

ITIS/CS 5180 Mobile Application Development  
Homework 2

---

**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 group should submit only one assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Export your Android project and create a zip file which includes all the project folder and any required libraries.
5. Submission details:
  - a. Compress the contents of your project folder. The file name is very important and should follow the following format: **HW2\_Groups#.zip**
  - b. Only one group member is required to submit on behalf of the whole group.
  - c. You should submit the assignment through Canvas: Submit the zip file.
6. The required Android Virtual Device (AVD) should have **minimum SDK version set to 20 and target SDK at 25**.
7. **Failure to follow the above instructions will result in point deductions.**

## Homework 02 (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 1.

### Important App Requirements:

1. 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.**

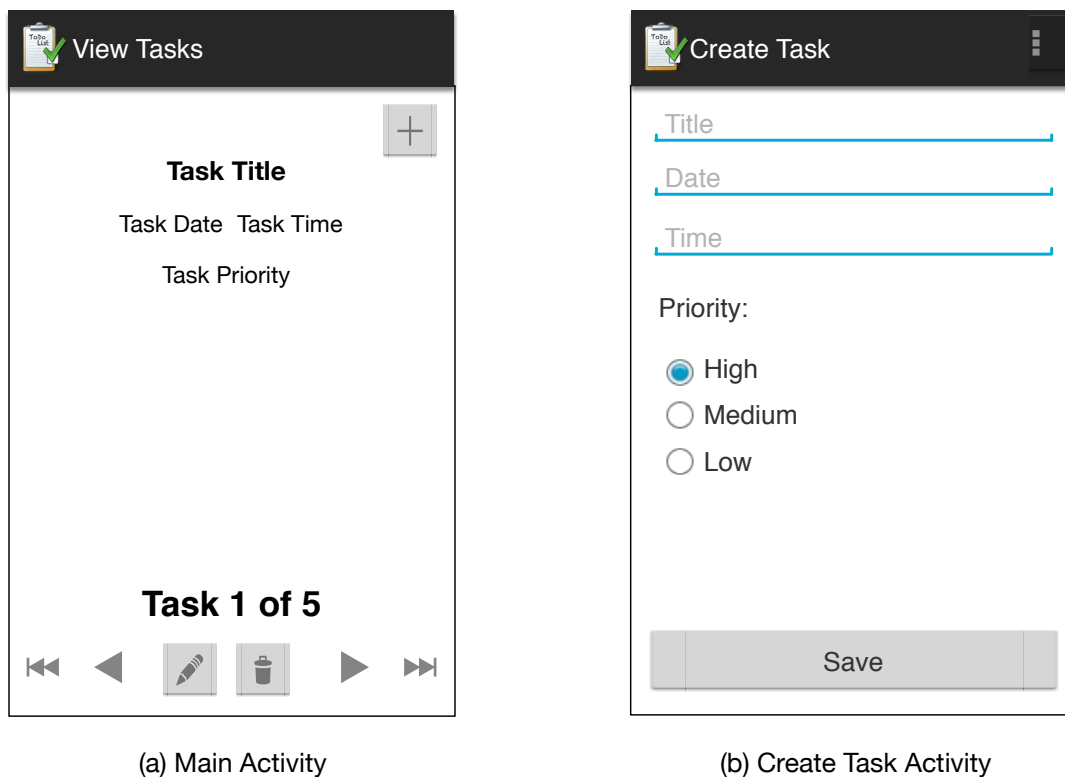


Figure 1, App Wireframe

### Main Activity (50 Points)

The Main Activity (View Tasks) enables the user to display the currently stored tasks as show in Figure 1(a). The Main Activity should maintain a LinkedList of the current Task Objects. The activity requirements are as follows:

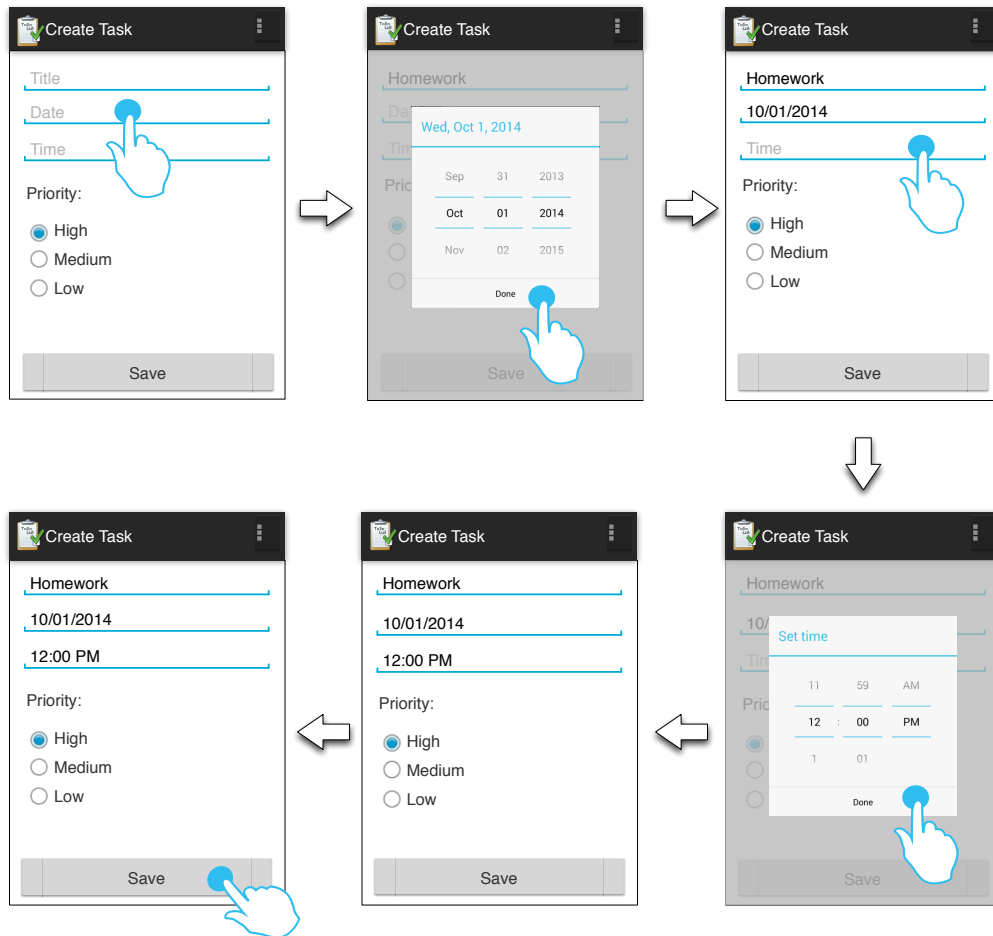
- The task list should be stored in the Main Activity. The task list should be ordered by task date and time, ordered in **ASCENDING** order.

- Each stored task should be displayed as shown in Figure 1(a). Note the buttons at the top and bottom of the screen.
- At the bottom of the interface should indicate “Task M of N”, where M is the current task index and N is the total number of tasks. These numbers should be dynamically updated when tasks are added or deleted, or you navigate to another task.
- The buttons at the bottom of the interface described from left to right:
  - **Goto first task:** Should display the first task stored in the linked list.
  - **Goto previous task:** Should go to the previous task. If the currently displayed task is the first task, then display a toast indicating that the current task is the first task.
  - **Edit current task:** Should display the Edit Task activity.
  - **Delete current task:** Should delete the currently displayed task, update the interface and display the first task.
  - **Goto next task:** Should display the next task, if the currently displayed task is the last task in the list then should display a toast indicating that the current task is the last task.
  - **Goto last task:** Should display the last task stored in the linked list.
- At the top of the Main Activity should display 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.

### **Create Task Activity (30 Points)**

The Create Task Activity is displayed in Figure 2. Implement the following requirements:

- When the user selects “+” button in the Main Activity, the Create Task Activity should be started as shown in Figure 2.
- 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 2, 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 2.
  - 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. The Main Activity should navigate to the newly stored task and display it.

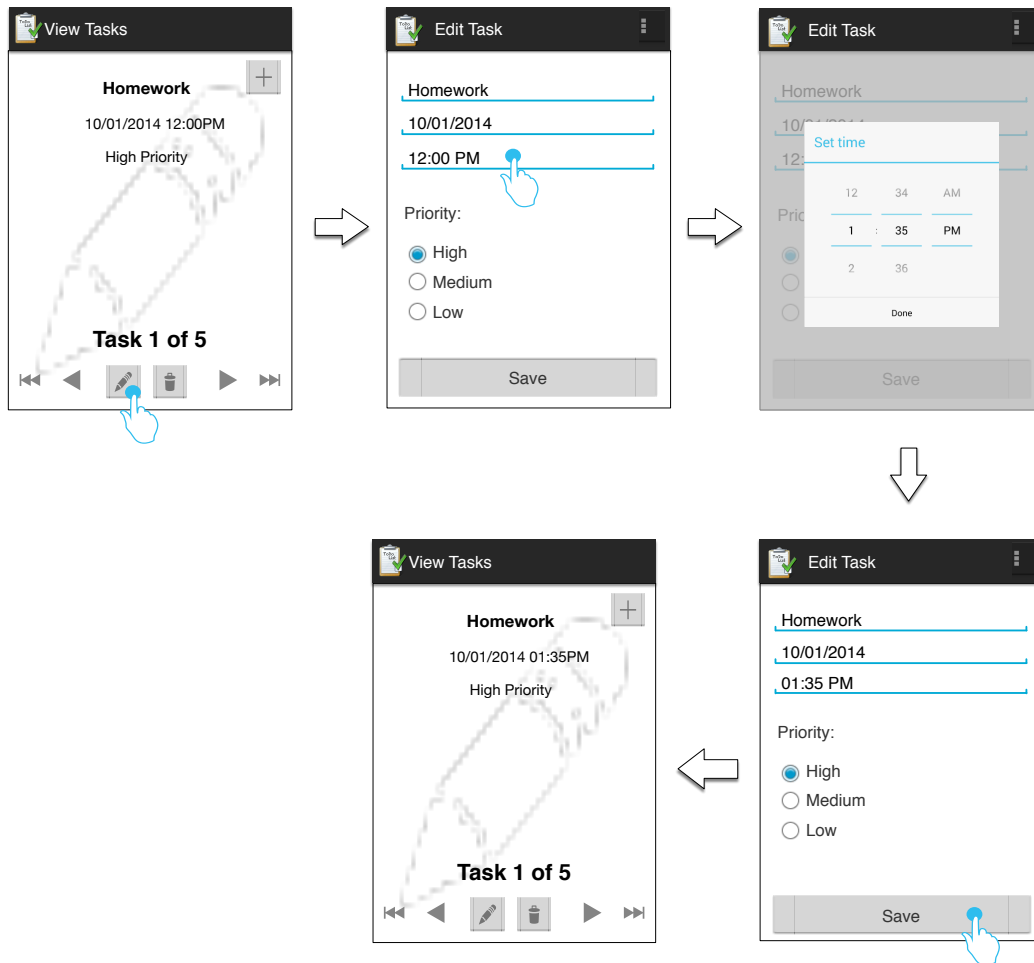


**Figure 2, Create Task**

### **Edit Task Activity (20 Points)**

The edit task activity is displayed in Figure 3. Implement the following requirements:

- When the user selects a Task in the Main Activity, and clicks the “Edit”, the Edit Task Activity should be started as shown in Figure 3.
- The task info should be populated in the Edit Task Activity as shown in Figure 3.
- 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 Main Activity. The Main Activity should display the edited task to show the latest updates.



**Figure 3, Edit Task**