CMSC436: Fall 2013 – Week 5 Lab

Objectives:

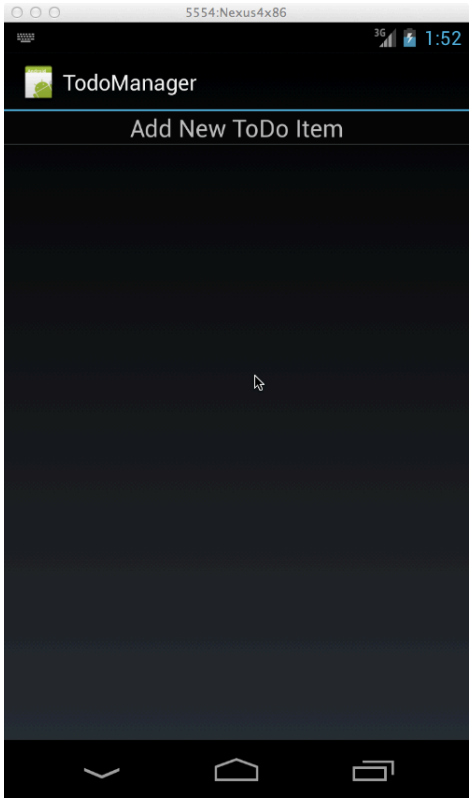
Familiarize yourself with Android's User Interface (UI) classes. Create a simple application using a variety of UI elements including: Buttons, TextViews and Checkboxes.

Overview:

In this lab, you will create a ToDo manager application. The application creates and manages Todo List Items. You will design this application's UI, including its layout and resource files. You will also implement the application's features by yourself. As a result, we will not provide a skeleton for this lab.

Exercise A: A basic ToDo manager application

The first time the application runs it will have no ToDo Items and therefore its initial UI will look something like this:

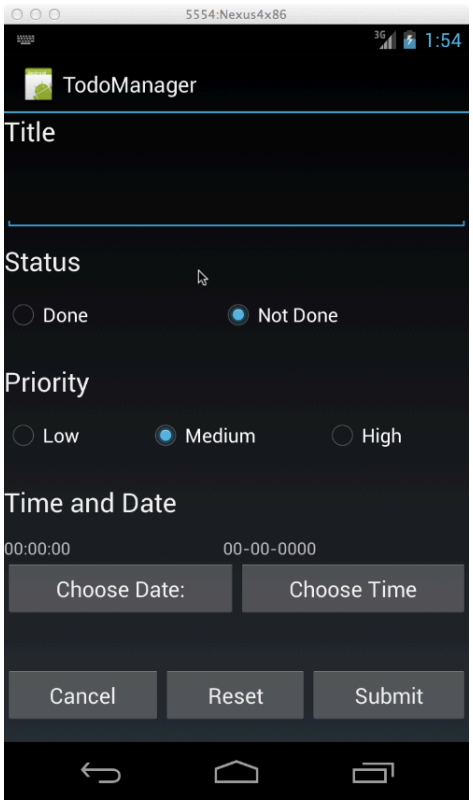


This UI contains a ListView for displaying existing Todo Items. As ToDo Items are created, they will be added to this ListView.

Initially, the application can either start with empty list or you can create some dummy ToDo Items. As you work through the Lab, you'll make ToDo Items persist across sessions.

The ListView, shown above, always displays a special item, labeled “Add New ToDo Item” in its last position. This position is the called the “footer.” When the user clicks on the ListView footer, a new Activity will be opened that allows the user to create a new ToDo Item.

You are free to define your own user interface for this new Activity, but it must include the fields shown in the figure below.



Specifically, ToDo items have the following fields:

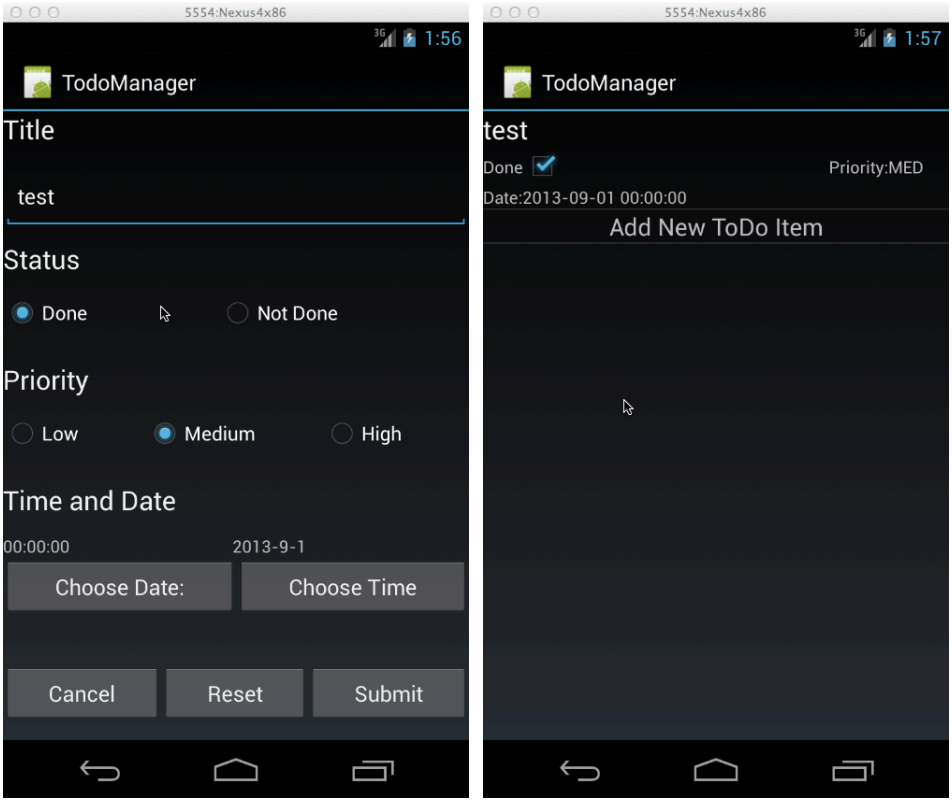
- Title: A user-provided String
- Status: {Done, Not Done}
- Priority: {Low,Med,High}
- Time & Date: A deadline for completing the underlying ToDo Item.

The “Add New ToDo Item” Activity's UI should also includes several buttons:

- Cancel – finish the Activity without creating a new ToDo Item.
- Reset – Reset the fields of the Todo Item to their default values.

- Submit – Create a new Todo Item with the user selected data fields & return to main Activity. When the application returns to the main Activity, the new ToDo Item should appear in the main Activity's Listview.

For example, if the user adds a new ToDo Item to an empty ToDo list, as shown below, then the main Activity's ListView should update to display the new ToDo Item.



Back in the Main Activity, the user should be able to toggle the Done checkbox to indicate that the ToDo Item is Done or Not Done.

To make Todo Items persist across sessions, you can write Todo items to a file and read existing Todo items from the file at application startup. Since you have not learned about file I/O, we provide you with a simple file I/O implementation, which can be used to read/write data from/to a file. You can download a file with the code from the class Labs & Lectures web page. Depending on how you implement your application, you may need to modify this code. More information about FileInputStream and FileOutputStream can be found from here: http://developer.android.com/reference/android/content/Context.html

Exercise B: An improved ToDo manager application

This part is optional, but if you've gotten through Part A, why not challenge yourself.

In this part, modify your application so that ToDo Items that are Not Done are displayed in the Main Activity with a different colored background than those that are Done. In addition, when the user toggles the Done checkbox, the background color should change as appropriate.

Exercise C: A ToDo manager application you might actually use

This part is again optional, but if you've gotten through Part B, you can't stop now.

In this part, modify your application so that if the user selects a ToDo Item in the Main Activity's ListView, a dialog pops up, allowing the user to delete the selected ToDo Item.

Deliverables:

Submit your project via the submit server. Your submission should include:
a. A zip file containing the source code eclipse project