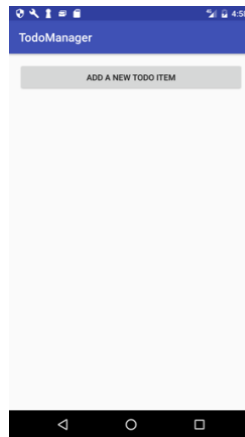


HOMework 2

In this homework, 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 need to implement the application's features by yourself.

PART 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 RecyclerView for displaying existing ToDo Items. As ToDo Items are created, they will be added to this RecyclerView. Initially, the application can either start with empty list or you can create some dummy ToDo Items. As you work through the Homework, you'll make ToDo Items persist across sessions.

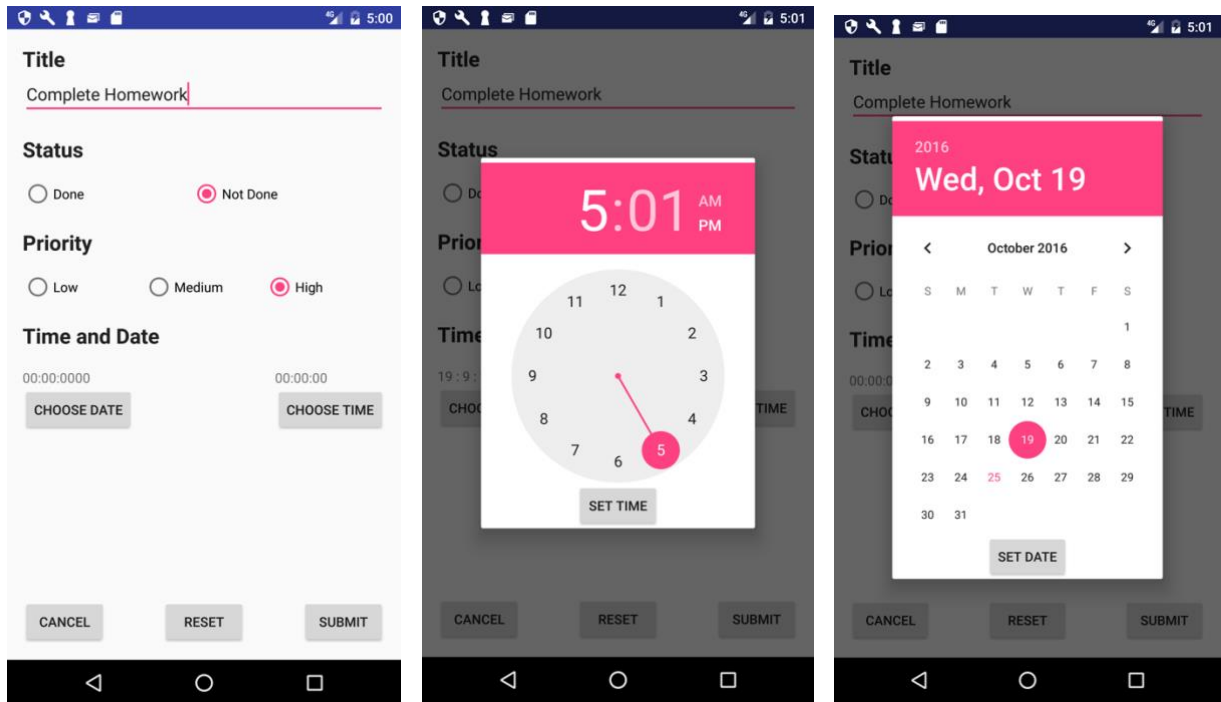
The RecyclerView, shown above, always displays a special item, labeled “**Add New ToDo Item**” in its last position. This position is called the “**footer**.” When the user clicks on the **RecyclerView footer**, a new Activity will be opened that allows the user to create a **new ToDo Item**. You are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

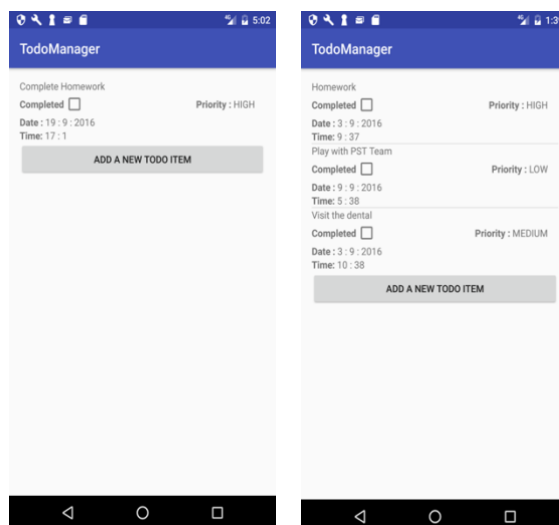
1. Title: A user-provided String
2. Status: {Done, Not Done}
3. Priority: {Low,Med,High}
4. Time & Date: A deadline for completing the underlying ToDo Item. The “Add New

ToDo Item” Activity's UI should also includes several buttons:

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



• **Submit Button** – Creates 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 **RecyclerView**. For example, if the user adds a new ToDo Item to an empty ToDo list, as shown below, then the main Activity’s RecyclerView should update and display the new ToDo Item.

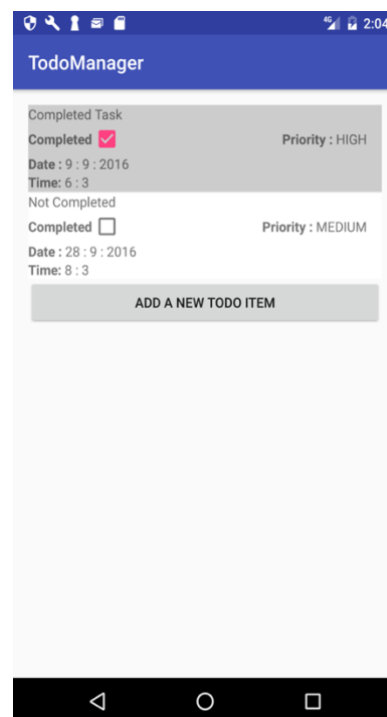
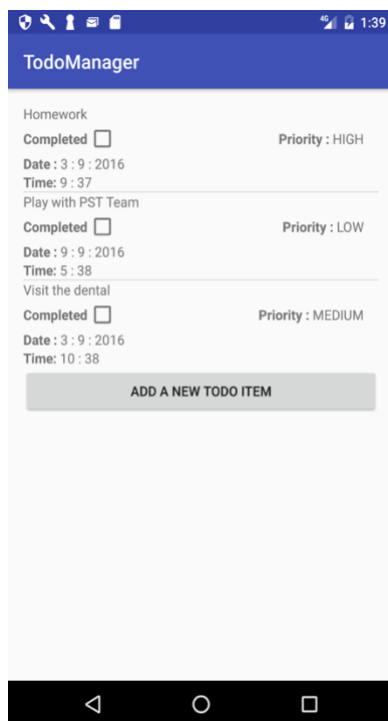


Back in the Main Activity, the user should be able to toggle the Done checkbox to indicate that the ToDo Item is **Done** or **Not Done**.

Optional→To make **ToDo** Items persist across sessions you can use `SharedPreferences`. You can save all your to-do list items into a shared preference and read them back as soon as the application starts. Since you have not learned about `SharedPreferences`, **you are required to do this** for the moment. However, **if you are curious** on how to implement this you can find more information about `SharedPreferences` here: <https://www.journaldev.com/9412/android-shared-preferences-example-tutorial>.

PART B: AN IMPROVED TODO MANAGER APPLICATION

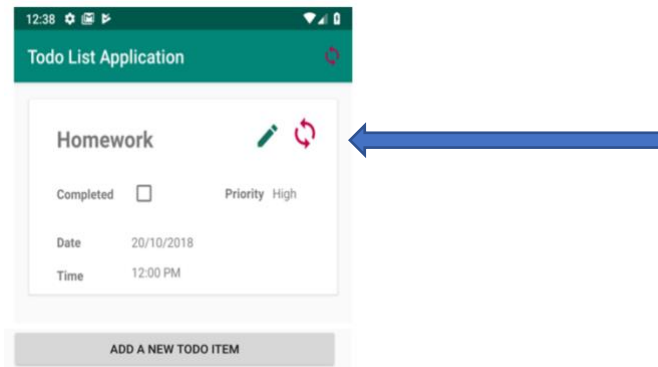
In this part, modify your application so that **ToDo Items** that are **Not Done** are displayed in the Main Activity with a different **colored background** than those that are Done. In addition, when the user toggles **the Done checkbox**, the background color should change as appropriate.




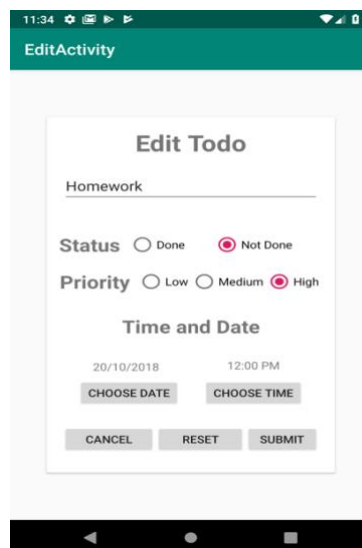
Part C: A ToDo manager application you might actually use


In this part, you do the following modifications to your application.

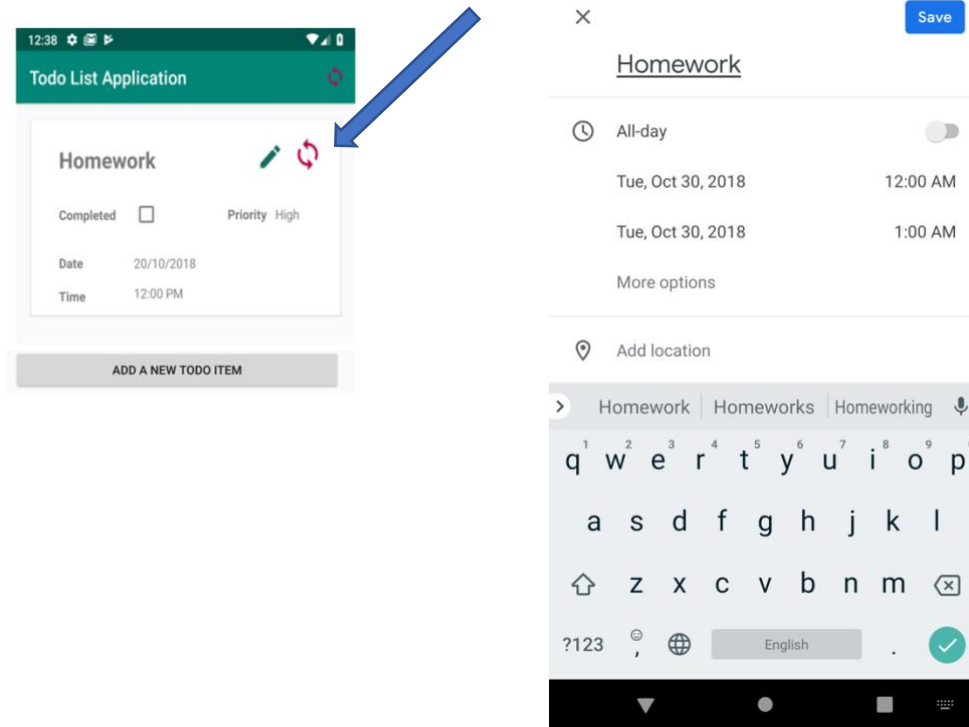
1. Redesign your ListItem layout to use **Card View** for your list items instead of the simple layout that you implemented above. Also add the following two icons. One for editing and another one for synching.



2. When the user presses on the edit icon  you should Open an Activity that allows the user to edit their todo item. You can reuse the **Add Activity** and pass the todo as a parcelable. Then differentiate when it is Add vs Edit. Otherwise you will need to create a third activity for editing.



3. Finally, when the user presses on the sync Icon  the application should make a calendar entry of the todo list item on Google calendar or any other calendar configured on the user's device.



Deliverables:

1. Submit your **project code** via Github. Make sure you remove the APK file by cleaning your project before submission or by adding the apk to the .ignore file. Otherwise the file size will be huge.
2. Submit the **Testing Sheet [Word file]** with all the screenshots. Do not convert it to PDF. If you do not submit this you will lose 15%.
3. You will be required to demonstrate your work in the Lab so be prepared to answer all questions related to your work.