Mobile App Development

In-Class Assessment 2

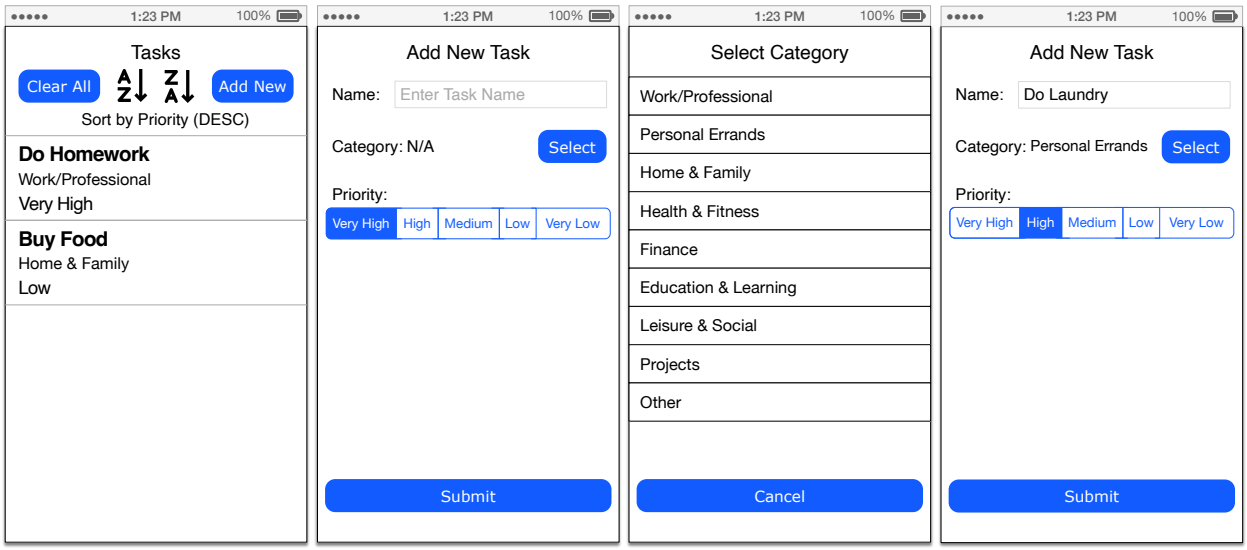
Basic Instructions:

1. This is and In Class Assessment, which will count for 8% of the total course grade.
2. This assessment is an individual effort. Each student is responsible for her/his own assessment and its submission.
3. Once you have picked up the assessment, you may not discuss it in any way with anyone until the assessment period is over.
4. During the assessment, you are allowed to use the course videos, slides, and your code from previous home works and in class assignments. You can use the internet to search for answers. You are NOT allowed to use code provided by other students or solicit help from other online persons.
5. Answer all the assessment parts, all the parts are required.
6. During the assessment the teaching assistants and Instructors will pass by each student and ask them to demonstrate their application. Your interaction with the teaching assistants and instructors will be taken into consideration when grading your assessment submission.
7. Please download the support files provided with the assessment and use them when implementing your project.
8. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
9. Create a zip file which includes all the project folder, any required libraries, and your presentation material. Submit the exported file using the provided canvas submission link.
10. **Do not try to use any Social Messenger apps, Emails, Or Cloud File Storage services in this exam.**
11. **Failure to follow the above instructions will result in point deductions.**
12. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

In-Class Assessment 2 (100 Points)

In this assignment you will be building an application that uses UITableViews to manage a tasks list. The app requirements are as follows:

1. Please use the provided skeleton app.
2. The array of Task objects should be stored in the Tasks UIViewController to maintain the list of tasks added.



(a) Tasks Screen **(b) Add New Task** **(c) Select Category** **(d)Add New Task**

Figure 1, Application User Interface

Part 1, Tasks UIViewController (45 Points):

This UIViewController displays the UITableView of tasks as shown in Figure 1(a). Requirements are:

1. This UIViewController should store and maintain a mutable array of Task objects to be used to enable the addition and deletion of tasks.
2. The array of Tasks should be be displayed in a UITableView as shown in Figure 1(a).
   a. Each row item should display the name, category and priority.
   b. Implement the Delegate and DataSource required by the UITableView.
3. Clicking the “Clear All” button should:
   a. Remove all the tasks from the tasks array.
   b. Refresh the UITableView to display no tasks as the tasks array is empty.
4. Clicking the “Add New” button should segue to the Add Task UIViewController, which should be presented modally.
   a. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the newly created user back from the Add User UIViewController.
   b. Upon receiving the new Task object, add the Task object to the tasks array, and refresh the UITableView to display the updated list of tasks.
5. Clicking the “Z→A” button should:
   a. Sort the list of Tasks in descending order by task priority, where Very High is the largest value and Very Low is the lowest value.
   b. Refresh the UITableView to display the sorted array of Tasks.

c. Update the UILabel under the buttons to indicate that “Sort by Priority (DESC)” as shown in Figure 2(a).

6. Clicking the “A→Z” button should:
   a. Sort the list of Tasks in ascending order by task priority, where Very High is the largest value and Very Low is the lowest value.
   b. Refresh the UITableView to display the sorted array of Tasks.
   c. Update the UILabel under the buttons to indicate that “Sort by Priority (ASC)” as shown in Figure 2(b).
7. Clicking on the row item should:
   a. Segue to the Task Detail UIViewController which should be presented modally.
   b. Send the selected Task object to the Task Detail UIViewController.

Part 2, Add New Task UIViewController (20 Points):

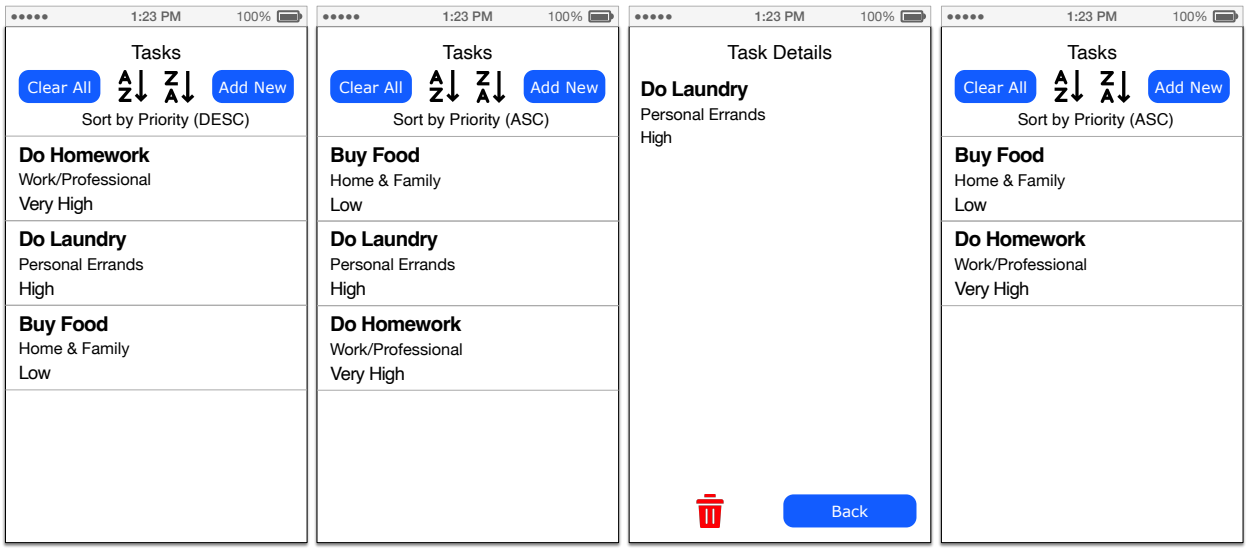
This UIViewController is used to add a new task. The requirements are listed below:

1. Clicking a “Select” button should:
   a. Segue to the Select Category UIViewController.
   b. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the selected value back from the Select Category UIViewController.
   c. Upon receiving the selected category value, the value selected should be displayed as shown in Fig 1(d).
2. Clicking the Submit button should:
   a. If any of the entries is not entered or selected then show an Alert Dialog message indicating the missing input.
   b. If all the required data is entered and selected then create a Task object which contains all the entered and selected values. Send the created Task object back to the Tasks UIViewController using any of the approaches such as Unwind Segue or Notification Center or Protocol/Delegate. If needed dismiss this UIViewController.

Part 3, Select Category UIViewController (20 Points):

This UIViewController allows the user to select a category value as shown in Fig 1(c). The requirements are listed below:

1. List the category selections in a UITableView. The category selections can be retrieved using the provide Data class.
2. Clicking a list item should:
   a. Send the selected item back to the Add Task UIViewController. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the selected value back.
   b. Dismiss this UIViewController if needed.
   c. Display the selected category value in the Add Task UIViewController as shown in Figure 1(d).
3. Clicking “Cancel” should simply dismiss this UIViewController.



(a) Tasks Screen **(b) Tasks Screen (ASC)** **(c) Task Details** **(d) Task Deleted**

Figure 2, Application User Interface

Part 4, Task Details UIViewController (15 Points):

This UIViewController displays the selected task details as shown in Figure 2(c). The requirements are listed below:

1. This UIViewController should receive a Task object from the Tasks UIViewController
2. Display the name, category and priority as shown in Figure 2(c).
3. Clicking “Delete” should:
   a. Send the Task object back to the Tasks UIViewController using any of the approaches such as Unwind Segue or Notification Center or Protocol/Delegate. If needed dismiss this UIViewController.
   b. The Tasks UIViewController should delete the received Task object from the tasks array, and should reload the UITableView to display the updated array. See Figure 2(d).
4. Clicking “Back” should simply dismiss this UIViewController.

Section:	
Student Name:	
Student ID:	

Part #	Features	Total	Grade
Part 3	Select Category: Displays the list of categories. Sends the selected category value back to the Add Task UIViewController. The selected value is sent to the Add Task UIViewController and is displayed in the Add Task UIViewController.	20	
Part 2	Add Task: Validation and sends the new Task object back to the Tasks UIViewController. The new task is added to the tasks array in the Tasks UIViewController.	20	
Part 1	Tasks: Using a UITableView, displays the list of tasks.	15	
Part 1	Tasks: Clear all - Deletes all the Tasks in the array of tasks and notifies the UITableView to reload and refresh the displayed list.	10	
Part 1	Tasks: Sorts ASC, sorts the Tasks array in ascending order by priority and notifies the UITableView to reload and refresh the displayed list.	10	
Part 1	Tasks: Sorts DESC, sorts the Tasks array in descending order by priority and notifies the UITableView to reload and refresh the displayed list.	10	
Part 4	Task Details: Receives and displays the Task object. The delete feature, deletes the Task object from the Tasks array in the Tasks UIViewController, and the Tasks UIViewController reloads to show the updated list.	15	
	Total	**100**	
Table 1: Grading Key			