IT University of Copenhagen

Mobile App Development (Spring 2016)

Mandatory Assignment #1
Tingle App Version 4

Author:
Sjúrður í Sandagerði `sjri@itu.dk`

Teacher:
Jørgen Staunstrup
Radoslav Todorov

Teaching Assistant
Abdulrashid Mohammed

March 2nd, 2016

1 Tingle App

1.1 Layout

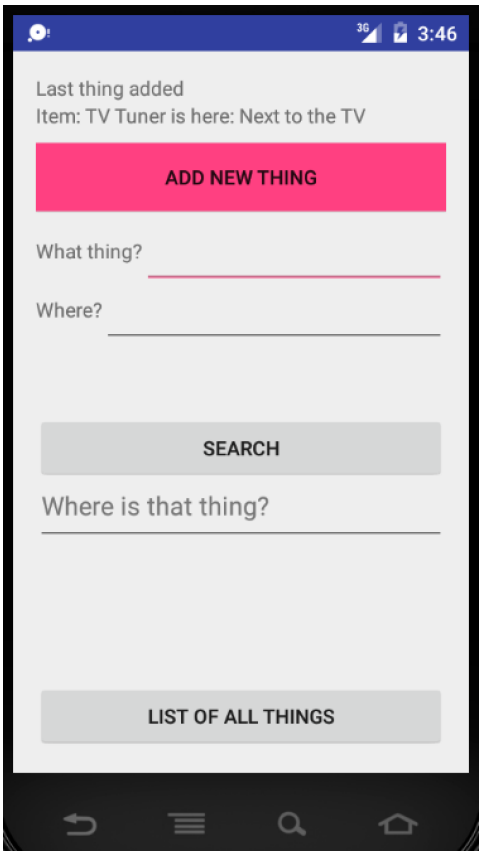

Figure 1.1: The main view (Tingle Activity) in portrait mode.

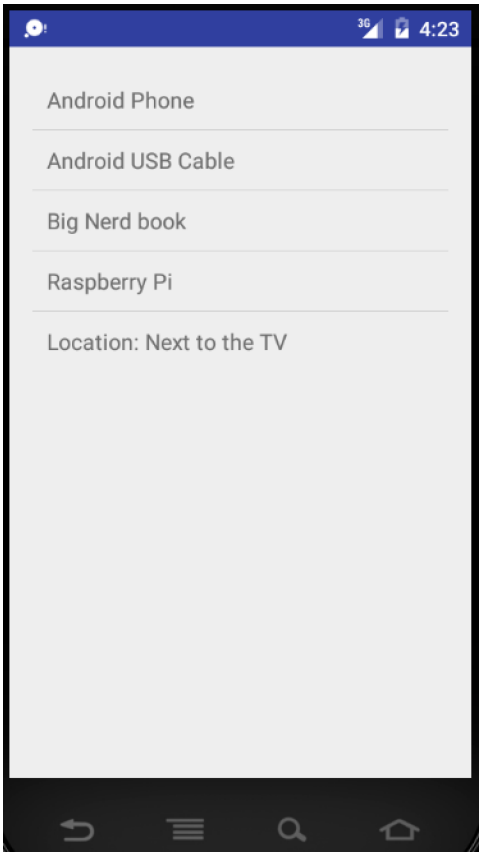

Figure 1.2: Portrait view of the List Activity.

When running the Tingle App the first view the user sees is shown in Figure 1.1. In this view the user can add new things and their whereabouts to the database, search for exists things and finally list all things in the database. Clicking the button `List of All Things` will open up the List Activity shown in Figure 1.2. The list view is a scrollable list showing all items in the database. Tapping an item in the list will toggle the *What* (i.e the name of the item) and *Where* (the location of the item). The user can also delete an item by Long pressing on an item. A popup dialog appears as seen in Figure 1.3.

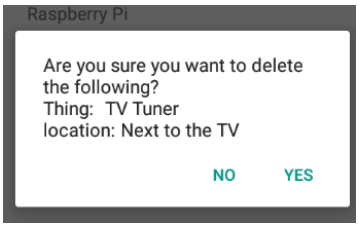

Figure 1.3: Verification dialog when deleting an item.

Changing the orientation of the device to landscape displays the two views, Tingle view and List view, side-by-side. The same functionality applies to the views in landscape mode as they do in portrait mode as described above. The only difference is the removal of the `List of All Things` button since the list is already visible to the right. This is a choice of design. One could also make it possible to open up the list in landscape view showing more information about each item.

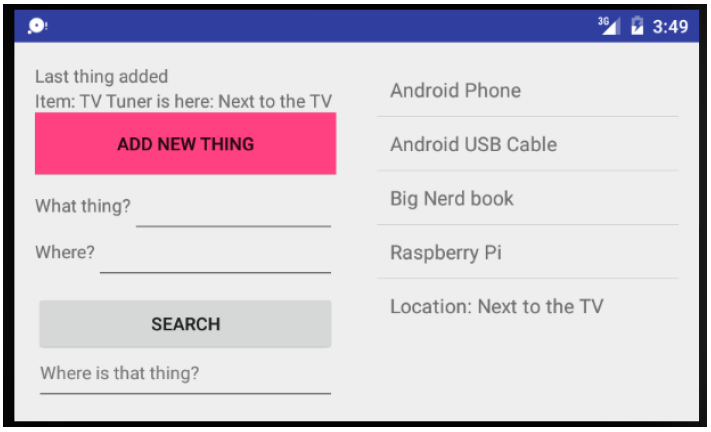

Figure 1.4: Landscape view showing both the main functions as well as the list of things side by side.

1.2 Design

The main design follows the convention explained in chapters 7 and 8 in the book *The Big Nerd Ranch Guide 2nd Edition* using fragments. I have separated the main view (`Tingle`) and the list view (`List`) into two different activities, (`TingleActivity` and `ListActivity`) respectively. When the user changes the orientation of the device how the fragments are loaded depends on which activity is currently running. In my design this causes some duplication in code since the layout is identical. I chose not to make a distinct layout for the list view in landscape mode which has resulted in the code being near duplicated between the two Activity classes. The fragments have no knowledge of each other in order to obtain a more loosely coupled design. The communication between fragments has been moved up to the parent Activity classes instead via interfaces. For the user to be able to toggle the name and location of an item I have implemented a custom TextView class called ToggledTextView. This class holds a single boolean member variable `isToggled` that is flipped each time the user clicks an item in the list to keep track which information is displayed (What vs Where).

1.3 Extensions

A search function has been implemented making it possible for the user to search for an item in the database. The user can access this function in the Tingle view. A delete function has also been implemented making it possible to remove items from the database. To delete an item the user can long-press an item until a verification dialog appears.

1.4 Testing

The testing of the Tingle App has been separated into four different parts using only black-box testing. One for each view and one for each orientation of the device. As can been seen in the tables below the cases are similar for Portrait mode and Landscape mode. This redundant testing is done to make sure there are no unforeseen bugs when changing the orientation.

ID	Test Case	Expected Output	Actual Output
A	Running the App	The application starts and the Tingle view is shown	The application starts and the Tingle view is shown
B	Clicking the `List of All Things` button opens up the List view.	The List view is shown.	The List view is shown.
C	It is possible to add an item to the database.	An item is successfully added to the database.	An item is successfully added to the database.
D	It is possible to search for an item in the database.	A Toast is displayed showing the location of the found item.	A Toast is displayed showing the location of the found item.
E	Clicking `List of All Things` opens up a the List view. Clicking the android back button navigates back to the Tingle view	User is successfully navigated back to the Tingle view.	User is successfully navigated back to the Tingle view.

Figure 1.5: Test cases of the Tingle view in Portrait mode.

ID	Test Case	Expected Output	Actual Output
A	Running the App	Both the Tingle view and List view are shown side-by-side.	Both the Tingle view and List view are shown side-by-side.
B	Clicking the `List of All Things` button in the Tingle view opens up the List view.	The List view is shown.	The List view is shown.
C	It is possible to add an item to the database.	An item is successfully added to the database.	An item is successfully added to the database.
D	It is possible to search for an item in the database.	A Toast is displayed showing the location of the found item.	A Toast is displayed showing the location of the found item.
E	Changes in Layout compared to Portrait mode	The button `Lost of All Things` is not present.	The button `Lost of All Things` is not present.

Figure 1.6: Test cases of the Tingle view in Landscape mode.

ID	Test Case	Expected Output	Actual Output
A	It is possible to click an item in the list	Toggles Name / Location of the item.	Toggles Name / Location of the item.
B	It is possible to Long click an item in the list.	A dialog is displayed asking if the user wants to delete the clicked item.	A dialog is displayed asking if the user wants to delete the clicked item.
C	When deleting an item the user can decline the request by pressing `No`.	The item is not deleted.	The item is not deleted.
D	When deleting an item the user can accept the request by pressing `Yes`.	The item is deleted from the list.	The item is deleted from the list.
E	It is possible to delete items one by one until the list is empty.	All items are deleted and the app does not crash.	All items are deleted and the app does not crash.
F	If there are many items in the database and not all items fit on the screen the user can scroll the list.	The list is scrollable.	The list is scrollable.

Figure 1.7: Test cases of the List view in Portrait mode.

ID	Test Case	Expected Output	Actual Output
A	It is possible to click an item in the list	Toggles Name / Location of the item.	Toggles Name / Location of the item.
B	It is possible to Long click an item in the list.	A dialog is displayed asking if the user wants to delete the clicked item.	A dialog is displayed asking if the user wants to delete the clicked item.
C	When deleting an item the user can decline the request by pressing `No`.	The item is not deleted.	The item is not deleted.
D	When deleting an item the user can accept the request by pressing `Yes`.	The item is deleted from the list.	The item is deleted from the list.
E	It is possible to delete all items in the list.	All items are deleted.	All items are deleted.
F	If there are many items in the database and not all items fit on the screen the user can scroll the list.	The list is scrollable.	The list is scrollable.

Figure 1.8: Test cases of the List view in Landscape mode.

1.5 Issues

As documented in the Test Cases no runtime issues have been found. This is not to say that there are no bugs in the software but the main usage of the application has been proven to work as expected.

The structure of the code on the other hand might not be optimal. There is redundancy with the landscape layout classes for both `TingleActivity.java` and `ListActivity.java` as well as the landscape files `activity_list.xml` and `activity_tingle.xml`. In the current setup these classes and layout files are more or less the same. To rid of this duplication a common dual-mode (landscape mode) layout file could be created.