

CSC 372/472: Mobile Applications Development for Android

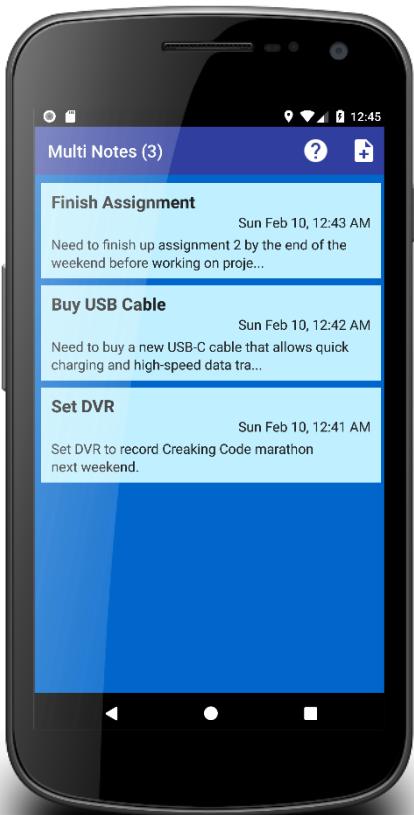
Assignment 2 – Note Pad (200 pts)

Uses: RecyclerView, Multi-Activity, JSON File, Option-Menus

App Requirements

- This app allows the creation and maintenance of multiple notes. Any number of notes are allowed (including no notes at all). Notes are made up of a **title**, a **note text**, and a **last-update time**.
- There is *no need* to use a different layout for landscape orientation in this application. **One layout must work fine in either orientation.**
- Notes should be saved to (and loaded from) the internal file system in **JSON** format. If no file is found upon loading, the application should **start with no existing notes and no errors**. (A new JSON file would then be created when new notes are saved).
- JSON file loading** must happen in the **onCreate** method. **Saving** should happen in the **onPause** method.
- A simple java **Note class** (with title, note text, and last save date) should be created to represent each individual note in the application.
- The application is made up of **3 activities**. These are described below:

1) Main Activity:



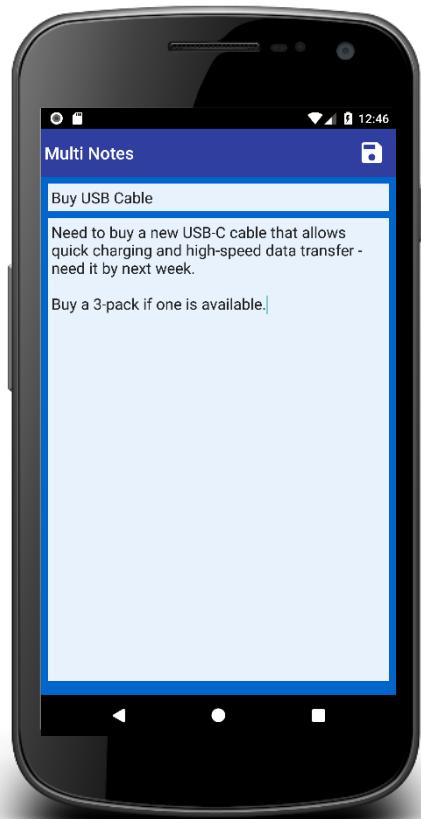
- Notes should be displayed in a list, **in time order** (latest-update-first, oldest-update-last). The note list must be implemented using the **RecyclerView**.
- The **Main Activity** will allow the user to **create a new note** via an **Add** options-menu item **+**. Pressing this button will open the **Edit Activity** (described next) with empty Title and Note Text areas.
- The main activity will allow the user to **edit an existing note** by **tapping** on an existing note in the displayed list of notes. Doing so will open the **Edit Activity**, displaying the note's Title and Note Text – both available for editing.
- The main activity will also have an **Info** options-menu item **i** that will open the **About Activity** (described later) when pressed. The **About Activity** indicates the **application's name**, the **date & author**, and the **version number**.
- Notes can be **deleted** from the **Main Activity** by **long-pressing** on a note entry. Upon doing so, a confirmation *dialog* will be opened where the user can confirm (or cancel) the delete operation.
- Note list-entries contain the **note title** (bold), the **last-save-time**, and the **first 80 characters of the note text** (see image). Notes titles or note text with more than 80 characters should display **...** after the first 80 characters are displayed.
- The current **number of notes** is displayed in the **title-bar**.
- One layout must work fine in either orientation.



Main Activity (cont.)

- The activity background should be set to a color other than white (preferable a darker color). The list entries should be a lighter version of that same color.
- Icons for the Add and Info menu items should be selected from Google's Material Design icon set (<https://material.io/icons/>)

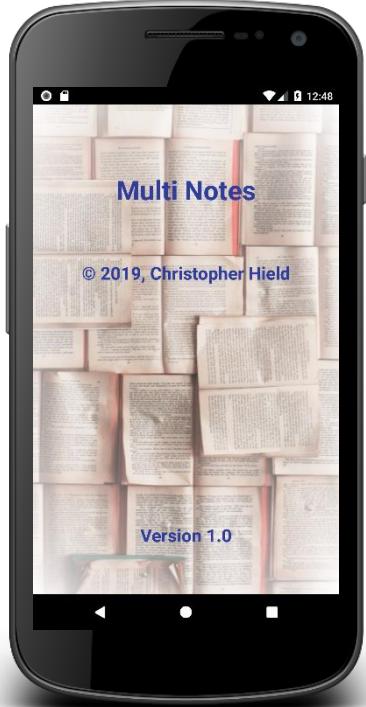
2) Edit Activity



- The **Edit Activity** contains editable fields for the note *title* and note *text*. The last-save time is NOT displayed here and is never user-editable – it should be automatically generated and saved when the note is saved.
- The note title is a single-line text field (EditText) where the user can enter or edit a title for the current note (no size limit).
- The note text is a multi-line text area (EditText) with no size limit. This should have scrolling capability for when notes exceed the size of the activity.
- The **Edit Activity** allows the note title & text to be saved by either:
 - a) Pressing the Save options-menu item . This will return the new note to the MainActivity, and then exit the Edit Activity. MainActivity will then add the note to the MainActivity's list of notes. Note that if no changes have been made to the current note, the Edit Activity simply exits.
 - b) Pressing the Back arrow (on the nav bar) to exit the activity. This will first display a confirmation dialog where the user can opt to save the note (if changes have been made) before exiting the activity. If saved, the new note is returned to the MainActivity, and then exit the Edit Activity. MainActivity will then add the note to the MainActivity's list of notes. Note that if no changes have been made to the current note, the Edit Activity simply exits.
- Note this "Save" in steps "a" and "b" does not imply saving to file – it implies updating an existing note object in the Main Activity's note list or adding a new note to the Main Activity's note list.
- A note without a title is not allowed to be saved (even if there is note.text). If such an attempt is made simply exit the activity without saving and show a Toast message indicating that the un-titled activity was not saved.
- The activity background should be set to a color other than white (the same color used in the Main Activity). The title and text fields should be a lighter version of that same color (the same color used in the Main Activity).
- Icons for the Save menu item should be selected from Google's Material Design icon set (<https://material.io/icons/>)



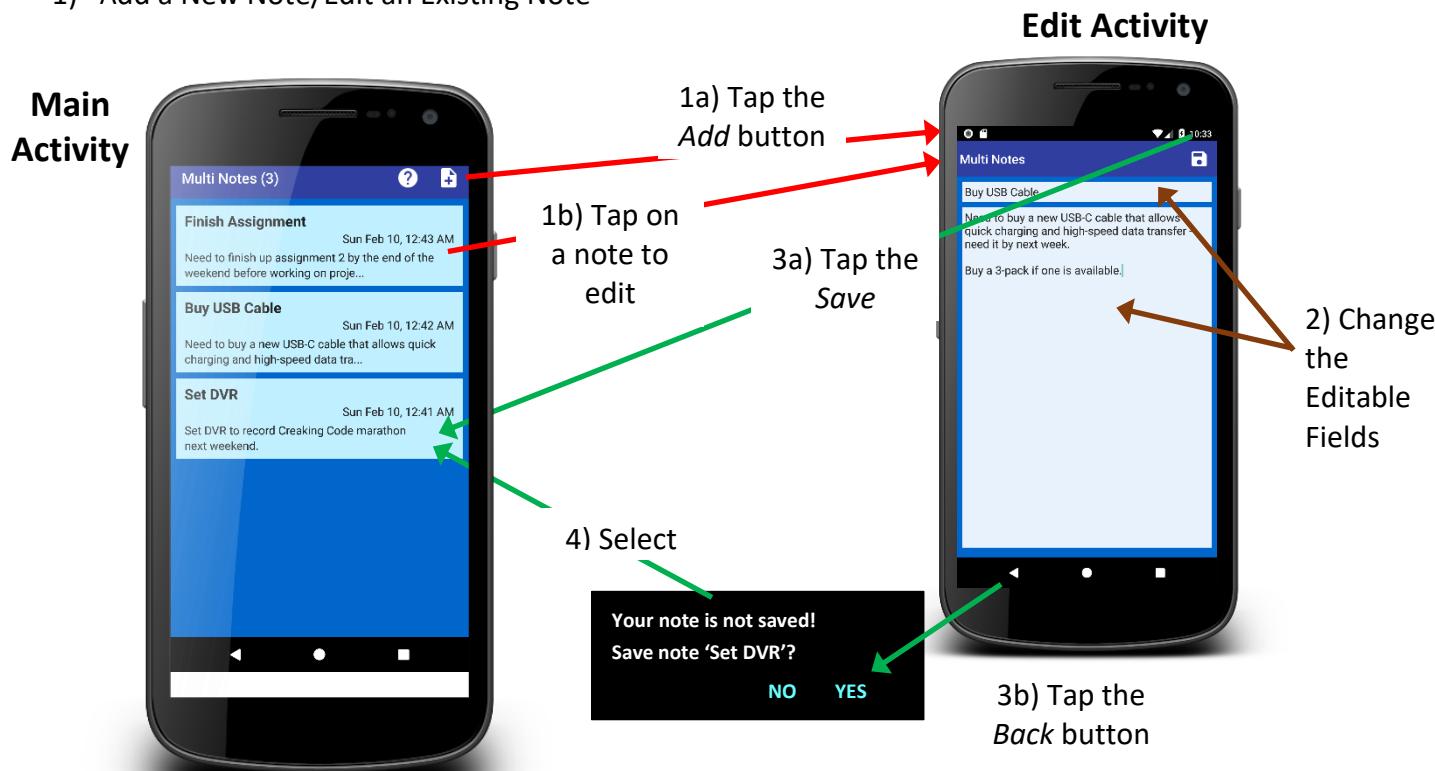
3) About Activity



- The **About Activity** should contain a full-screen image background (use any non-copywritten image you desire).
- Over the background image, key information on the application should be clearly displayed (clearly readable). This information should include the *application title*, a *copyright date and your name*, and the *version number* (1.0).
- There is no functionality present on this activity. The only action a user can take is to press the Back arrow to exit the activity.

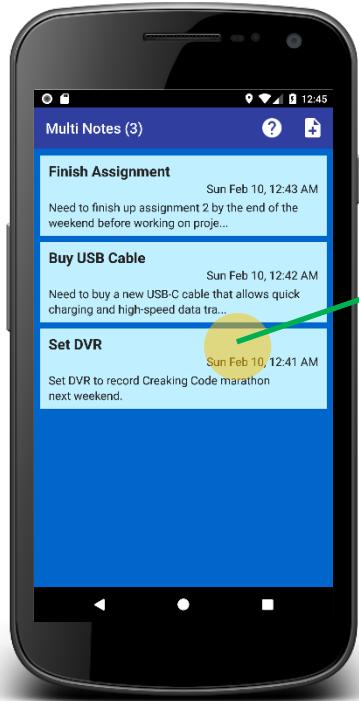
Application Flow Diagrams (*showing how the various behaviors should work*)

1) Add a New Note/Edit an Existing Note



2) Delete an Existing Note

Main Activity



1) Long-
Press a
Note

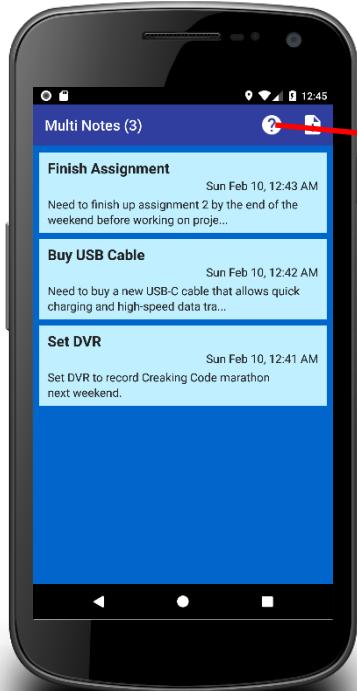
Delete Note 'Set DVR'?

NO YES

2) Selecting YES will
remove the note from
the Main Activity's note
list (and from the
display).

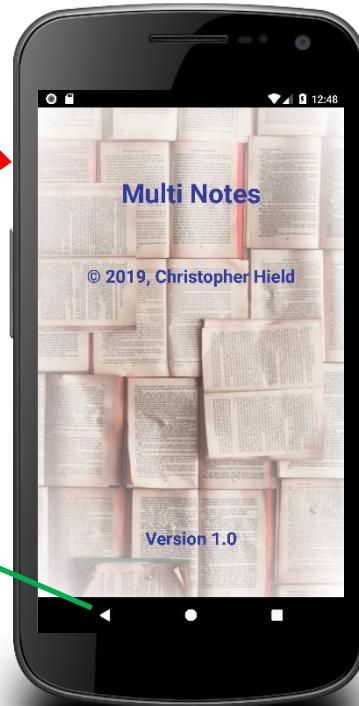
3) View the application "about" details

Main Activity



1) Tap the
Info button

About Activity



2) Tap the
Back
button to

Assignment Assistance

If you are stuck on an assignment problem that you have exhaustively researched and/or debugged yourself, you can email me a ZIP file of your entire project so that I can examine the problem. All emailed assistance requests must include a detailed description of the problem, and the details of what steps you have already taken in trying to determine the source of the problem.

Note: To make your submission zip file smaller, before zipping your project file, you can remove the ".gradle" folder (found in your project's root directory), and remove the "build" folder (found in the "app" folder in your project's root directory).

Submissions & Grading

- Submissions must consist of your zipped project folder (please execute Build => Clean Project before generating the zip file). Submissions not following these requirements will be penalized.
 - *Note: To make your submission zip file smaller, before zipping your project file, you can remove the ".gradle" folder (found in your project's root directory), and remove the "build" folder (found in the "app" folder in your project's root directory).*
- Submissions should reflect the concepts and practices we cover in class, and the requirements specified in this document.
- Late submissions will be penalized by 10% per week late. (i.e., from one second late to 1 week late: 10% penalty, from one week late to 2 weeks late: 20% penalty, etc.).
- The following are the key points that will be examined in Project when graded:
 - General (25%)
 - Loads and Saves Notes to a JSON file – no errors displayed if file is not present.
 - Load is initiated in the *onCreate* method, save is performed in the *onPause* method.
 - A separate Note class used to represent note data
 - App is functional and visible in portrait and landscape orientations
 - MainActivity (37%)
 - Options Menu is present in App-Bar and is displayed as Icons
 - Clicking Add button opens the Edit Activity
 - Clicking Info in the list opens the About Activity, displaying the note's title and content
 - Long-click on a note in the list displays Delete Confirmation dialog.
 - Approving dialog results in note deleted from list
 - Cancelling dialog leaves the list unchanged
 - Activity background color is a color other than white
 - List uses *RecyclerView*

- Note list is displayed in order of most recent first
- List entry contains a bold title, the last-save-time, and note text is limited to 80 chars
 - Adds “...” for titles and notes containing more than 80 chars
- EditActivity (28%)
 - Contains editable fields for title (single line) and note text (multi-line)
 - Options Menu is present in App-Bar and is displayed as Icons
 - Clicking Save button saves the note, closes the *EditActivity*, and updates the *MainActivity* list
 - Clicking Back button displays Save Confirmation dialog.
 - Approving dialog results in note saved, closes the *EditActivity*, and updates the *MainActivity* list
 - Cancelling dialog closes the *EditActivity* and leaves the list unchanged
 - Notes with no title are not saved (Toast message informs the user)
 - Activity background color is a color other than white
- About Activity (10%)
 - Contains full-screen background image
 - Shows application title, copyright data and student name, and Version 1.0

NOTE

This assignment is worth 200 points (your first assignment was worth 100 points).

This means (for example) that if you get 89% on this assignment, your recorded score will be:

$$(89\% * 200 \text{ points} = 178 \text{ points})$$

Note that this also means that the 10% late submission penalty will be $10\% * 200 \text{ points} = 20 \text{ points}$.

If you do not understand anything in this handout, please ask.

Otherwise the assumption is that you understand the content.

Unsure? Ask!