

Assignment A5 - Widgets and Notifications

In this assignment, we'll add notifications and a widget to my to-do application.

Each todo item will be assigned a due time (10 seconds for this example), and will be marked "due" after that time. A notification will display how many items are due and allow them to be marked "Done" or snoozed for another 10 seconds. A widget will display the number of due items, and allow new items to be created.

You will also convert the to-do application to use the Android DataBinding Framework, simplifying data access and update.

Prerequisites

This assignment assumes you have completed the following modules

- [1.2 Android Versions](#)
- [1.3 Development Environment](#)
- [1.5 Gradle Basics](#)
- [1.7 Activities](#)
- [2.1 Intents](#)
- [6.1 Services](#)
- [8.1 Notifications](#)
- [8.2 Widgets and Broadcast Receivers](#)
- [9.1 Data Binding](#)
- [9.2 Alarm Manager](#)

NOTE: Be sure to name this project HW5 and its packages starting with lastname.firstname.hw5.

I recommend you approach this project in the following phases. I strongly recommend you commit after each step, or at least make a copy of the project directory in case you need to back out.

1. Download the Todo application starter code from <https://1drv.ms/u/s!AqvsXD36f5gmhK02AisjYAhd9p5cQg>
2. Run the starter code to get a feel for what it does.
3. Convert the Todo application to use the Android DataBinding Framework
4. Test to be sure the application works properly
5. Add a due time (long) and status (text - pending/due/done) to TodoItem and to the database
6. When a new item is saved, set the due time to 10 seconds from the current time. (NOTE: I just added this - if you already had set the due time using an input, you can keep it that way)
7. Update the user interface to display the due time (as a time/date string) and status. Both are read-only.
8. Add a "Mark Done" button to the action bar that will change the status of a displayed todo-item to Done
9. Add a "Snooze" button to the action bar that will add 10 seconds to the due time for the displayed to-do item
10. Use the AlarmManager to report when the next item is due. Only one alarm should be active at a time. Note that you will need to get the minimum due time from the database to do this. (You'll need to figure out how to do this.)
11. When the alarm is reported, send/update the notification for the app.
12. When the alarm is reported, send a broadcast intent to update the widget. Do not send data in the intent; the widget should fetch all data it needs from the content provider.
13. When an item is marked "done" or snoozed (its due-time changed), update the widget and notification. NOTE: You should NOT send this update from the application; these should be sent from the content provider because it is responsible for the data and eventually other apps could use the same content provider.

The Notification

- Must always be up to date with regard to the actual state of the data. In particular, when it's first displayed the data must be correct.
- Display the number of due items
- Display the names of the first five due items
- When tapped, display the to-do list
- Include a "Done" action - marks all due items done
- Include a "Snooze" action - sets the due time of all due items to 10 seconds from now

- If there are no due items, the notification should be canceled.

The Widget

- Must always be up to date with regard to the actual state of the data
- Display "N Item(s) Due" or "1 Item Due". See <http://blog.vogella.com/2011/11/22/plurals-in-android> for a good explanation of how you can make this very easy and potentially localized. Note that you can't use this to display "No Items Due" unfortunately because of the rules they use for English, so "0 Items Due" will suffice
- When tapped, display the to-do list
- Include a "Snooze" button (use an icon, not text) that snoozes all due items for 10 seconds
- Include a "+" button to create a new to-do item

Notes:

- You can change *any* of the provided starter code.
- This assignment is Activity-based, not fragment based, and doesn't do side-by-side layouts. You will need to set up the back stack properly when jumping to the edit page for a todo item.
- Do not add a "display" activity or convert to use fragments; this app only has a list and edit activity
- There is no "swipe to delete" in the starter code; do not implement it.
- Do not overthink this assignment, especially for converting the RecyclerView to use Data Binding. Do not attempt to use a generic adapter/recycler view for it; just do it directly like the first RecyclerView example.
- The notification should be ongoing as long as there are any items due
- Be sure that the content provider is the thing that triggers the widget update and notifications. It could end up used from multiple apps! If you send the notification or broadcast from the activity, and another app used the content provider, the notification and widget would not be correct! Note that this does not mean they need to be actually sent directly from the content provider. It's possible to minimize duplicated code through smart use of a broadcast intent.

Submitting Your Assignment

To submit your assignment (do this outside Android Studio):

1. Close the project in Android Studio (and/or quit Android Studio)
2. Find the directory for your project
3. Delete the build and app/build directories from the project folder
4. Zip the project directory. Be sure to get all files and subdirectories, including those that start with '.'. Be sure to name the zip file lastname.firstname.hw5
5. Go to the Blackboard course website
6. Click on the **Assignment Submission** link on the left navigation bar
7. Click on the **Assignment A5** link
8. Attach your submission and enter any notes
9. Be sure to **submit** your assignment, not just save it!

Content License

All non-source-code content in this course is licensed to you for your use in this course only. You may not distribute or reuse any content of this course for any purpose.

Source Code License

Source code in this course is licensed under the Apache License, Version 2.0 (the "License"); you may not use these files except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

