Displaying Lists & Activity Lifecycle

CSC2007 & ICT2105 Mobile Application Development Spring 2023

Overview

This lab provides guidelines to create a simple app to gain familiarity with the Android RecyclerView class, observe the Activity Lifecycle and configuration changes.

Outcomes

Upon completion of the session, you should be able to:

- Create a simple interactive list interface
- Understand the Activity lifecycle
- Handle configuration changes

Learn more about the Activity Lifecycle and RecyclerView

The following are structured exercises to learn about user navigation, fragments, the activity lifecycle and RecyclerViews. Follow the instructions and steps in the links below:

Fragments: https://codelabs.developers.google.com/codelabs/kotlin-android-training-create-and-add-fragment/index.html?index=..%2F..android-kotlin-fundamentals#3
Navigation: https://codelabs.developers.google.com/codelabs/kotlin-android-training-add-navigation/index.html?index=..%2F..android-kotlin-fundamentals#0
RecyclerView: https://developer.android.com/courses/pathways/android-basics-kotlin-unit-2-pathway-3

https://codelabs.developers.google.com/codelabs/kotlin-android-training-recyclerview-fundamentals/index.html?index=..%2F..android-kotlin-fundamentals#0

https://developer.android.com/codelabs/basic-android-kotlin-training-recyclerview-scrollable-list
https://github.com/android/views-widgets-samples/tree/main/RecyclerViewKotlin/

Activity Lifecycle: https://codelabs.developers.google.com/codelabs/kotlin-android-training-lifecycles-logging/index.html?index=..%2F..android-kotlin-fundamentals#0

https://codelabs.developers.google.com/codelabs/kotlin-android-training-complex-lifecycle/index.html?index=..%2F..android-kotlin-fundamentals#0

Creating a grocery list application

This section provides an exercise to create a simple grocery list application. It consists of a user interface to display a list of grocery items, each with a checkbox. The goal is to display an interactive checklist of grocery list items, that allows an item to be checked and unchecked.

Fork the repo **ict2105-lab03-2023 or csc2007-lab03-2023** and inspect the code within the project.

In the activity_main.xml file for the MainActivity, add the correct elements so that this application can display a RecyclerView.

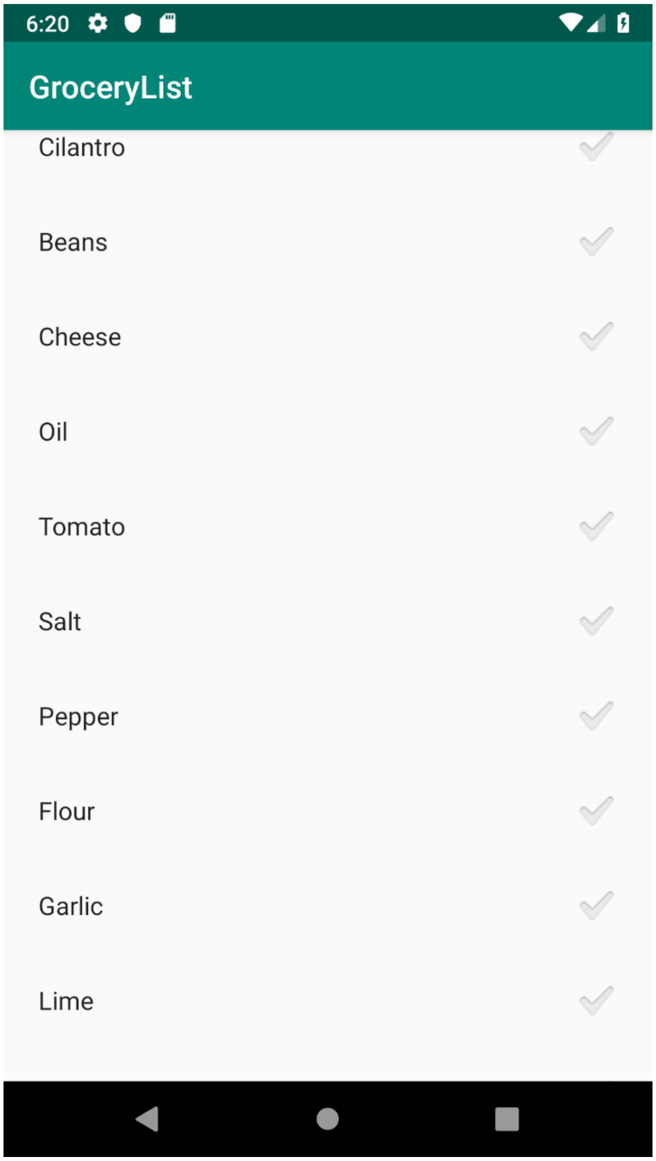
Design the layout of the screen for the MainActivity to be similar to the following screenshot. A **RecyclerView** and **CheckedTextView** has to be used.

The grocery list items are in the following order:

- Cilantro
- Beans
- Cheese
- Oil
- Tomato
- Salt
- Pepper
- Flour
- Corn
- Garlic
- Lime
- Onion
- Rice
- Cabbage
- Avocado

(continued on next page)

IMPORTANT: Ensure that the RecyclerView id is named recyclerViewGroceryList within the XML. Ensure there are no spelling errors or omissions on the grocery list items.


6:20

GroceryList

Cilantro ✓

Beans ✓

Cheese ✓

Oil ✓

Tomato ✓

Salt ✓

Pepper ✓

Flour ✓

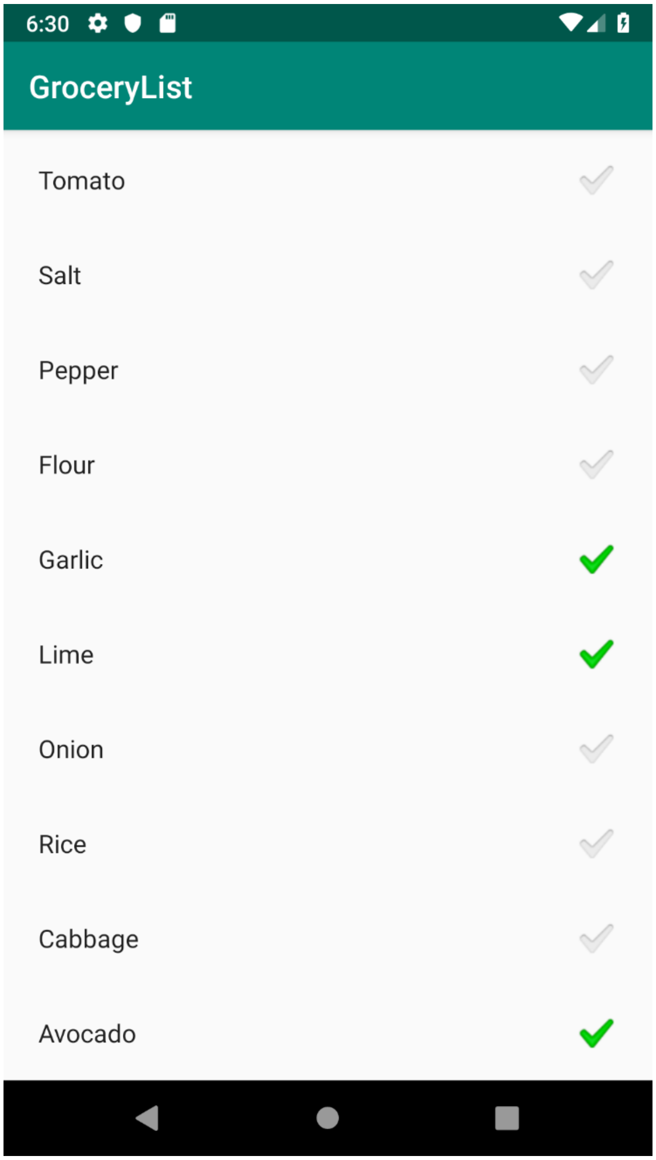
Garlic ✓

Lime ✓

When MainActivity is first loaded, all the grocery items are displayed in a RecyclerView. All the check boxes are <u>not checked</u>.

Implement the logic to display the grocery list in a RecyclerView, and then implement logic to keep track of which items are checked and unchecked.

IMPORTANT: Ensure that the list item id is named groceryListTextView within the XML.



6:30

GroceryList

Tomato ☐
Salt ☐
Pepper ☐
Flour ☐
Garlic ☑
Lime ☑
Onion ☐
Rice ☐
Cabbage ☐
Avocado ☑

When the device is rotated, the app must save and restore the list scroll position and check box items. How do you do this? Test this by rotating the emulator using the buttons or rotating the physical device.

Observe the Activity Lifecycle

On MainActivity, override all the activity lifecycle callback methods. (There are 6 in total, which ones are they?) The system invokes each of these callbacks as an activity enters a new state.

Add a debug log message within each of the activity callback methods with tag name as "MainActivity", and the callback name as the message. It should look something like the following in logcat:

```
2023-01-28 02:35:22.552 14645-14645/edu.singaporetech.grocerylist
D/MainActivity: onCreate()
2023-01-28 02:35:22.563 14645-14645/edu.singaporetech.grocerylist
D/MainActivity: onStart()
2023-01-28 02:35:22.569 14645-14645/edu.singaporetech.grocerylist
D/MainActivity: onResume()
```

Totally close and kill the app (not suspend, but kill the app from Settings):

- Navigate to the Apps tab in "Settings"
- Go to the "All apps" tab, scroll to the app, and click on it.
- Tap "Force Stop" to kill the process.

From a vertical portrait position, do the following in sequential steps:

- Launch the app
- Rotate the device 90 degrees left to landscape
- Then rotate the device 90 degrees vertical again.

What are the order of the lifecycle callbacks? Put the logcat output of lifecycle calls in a text file called **`lifecycle.txt`** in the root directory of the repo. (delete non-lifecycle log output from the text file)

Write an explanation for the lifecycle callbacks at the end of the text file **`lifecycle.txt`**

Lab Exercise 3

Due Date: Wed Feb 1, 2023 2359 hrs

1. Fork the repo **ict2105-lab03-2023** or **csc2007-lab03-2023.**
2. Implement the missing tags within activity_main.xml to display the list.
3. Design the layout of the screen similar to the given screenshots. Use a **RecyclerView**. Ensure there are no spelling errors on all grocery item text and that everything is in the same order. The id for the RecyclerView should be named **recyclerViewGroceryList**.
4. Implement the logic on MainActivity screen, to display the grocery list, and the logic to keep track and display which items are checked and unchecked.
5. Implement the logic to save and restore the list scroll position and check box items when the device is rotated.
6. Override all the activity lifecycle callback methods within MainActivity.
7. Add a debug log message within each of the activity callback methods with tag name as “MainActivity”, and the callback name as the message
8. Answers for rotation exercise, put the logcat output of lifecycle calls (only) in a text file called **`lifecycle.txt`** in the root directory of the repo.
9. Write an explanation for the lifecycle callbacks at the end of the text file **`lifecycle.txt`**
10. Commit and push all changes to your forked repository **ict2105-lab03-2023** or **csc2007-lab03-2023.**

END OF DOCUMENT