

Lab – User Interface Classes

Create a complex user interface with Android's user interface classes

Objectives:

Familiarize yourself with Android's User Interface (UI) classes. Create a simple application that uses a variety of UI elements including: Buttons, TextViews and Checkboxes. You will also reinforce the knowledge you've gained in previous lessons by implementing a larger portion of the application from scratch.

Overview:

In this Lab, you will create a ToDo Manager application. The application creates and manages a list of ToDo Items (i.e., things that you need “to do.”) You will design this application's user interface, including its layout and resource files. You will also implement a bit more of the application's features than you did in previous Labs. Do NOT modify any resource IDs contained in the skeleton layout files as this may break our test cases.

Exercise A: The Basic ToDo Manager

The main Activity for this application is called Lab5_UI Labs. When the Activity runs, but there are no previously saved ToDo Items, its initial UI will look something like this:



Figure 1: Initial View

This user interface contains a single RecyclerView that displays all existing ToDo Items. As shown above, the last row of the RecyclerView always displays a special View, with the words, “Add New ToDo Item.” The first position within the RecyclerView is known as the “header”.

When the user clicks on the RecyclerView’s header the application will start a new Activity called AddToDoActivity that allows the user to create and save a new ToDo Item.

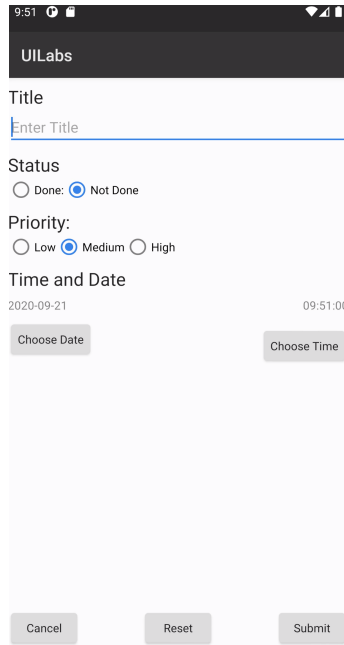


Figure 2: Adding a New ToDo Item

ToDo items have at least the following fields. Default values appear in bold:

- Title: A user-provided String. The default Title is the empty String ("").
- Status: one of {Done, **Not Done**}
- Priority: one of {Low, **Med**, High}
- Time & Date: A deadline for completing this ToDo Item. The default deadline is **7 days from the current date and time**.

This Activity's user interface includes the following buttons:

- Cancel – finish the Activity without creating a new ToDo Item.
- Reset – reset the fields of the ToDo Item to their default values and update the display to reflect this.

- Submit – create a new ToDo Item containing the user-entered / default data fields and return it to ToDoManagerActivity. When the application returns to ToDoManagerActivity, the new ToDo Item should appear in the RecyclerView.

For example, if the user creates and submits a new ToDo Item to an empty ToDo list, then once the application returns to the ToDoManagerActivity, its RecyclerView will contain the new ToDo Item, as shown below.

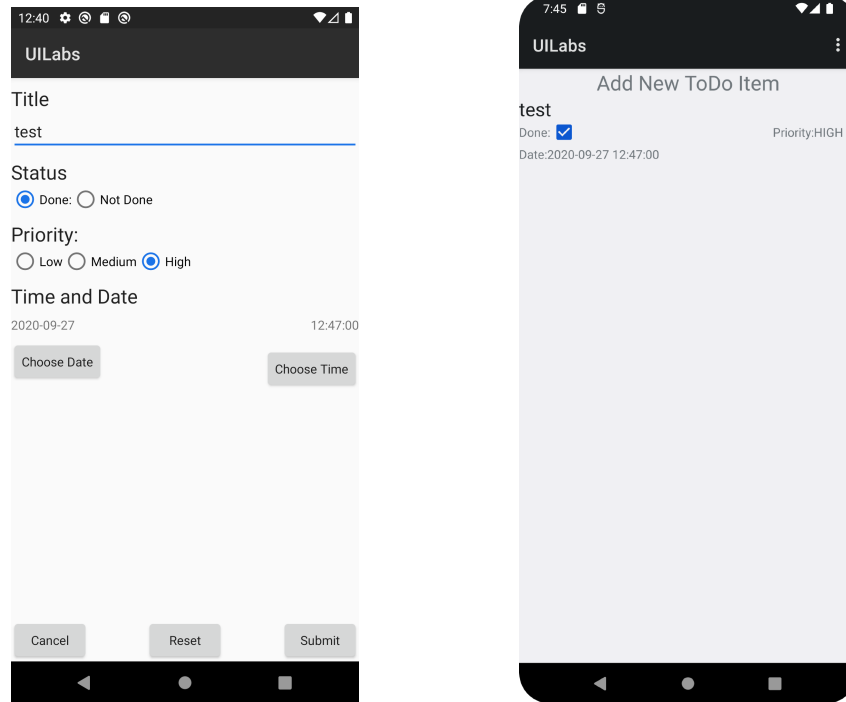


Figure 3 (left): The user creates a new ToDo Item. Figure 3 (right) After submitting the new ToDo Item, the application returns to the main Activity, displaying the new ToDo Item.

Back in the Main Activity, the user will be able to toggle the Done checkbox to indicate that the ToDo Item's status has changed, say from “Not Done” to “Done”.

See the UILabs.mp4 screencast to see the app in action.

Implementation Notes:

1. Run git pull upstream main branch. Immediately push this to your project on gitlab to make sure you have it there as well.
2. Implement the project according to the specifications described above. To implement the Lab, look for comments in the skeleton files containing the String `"//TODO."` As with previous Labs, these comments contain hints as to what you need to do to complete the project. However, be

aware that from here on out these comments will become increasingly less comprehensive, requiring you to make more decisions about how to implement the entire project and satisfy its requirements.

Testing

We will be testing this Lab using the following test cases:

Test Case 1

1. Click More Options on top right of the screen
2. Click 'Delete All' to delete all existing Todos
3. Add a new todo item with the name 't4', priority 'LOW', status 'Done'
4. Click 'Submit'
5. The todo should show up on the application's main view with the correct properties (name, priority, status).

Test Case 2

1. Click More Options on top right of the screen
2. Click 'Delete All' to delete all existing Todos
3. Add a new todo item with the name 't1', priority 'LOW', status 'Done', date '01/28/2014', and time '9:19'.
4. Click 'Submit'
5. The todo should show up on the application's main view with the correct properties (name, priority, status, date, time).

Test Case 3

1. Click More Options on top right of the screen
2. Click 'Delete All' to delete all existing Todos
3. Click 'Add new Todo Item' and fill in the name 't1'
4. Click Cancel
5. At this point, the application should go back to its main view and no todo should be added
6. Add a new todo item with some properties, hit submit and make sure the todo shows up on the application's main view with the correct properties (name, priority, status, date, time).

Test Case 4

1. Click More Options on top right of the screen

2. Click 'Delete All' to delete all existing Todos
3. Click 'Add new Todo Item' with name 't2' and fill in all properties
4. Click 'Reset'
5. Click 'Submit'
6. Make sure the newly added Todo item has all its field reset (i.e., the name is not 't2', the status box is unchecked, and the priority matches the regex'
7. On ToDo list screen verify that the 'Status' checkbox checks and unchecks on clicking the checkbox

Warnings:

1. We will be testing your app on a Pixel 5 AVD emulator with API level 31. To avoid configuration problems, you should test you app against a similarly configured AVD. Also, when testing, make sure that your device is in Portrait mode when the test cases start running.
2. On startup, these test cases delete all existing ToDoItems.
3. All tests will be weighted equally when being graded.

Submission

To submit your project just save and commit all your changes locally and push to your gitlab project. In the commit message for your final solution please leave the following message: "completed Lab5 implementation".