ITIS/ITCS 4180/5180 Mobile Application Development
Homework 2
Date Posted: 01/22/2015 at 11:00
Due Date: 01/29/2014 at 23:55

Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each student in the group is required to submit the assignment on moodle.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your project as follows:
   a. From eclipse, choose *"Export…"* from the File menu.
   b. From the Export window, choose *General* then *File System*. Click *Next*.
   c. Make sure that your project for this assignment is selected. Make sure that all of its subfolders are also selected.
   d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
   e. When exporting make sure you select *Create directory structure for files*.
   f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
   a. All the group members should submit the same zip file.
   b. The file name is very important and should follow the following format: **Group#_HW02.zip**
   c. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

Homework 2 (100 Points)

In this assignment you will get familiar with Android intents (Explicit/Implicit), data passing between activities, starting activity for result, alert dialogs and dynamic UI management. This is a To-Do List App, where user gets to enter a task title, the time and date, along with other task details. User should be able edit, display and delete tasks. This is a multi screen application, the Main Activity is displayed in Figure 2.

Important App Requirements:

1. The required Android Virtual Device (AVD) should have **minimum SDK version set to 14 and target SDK at least 19**. The app should display correctly on Nexus 5 sized device. Your assignment will not be graded if it does not meet these requirements, and you will not be granted any points on your submission.
2. All strings should be read from your strings.xml.The string values used for the text labels, and button labels should be read from the strings.xml file and should not be hardwired in the layout file.
3. **You should not use static variables to share data between activities, and should use intent extras to share data between activities.**

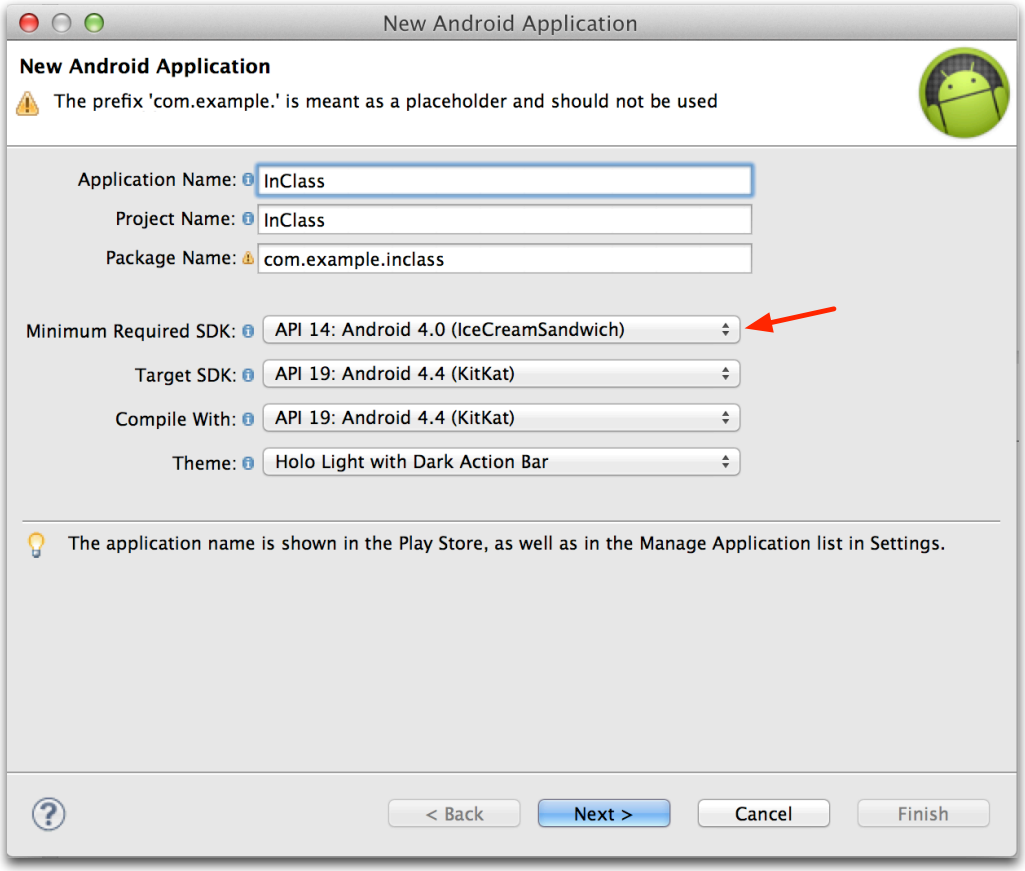

Fig 1. Choosing Minimum Required SDK to 14

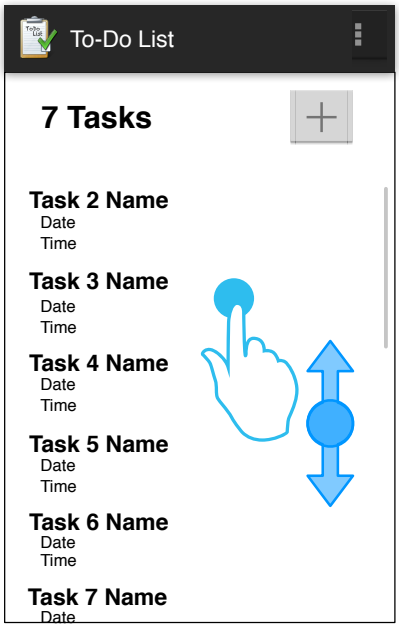

Figure 2, Main Activity Wireframe

Main Activity (35 Points)

The Main Activity displays all the current tasks, as show in Figure 2. The Main Activity should maintain a LinkedList of the current Task Objects. The activity requirements are as follows:

- The task list should be stored in the Main Activity. This list should be displayed using a combination of ScrollView and LinearLayout, where each row could be a LinearLayout containing TextViews for the Title, Date, and Time. **You should not use a ListView in this assignment.** The goal is to expose you to the managing a list without using the ListView component. See Figure 2.
- At the top of the Main Activity should display the total number of tasks stored, which changes dynamically as tasks are added or removed, and there should have a "+" button to allow the creation of new tasks.
- Clicking the "+" button should start the Create Task Activity, you should consider using the start activity for result, as the Create Task Activity should send back the created task to the Main Activity. Upon returning from the Create Task Activity the list of tasks should be updated and redisplayed to show the newly added task in displayed layout.
- Clicking a task should start the Display activity. If the Display activity changes or deletes the task, then this change should be reflected on the Main activity upon returning from the Display activity.

Create Task Activity (15 Points)

The Create Task Activity is displayed in Figure 3. Implement the following requirements:

- When the user selects "+" button in the Main Activity, the Create Task Activity should be started as shown in Figure 3.
- The task title should not exceed 20 characters. The date and time EditTexts should not be editable, you should set their key listeners to null, (setKeyListener(null)). When the date and time EditTexts are the corresponding date picker or time picker dialog boxes

should be displayed to enable the user to pick the date or time respectively. Figure 3, shows the date and time picker displayed when the date and time edit text is clicked. The selected time and date should be displayed in their corresponding edit text as shown in Figure 3.

- For information related to the time picker check http://developer.android.com/reference/android/app/TimePickerDialog.html
- For information related to the date picker check http://developer.android.com/reference/android/app/DatePickerDialog.html

- Clicking the Save button should save the fields and selected Priority level as a Task object and send it back as a result to the Main Activity so that it can be stored and displayed in the task list.

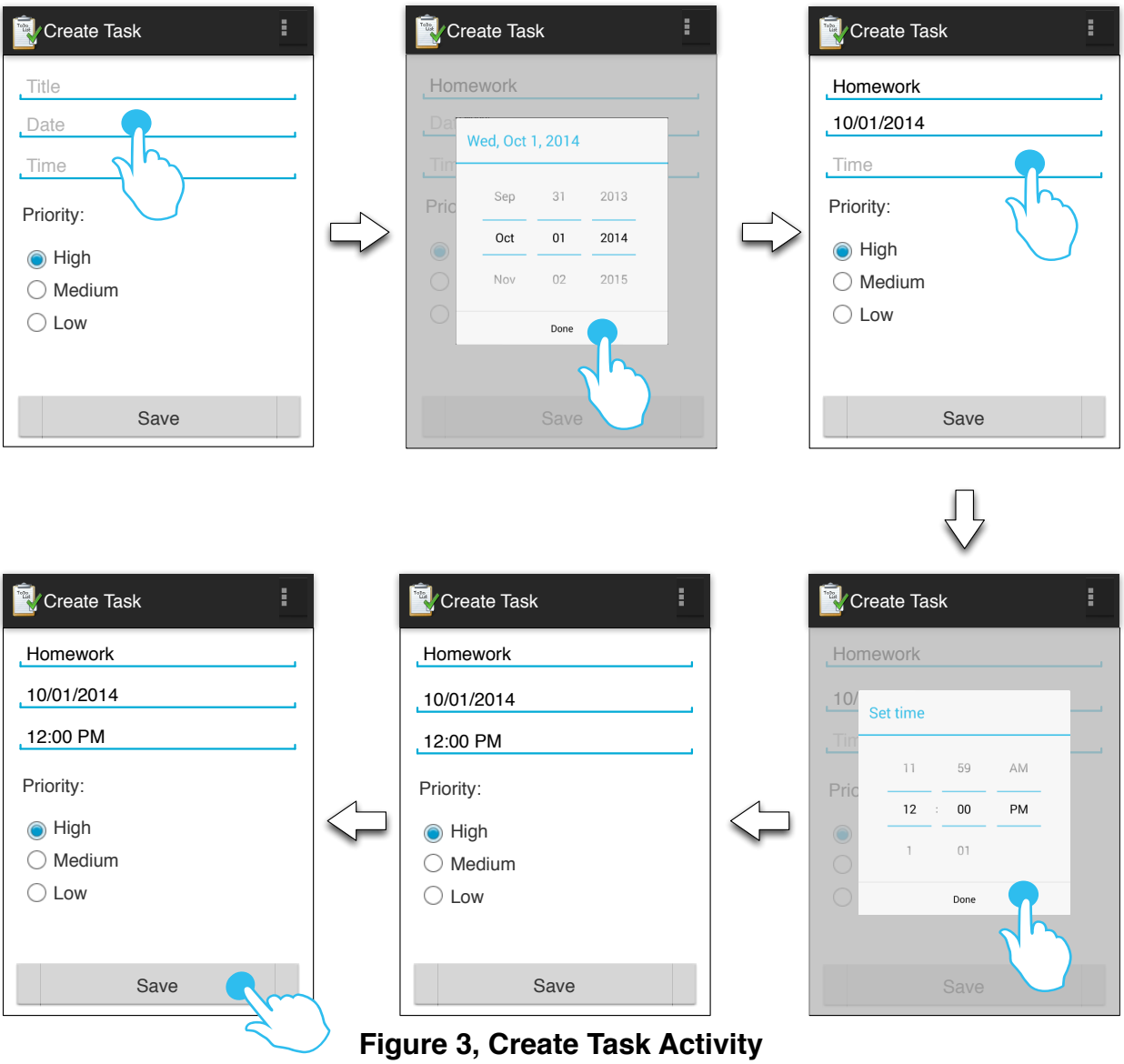

Figure 3, Create Task Activity

Display Activity (30 Points)

The Display Activity is displayed in Figure 3. Implement the following requirements:

- When the user selects a Task in the Main Activity, the Display Activity should be started as shown in Figure 3. This activity enables the user to view selected the task information.
- At the bottom of the activity should be two buttons, one for "Edit" and one for "Delete.
- Clicking the "Edit" button should start the Edit Task Activity. Upon returning from the Edit Task Activity the Display Activity should display the updated edited task.
- Clicking the "Delete" button should finish the Display Activity and return to the Main Activity, in addition the Main Activity should be notified of the deletion of the task in order to delete this task from the task list and update the displayed list. Figure 4, shows the flow of deleting a task. The task list should be updated to show the remaining Tasks, and the counter at the top of the activity should reflect the new count of tasks.

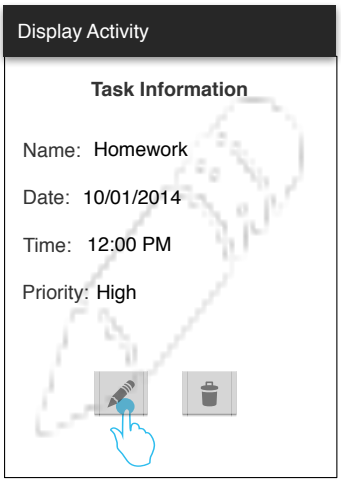

Figure 3, Display Activity

- The Display Activity should return to the Main activity when the back button is pressed. It should also notify the Main Activity of any changes performed on the task (edits if any).

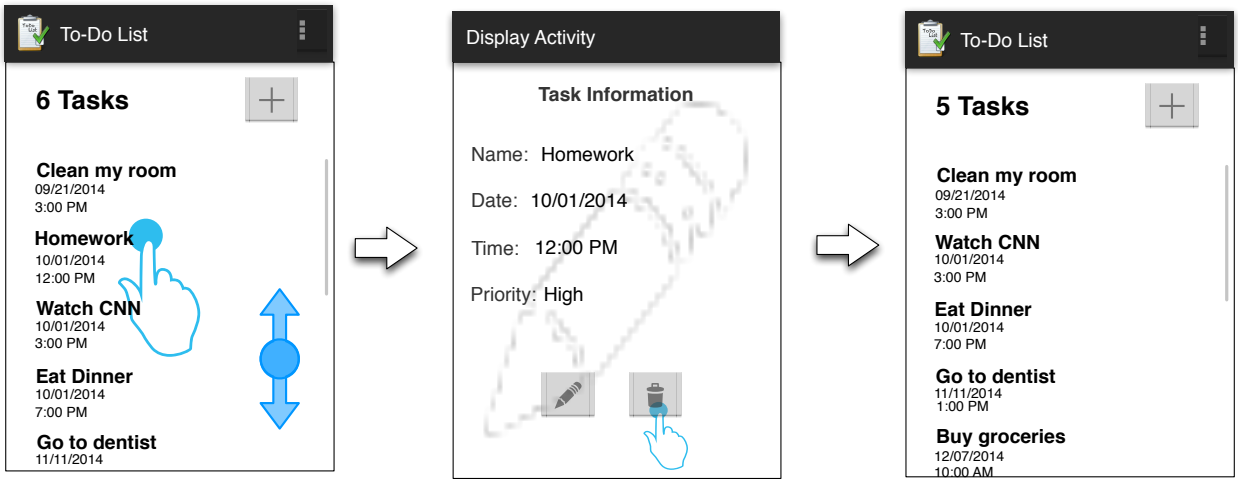

Figure 4, Deleting a task

Edit Task Activity (20 Points)

The edit task activity is displayed in Figure 5. Implement the following requirements:

- When the user selects a Task in the Main Activity, and the the "Edit" button in the Display Activity, the Edit Task Activity should be started as shown in Figure 5.
- The task info should be populated in the Edit Task Activity as shown in Figure 5.

- The UI for this activity is similar to the Create Task Activity and should follow the same text size, and time/date picker implementation and restrictions.
- Clicking the Save button should save the fields as a Task object and send it back as a result to the Display Activity.
- When the Display Activity is closed using the back button, it should return to the Main Activity and send it the required information in order to update this task in the Main Activity and to display the updated task in the task list.

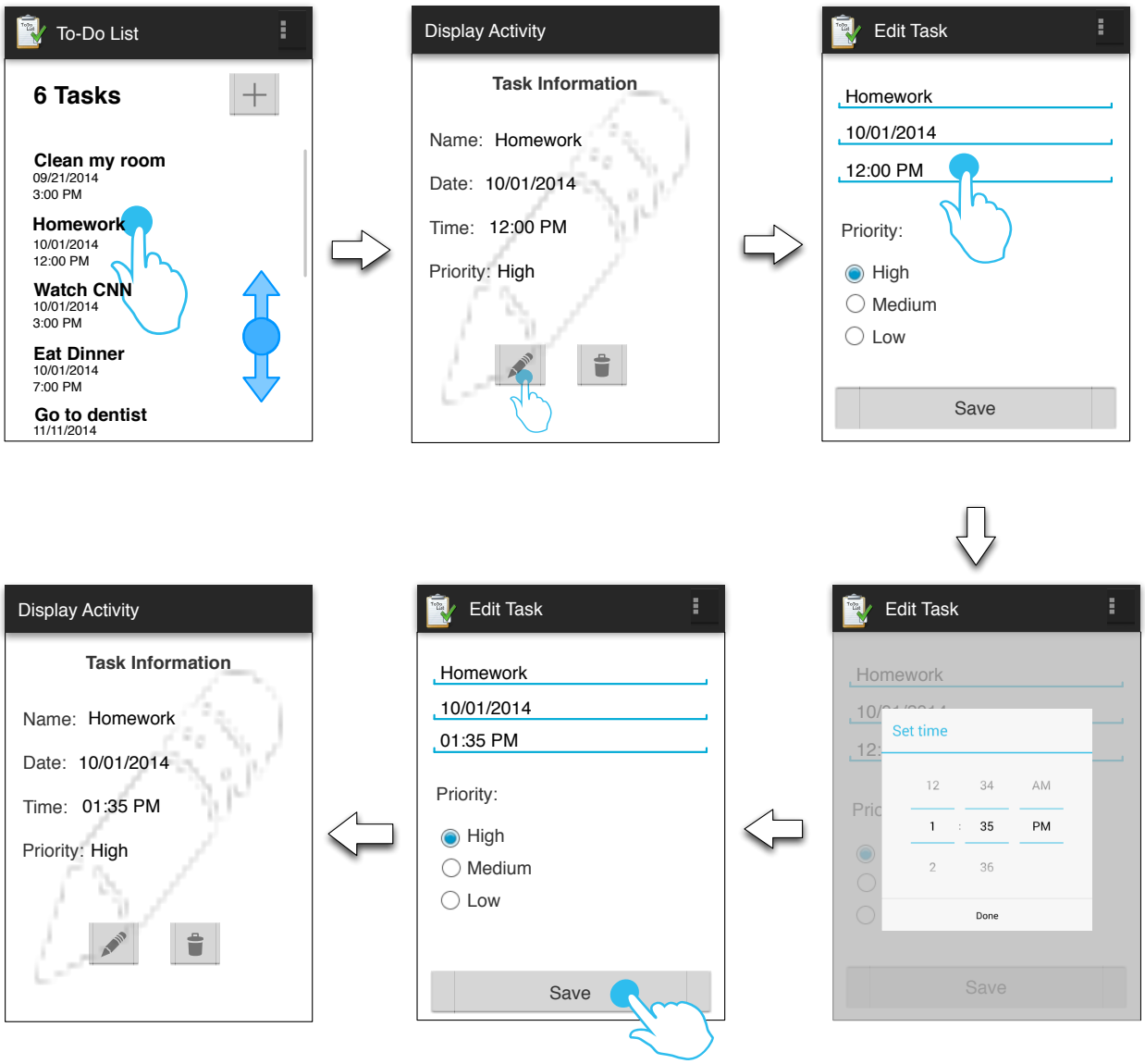

Figure 5, Edit Task Activity