420-616-AB Oracle Technologies
Fall 2019

September 24th
Worth 7%

Assignment II

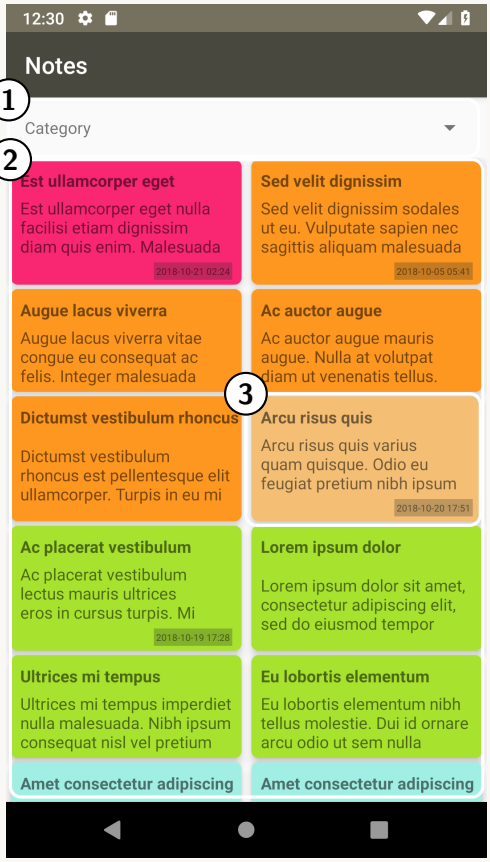
(Due on Check on Léa)

Design and implement an Android `Activity` and `Fragment` for a user to browse a list of notes. These notes are stored in an `sqlite` database locally on the Android device.

Add the new activity and fragment to your app from the previous assignment, but integrating the two activities will be done in the next assignment.

The user interface is will look like this. Each part of the UI is outlined in the following sections.

1. A spinner (drop-down) for choosing the list sort criteria.
2. A list of notes arranged in a grid with 2 columns.
3. Each note displays the title, body and reminder (if set) of the note. The category of the note as the background colour.



1 UI

The UI is demonstrated in the video alongside these instructions. Please watch it before continuing on. Below are the requirements and additional resources.

1.1 Sorting

Requirements. A `Spinner` lists the available sorts, described below. When the user changes the selected item in the spinner, or when a modification to the data takes place, the sort is reapplied. The list is initially sorted according the default sort.

Notes. The following sorts are supported:

Sort method	Description
Title	Sorts notes by title alphabetically in ascending order (A ◀ Z).
Creation Date	Sorts notes by the date they were created in descending order (present ▶ past).
Last Modified	Sorts the notes by the modification date in descending order (present ▶ past).
Reminder	Sorts notes by reminder in ascending order (present ◀ future). Any note without a reminder will appear at the bottom of the list.
Category	Groups notes by category, in any order.

1.2 Note Adapter and Holder

Requirements. Each `Note` is displayed as shown in the app preview, but the rounded corners and shading are optional. Implementation uses the "view-holder" pattern discussed in class, meaning you solution extends both the `RecyclerView.Adapter` and `RecyclerView.Viewholder` as `NoteAdapter` and `NoteViewHolder` respectively.

Notes.

- Use the `GridLayoutManager` with 2 columns.

1.3 Floating Menu

Requirements. When a note is touched, display a floating menu, called an `ActionMode`, with the following menu items: Reminder, Trash and Close.

- Reminder will start the date and time selection process from Assignment 1. Use the same code and classes.
- Trash will delete the note.
- Close will close the menu.

For Reminder and Trash, both the note in memory and in the database are affected.

The position of the floating menu are based on the position the user touched on the device screen.

Notes.

- Read through the following tutorial on Android's `ActionMode` class: `https://medium.com/over-engineering/using-androids-actionmode-e903181f2ee3`[1]. The menu design and event handling are similar to what you did in Assignment 1.
- Use the view's `OnTouchListener` to handle touch interaction with the list item (`https://developer.android.com/reference/android/view/View.OnTouchListener`).
- Don't forget to ensure that the adapter knows about the data change and that sorting is still correct.

[1] While the code is written in Kotlin, it shouldn't be too hard to understand how to write equivalent code in Java.

2 SQLite Database

Requirements. The set of notes displayed in the list are stored in an `sqlite` database called `notes.db`. The database consists of a single table called `note` that reflects the structure of the class `Note`. All modifications to `Note` objects in memory must also be stored in the database.

Use the classes from the lectures (package `sqlite`) to manage and interact with the `sqlite` database.

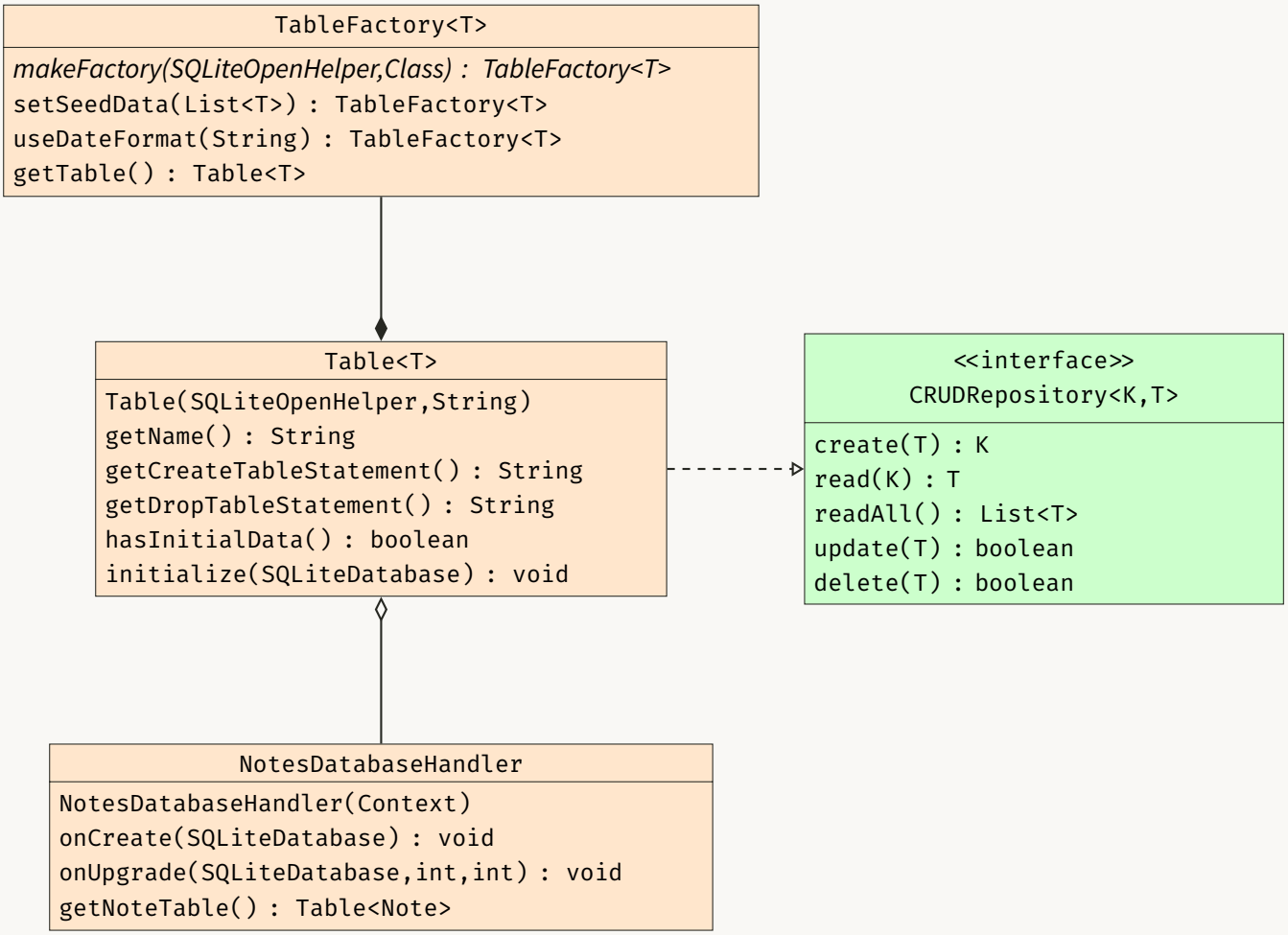
Notes.

- Recall that each table in `sqlite` has a primary key column called `_id` (auto incrementing) and that the provide classes already set this up for us.
- Recall that `sqlite` has no date datatype. Use this date format string for all dates stored in the database:

```
yyyy-MM-dd'T'HH:mm:ss.SSSZ
```

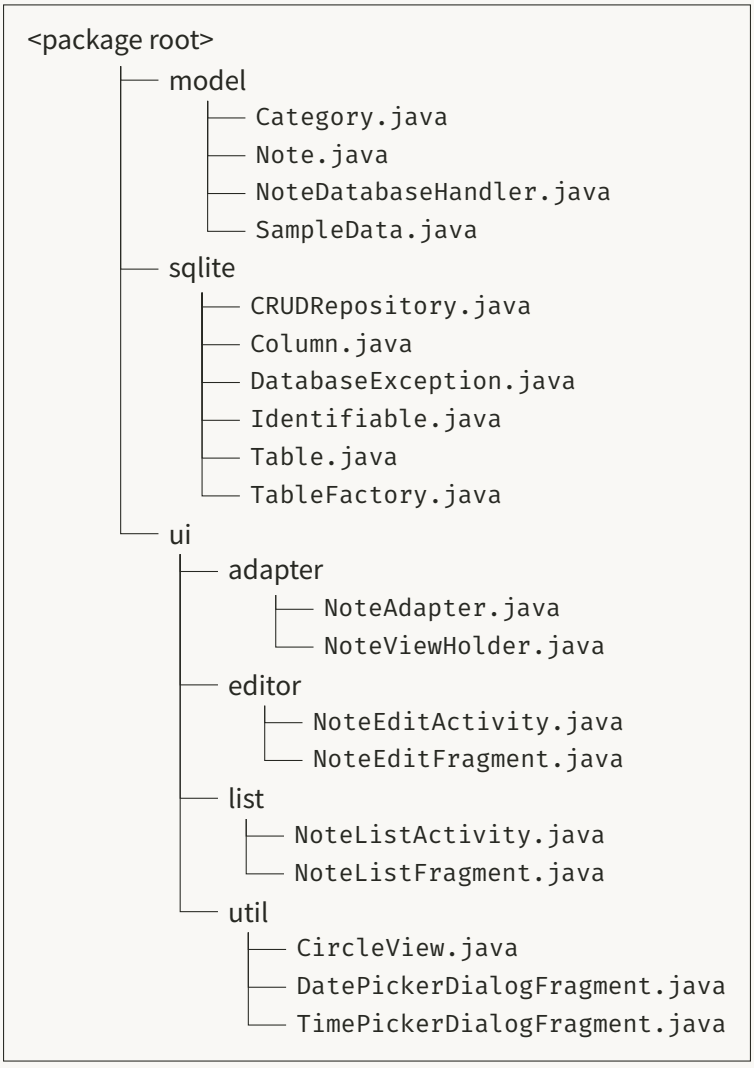
- See the tutorial video in the course notes on how to subclass the `Table` class using the factory class `TableFactory`.

Here is the UML class diagram of the important classes and interfaces in the `sqlite` package along with the `NotesDatabaseHandler`. See their documentation for specific information on each method.



3 Project Structure

Use the following directory structure for your project's source code, separating the app into model, ui, activities and utilities:



Use refactoring to rename the original class names made when creating the project.

4 Requirements

- Your program should be clear and well commented. It must follow the "420-616 Style Guidelines" (on Léa).

- Create an Android project with minimum SDK 23 - Android 6.0 (Marshmallow) or later.
- Your list activity uses a fragment.
- The UI meets all the requirements above.
- All notes are stored and modified in the `sqlite` database using the provided classes.
- The app uses the provided sample data: `sample-data.csv`.
- Submit using Git by following the instructions (on Léa).