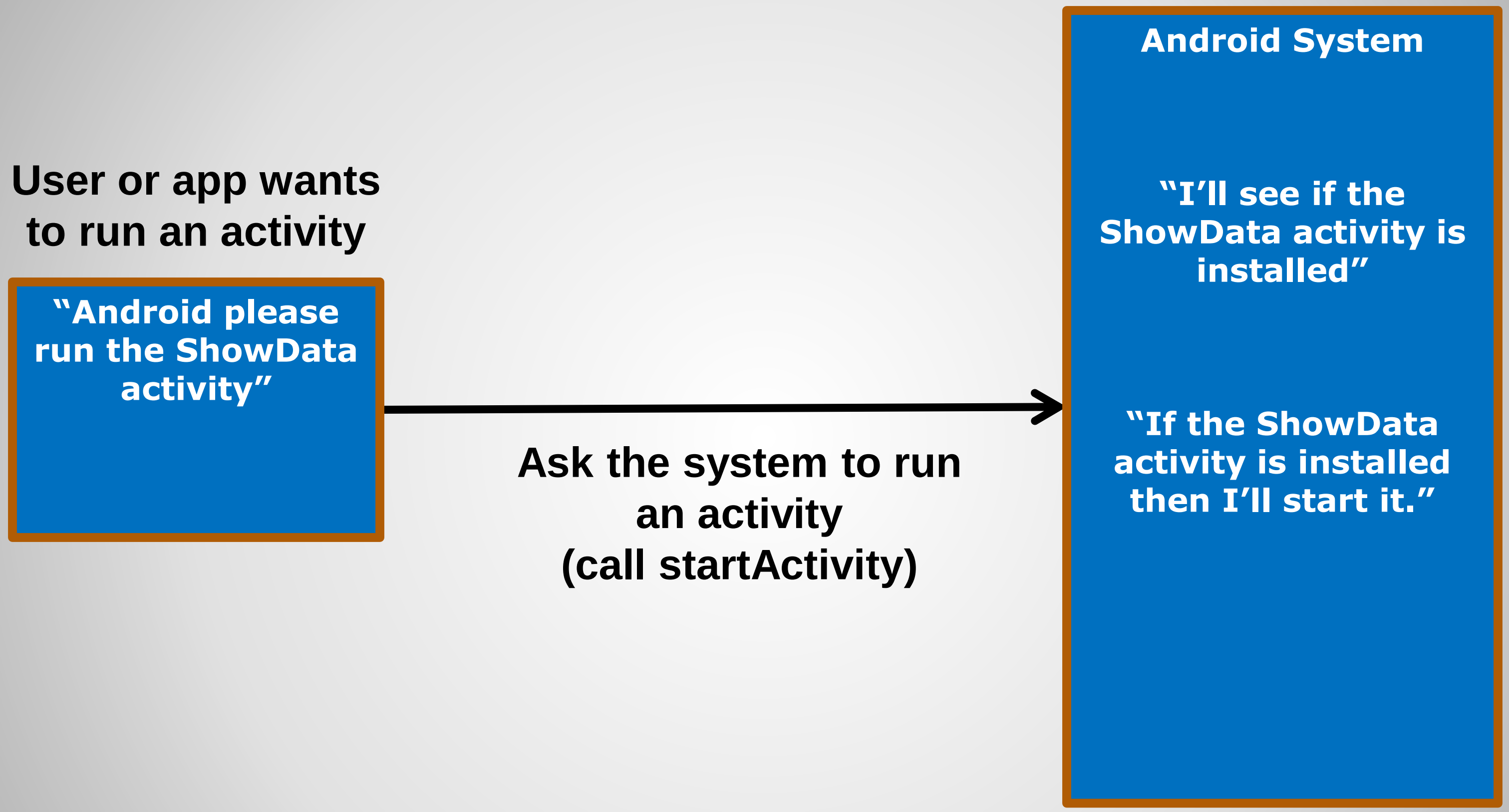
- The older style of Android UI development associated each screen with an activity.

- When the user needed to navigate to another screen another activity was started.
- Use the startActivity function to do this.
- The startActivity function asks Android to start the other activity for us.

- This is still useful with Jetpack Compose in some circumstances.
- For example, if we need to start other apps (like the camera app).

Starting an Activity

- How does an activity run in Android?



User or app wants to run an activity

"Android please run the ShowData activity"

Ask the system to run an activity (call startActivity)

Android System

"I'll see if the ShowData activity is installed"

"If the ShowData activity is installed then I'll start it."

Intents

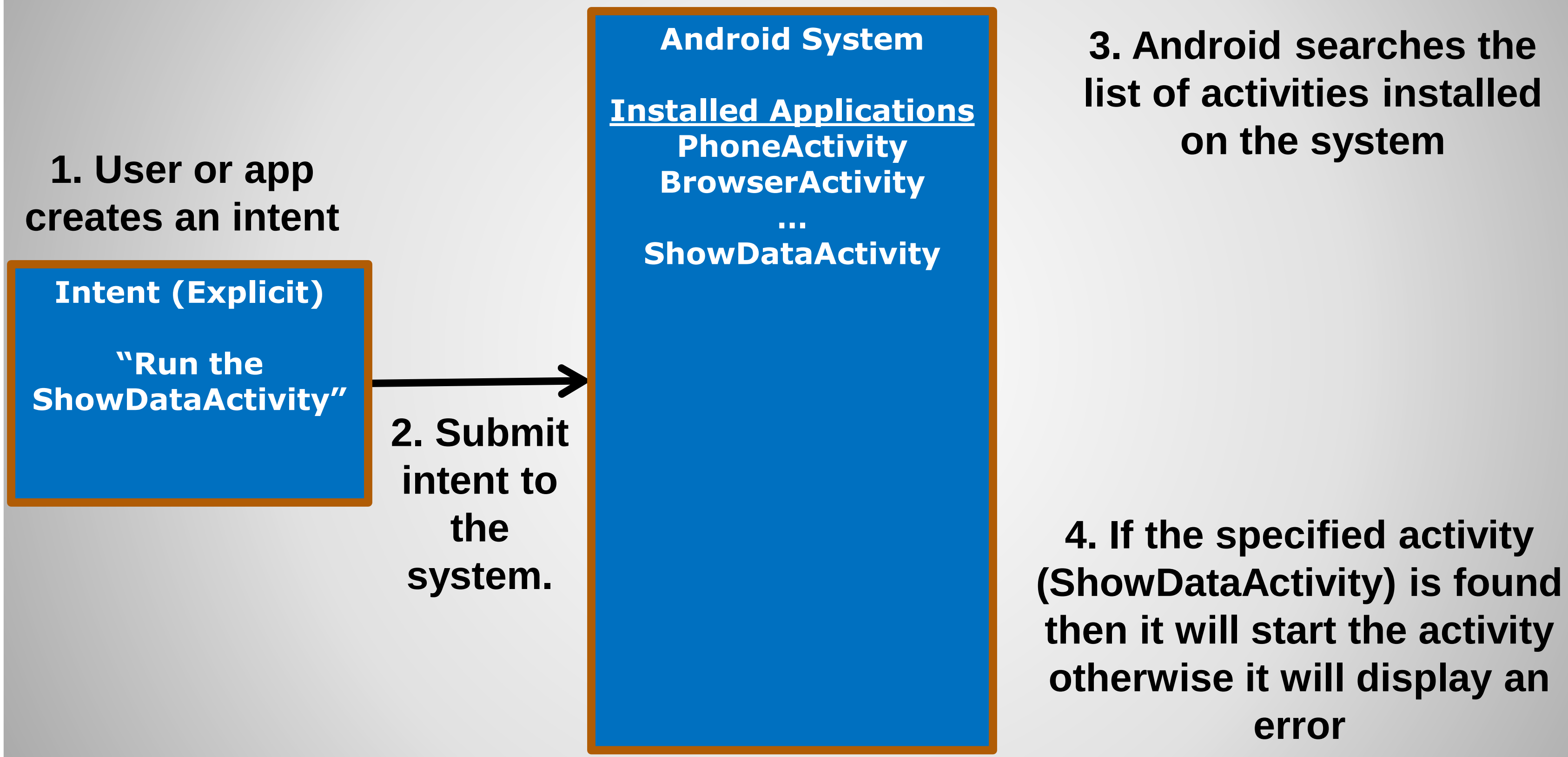
- The startActivity function must be passed an intent.

- An **intent** describes something that needs to get done (run an activity or perform some kind of action).
- From the Android website: "An intent is an abstract description of an operation to be performed." Here is the link: https://developer.android.com/reference/android/content/Intent

- The intent contains data that the system uses to fulfill the request.
- The system action of finding an activity to fulfill an intent is called **intent resolution**.

- Two main types of intents:
  - **Explicit**
  - **Implicit**

- A user or app needs to run a specific activity



1. User or app creates an intent

Intent (Explicit)

"Run the ShowDataActivity"

2. Submit intent to the system.

Android System

Installed Applications

PhoneActivity

BrowserActivity

…

ShowDataActivity

3. Android searches the list of activities installed on the system

4. If the specified activity (ShowDataActivity) is found then it will start the activity otherwise it will display an error

Using an Intent

- Intents are handled in different ways.

- **Explicit** – Indicates the exact activity to use. If the requested activity is not there, then nothing will run.

- **Implicit** – Lets the system decide which activity will best fulfill the request (intent resolution). If multiple activities can fulfill the request, then the user is presented with a list to choose from.

Implicit vs Explicit Intents

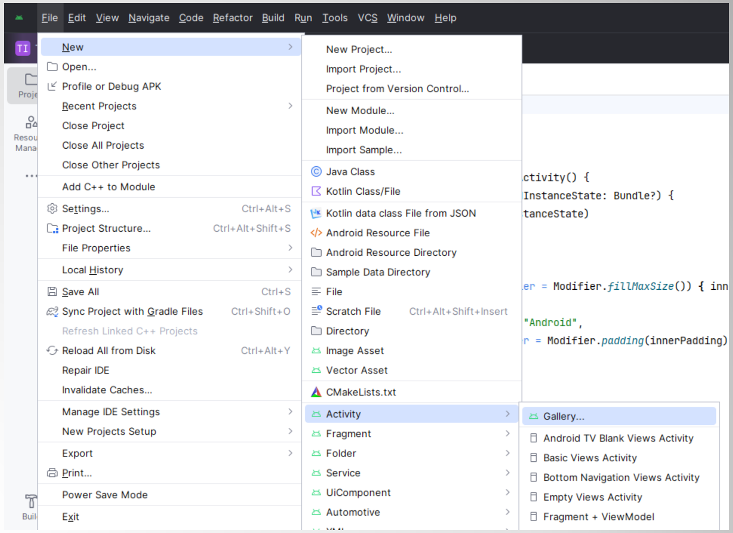
Explicit Intent

- Now on to explicit intents…

- An **explicit intent** starts a particular activity.
- You indicate the exact activity class that should start.
- It does **NOT** go through the intent resolution process.
- Here are instructions for creating a new activity and starting it using an explicit intent…

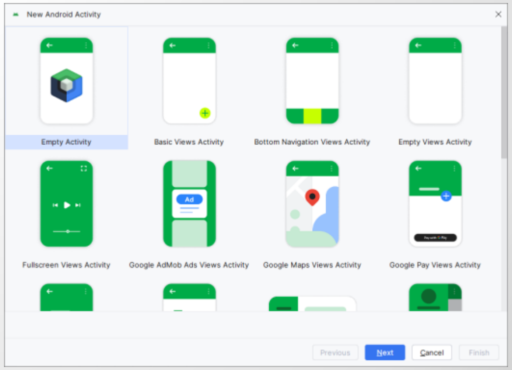
Explicit Intent

- Go to: File|New|Activity|Gallery.



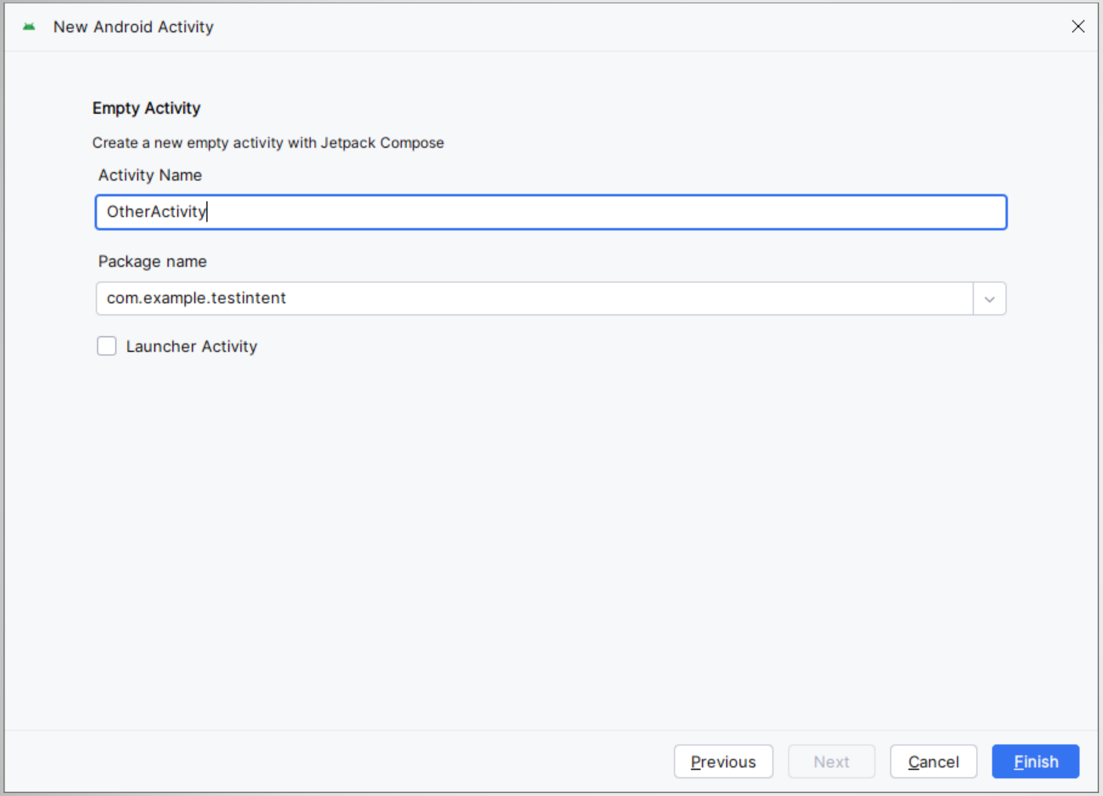
```
ctivity() {
InstanceState: Bundle?) {
tanceState)

er = Modifier.fillMaxSize()) { inn

"Android",
r = Modifier.padding(innerPadding)
```

- Choose Empty Activity.


New Android Activity
Empty Activity
Basic Views Activity
Bottom Navigation Views Activity
Empty Views Activity
Fullscreen Views Activity
Google AdMob Ads Views Activity
Google Maps Views Activity
Google Pay Views Activity
Previous
Next
Cancel
Finish

Create a New Activity in Android Studio

- Give the activity a name and choose finish:


New Android Activity

Empty Activity

Create a new empty activity with Jetpack Compose

Activity Name

OtherActivity

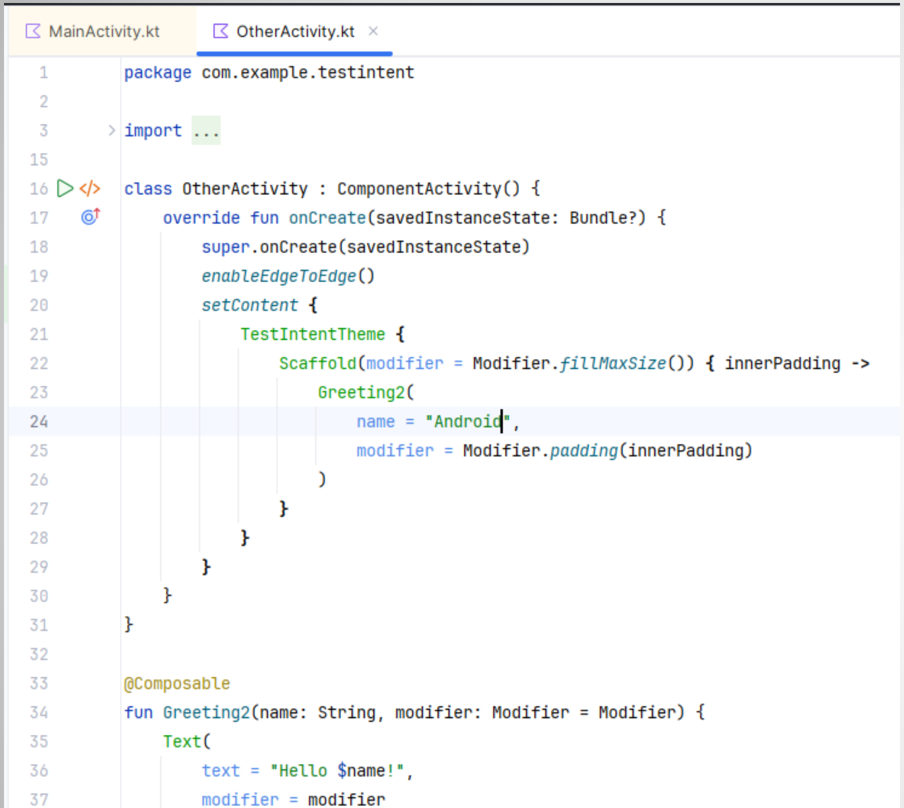
Package name

com.example.testintent

□ Launcher Activity

Create a New Activity in Android Studio

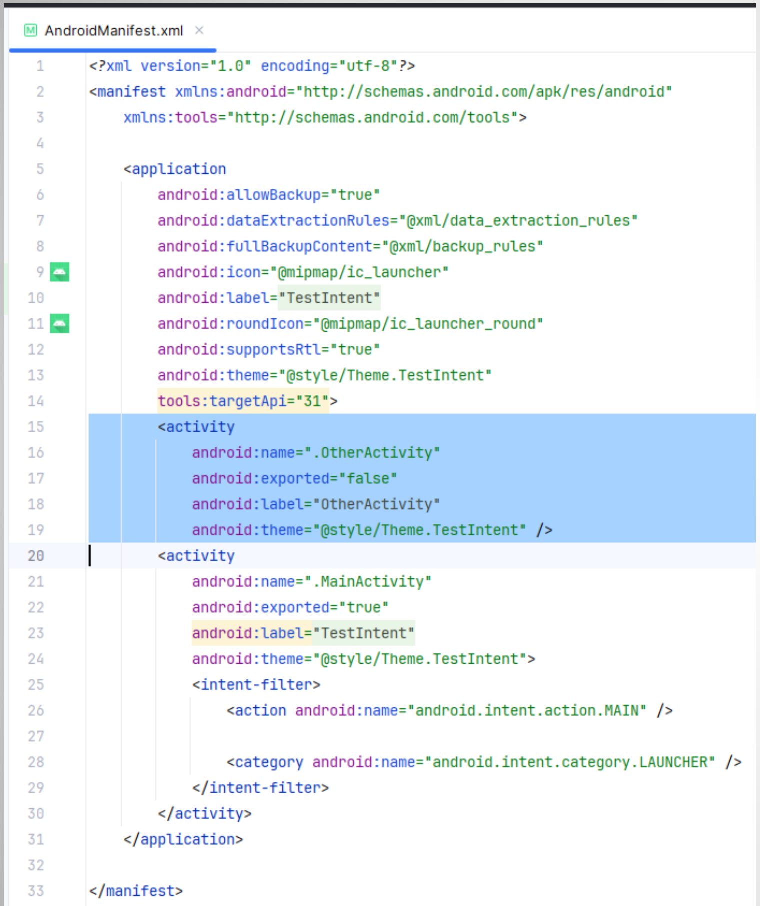
- Here is the code that was created for OtherActivity:



```kotlin
package com.example.testintent

import ...

class OtherActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            TestIntentTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    Greeting2(
                        name = "Android",
                        modifier = Modifier.padding(innerPadding)
                    )
                }
            }
        }
    }
}

@Composable
fun Greeting2(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
        modifier = modifier
```

Create a New Activity in Android Studio

- Register new activity in the manifest:



AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="TestIntent"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.TestIntent"
        tools:targetApi="31">
        <activity
            android:name=".OtherActivity"
            android:exported="false"
            android:label="OtherActivity"
            android:theme="@style/Theme.TestIntent" />
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="TestIntent"
            android:theme="@style/Theme.TestIntent">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Each activity in the app must be registered in AndroidManifest.xml. An <activity> element is automatically added to the manifest when creating an activity using File|New|Activity.

Here is the <activity> element for OtherActivity.

Register Activity in Manifest

Start Activity Using an Explicit Intent – INSIDE Composable

StartActivity – INSIDE a Composable Function

- First, create an Intent instance for the activity you want to start.
- Second, call startActivity passing the intent.

```
val context = LocalContext.current

// Create the intent
val i = Intent(context, OtherActivity::class.java)

// Execute
context.startActivity(i)
```

Name of activity to start

This code creates and runs an explicit intent

Start Activity Using an Explicit Intent – OUTSIDE Composable

StartActivity – OUTSIDE a Composable Function

- First, create an Intent instance for the activity you want to start.
- Second, call startActivity passing the intent.

Current Activity

Name of activity to start

```
// Create the intent
val i = Intent(this@MainActivity, MyOtherActivity::class.java)

// Execute
startActivity(i)
```

This code creates and runs an explicit intent

Explicit Intent Example Code

```
val context = LocalContext.current

// Create the intent
val i = Intent(context, OtherActivity::class.java)

// Execute
context.startActivity(i)
```

What would happen if the class MyOtherActivity is properly defined in MyOtherActivity.java but was NOT registered in AndroidManifest.xml?

Explicit Intent Example Code

```
val context = LocalContext.current

// Create the intent
val i = Intent(context, OtherActivity::class.java)

// Execute
context.startActivity(i)
```

What would happen if the class MyOtherActivity is properly defined in MyOtherActivity.java but was NOT registered in AndroidManifest.xml?

***ANSWER*: App runs but throws a runtime exception when startActivity executes.**

Passing Data Using Intents

- Now on to passing data using intents…

Passing Data Using Intent – INSIDE Composable

Put Data into Intent and Start Other Activity

- Use the putExtra methods to store data in an Intent.
- Data is stored in a map inside the intent (key/value pairs).

```
// This code is in MainActivity
val context = LocalContext.current


var studentId = 100
var intent = Intent(context, MyOtherActivity::class.java)
intent.putExtra("id", studentId)
context.startActivity(intent);
```

Put data in the intent. Uses "id" as the key. The key "id" will be used to retrieve the data from the other activity.

Passing Data Using Intent – INSIDE Composable

Get Data from an Intent

- Define a function to find the activity:

```
fun Context.findActivity(): Activity? = when (this) {
    is Activity -> this
    is ContextWrapper -> baseContext.findActivity()
    else -> null
}
```

Need to include this function to find the activity (called in the code below)

- Retrieving data from an Intent in the other activity

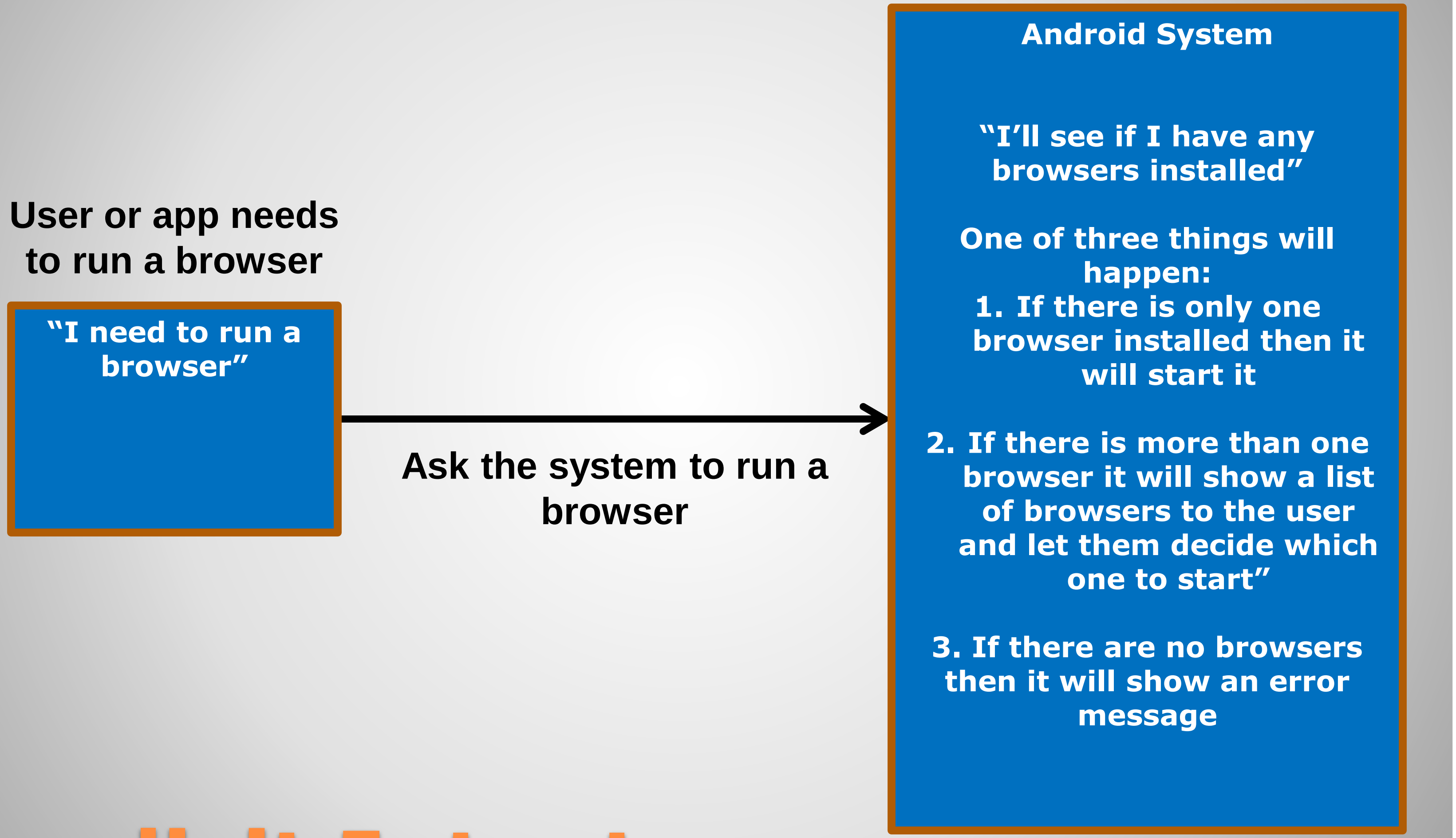
```
// This code is in MyOtherActivity
val context = LocalContext.current
val activity = context.findActivity()
val intent = activity?.intent

var id = intent!!.getIntExtra("id", 0)
```

Uses the key "id" to retrieve the data. If there is no key "id" then the default value 0 is used.

Implicit Intent

- Now on to implicit intents…

- I need to run a browser



User or app needs to run a browser

"I need to run a browser"

Ask the system to run a browser

Android System

"I'll see if I have any browsers installed"

One of three things will happen:

1. If there is only one browser installed then it will start it

2. If there is more than one browser it will show a list of browsers to the user and let them decide which one to start"

3. If there are no browsers then it will show an error message

Implicit Intent

Implicit Intent

- An **implicit intent** will cause the system to run a program that matches the information in the given intent.
- If multiple activities match the intent then the system will display a dialog to the user and they will choose which program to use.
- For example, you may need to run a browser or messaging application.
- Android will search the system for an installed application that can satisfy the description given in the intent.
- Here is an example implicit intent…

Implicit Intent – INSIDE Composable

```
val context = LocalContext.current

// Create the intent
val i = Intent(Intent.ACTION_DIAL,
          Uri.parse("tel:555-2368"))

// Execute
context.startActivity(i)
```

Need to use a phone app

This is the telephone number that I want to appear in the dialer

Implicit Intent – OUTSIDE Composable

```
// Create the intent
val i = Intent(Intent.ACTION_DIAL,
          Uri.parse("tel:555-2368"))

// Execute
startActivity(i)
```

Need to use a phone app

This is the telephone number that I want to appear in the dialer

Implicit Intent

```
// Create the intent
val i = Intent(Intent.ACTION_DIAL,
          Uri.parse("tel:555-2368"))

// Execute
startActivity(i)
```

Need to use a phone app

This is the telephone number that I want to appear in the dialer

What would happen if no apps on the device match this intent?

Implicit Intent

```
// Create the intent
val i = Intent(Intent.ACTION_DIAL,
          Uri.parse("tel:555-2368"))

// Execute
startActivity(i)
```

Need to use a phone app

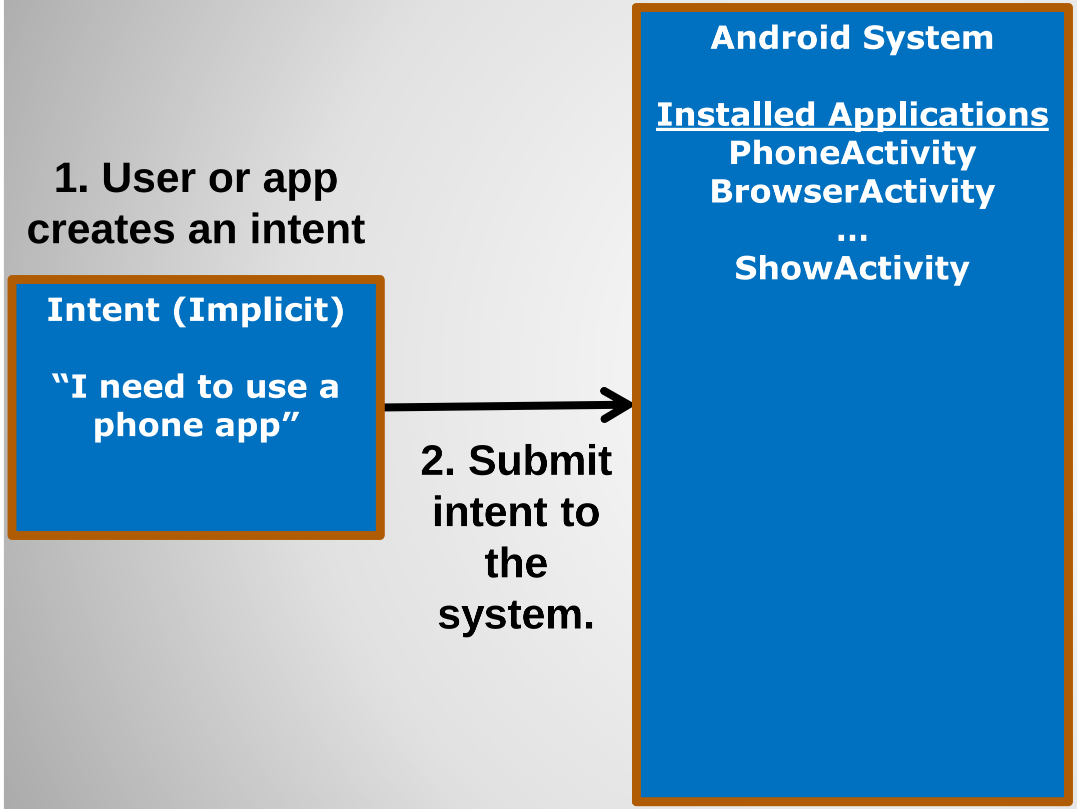
This is the telephone number that I want to appear in the dialer

What would happen if no apps on the device match this intent?

***ANSWER*: App runs but throws a runtime exception when startActivity executes.**

Using an Intent

- A user or app needs to run a phone app



1. User or app creates an intent

2. Submit intent to the system.

3. Android searches the list of activities installed on the system to see if one qualifies as a phone app

4. If an activity is found that is a phone app then it is started. If more than one app is a phone app then it will present the user with a list and let them decide which to run

Intent Resolution

- **Intent resolution** happens when an implicit intent is presented to the system.
- Android checks the list of installed apps to see if they satisfy the given intent.
- The installed apps have descriptions of themselves in the application manifest.
- These descriptions are what is “checked” when performing intent resolution.

- End of Slides

End of Slides