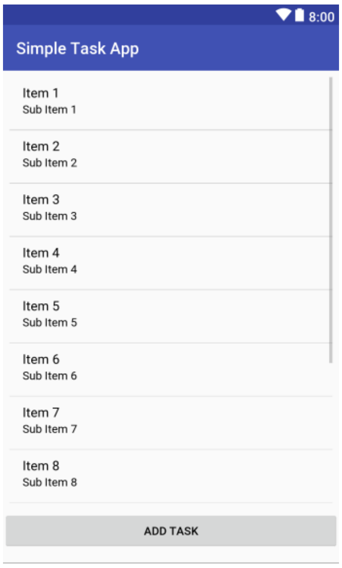
University of Applied Sciences Bremen
IS Medieninformatik B.Sc.
Summer Semester 24
Prof. Dr. Thorsten Teschke

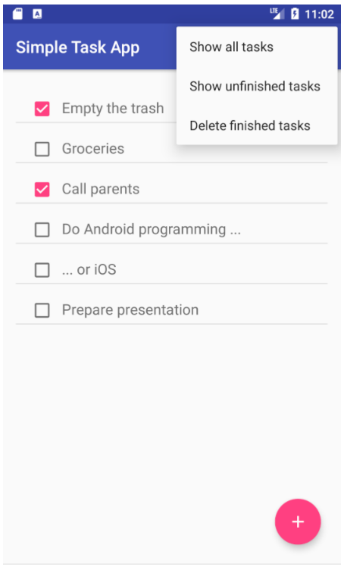
Mobile Computing

2. Lab Assignment: Activity for Listing Tasks

25.04.2024

Presentation of results on: 16.05. (A) / 23.05. (B)





Objectives

- Working with RecyclerViews and adapters
- Saving and restoring instance state

Your Task

Create a new activity (e.g. “TaskListActivity”) in your app which shows a list of tasks and a button for adding new tasks.

Steps:

- Define layout
- Retrieve tasks from repository (use the repository classes provided in folder “Lab Assignments (Standard Track)”)
- Implement an appropriate adapter for a list of tasks
- For now, it is enough for the Add button to present a *Toast* or *Snackbar* (“You pressed the Add button” or the like). In future versions of the app, the button should present the selected task in the `TaskDetailActivity` you developed in the first assignment.

The layout of your view might be as in the example illustrations (left: using classic Button for adding tasks; right: using Floating Action Button). A menu is not required at this time. The check boxes shown on the right-hand side represent a bonus task (see next page).

See page 2 for hints and bonus tasks.

Hints

1.) For the list view, you should use a `RecyclerView` class (you may alternatively use the `ListView` included in the standard Android API; the implementation of *list row reuse* and the *view holder pattern* is expected here as well).

2.) Your new activity now needs to be the launcher activity, i.e. you have to update the intent filter in your Manifest file.

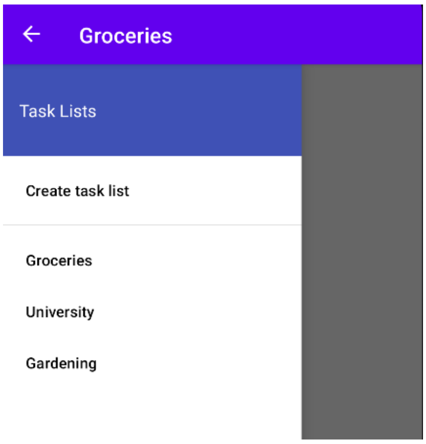
3.) You may choose whether to use a classic Add button or a Floating Action Button (FAB). Your activity will already be provided with a Floating Action Button, if you create your activity in Android Studio via *context menu* → *New* → *Activity* → *Basic Views Activity*. I do NOT recommend to use this project template for beginners, since Android Studio generates rather complex Java and layout code. I suggest you rather stick to the *Empty Activity* template and – if you want to add a FAB later - do some research on how to integrate it into your layout and code as soon as you have become familiar with the core concepts of Android development.

Bonus Tasks

If you finished the above task, you may gather bonus points by solving the following additional tasks.

1. List Item Layout: Present list items with an additional check box (cf. right-hand side figure on page 1).
   - Define your own item layout which includes a check box for each list item / task.
   - Make sure that the rows' checkboxes are initialized appropriately in your list adapter.
   - A click on the list item's check box should update the task object's *done* property. Hint: install an `OnClickListener` in each list item in the `onBindViewHolder()` method of the recycler view adapter (or the `getView()` method of a list view adapter).

2. Multiple task lists:
   - Add a `TaskList` class that has an ID, and a name. `Task` objects need to be associated with one `TaskList` instance.
   - Modify the repository with regard to creating task lists, creating tasks in task lists (or adding tasks to task lists), retrieving task lists and the tasks of a selected task list.
   - Add a *navigation drawer* menu which shows a list of available task lists and an entry for adding a new task list. Tapping on one task list should show the tasks of the selected list and the name of the selected list in your app's *action bar* (you may have to add an action bar, if not already present). Tapping on the entry for adding a task list should open a dialog where a user can enter the desired task list's name. The navigation drawer should be closed afterwards.


←
Groceries
Task Lists
Create task list
Groceries
University
Gardening