1 Problem Definition

Your goal in this assignment is to create a timer application. There are three view controllers created for this application, the Main View Controller, the Add Timer View Controller and the Countdown View Controller as shown in Fig 1(a), Fig 1(b) and Fig 1(c).

In the Main View Controller, a list of timers are listed. Each timer are associated with an "event", a "location" and a "remaining time (s)", which means some event will happen in some location with some remaining time.

When selecting on one timer from the list, the Countdown View Controller will show and the timer is triggered as shown in Fig. 1(c). The value of the remaining time will be decreased by 1 in every second until it equals to 0.

Once you go back to the Main View Controller, the counting down will stop and the remaining time shown in the Main View Controller should be updated correspondingly.

Also, you can add a new timer by clicking on the "+" button of the Main View Controller.

The hard part of this assignment will be using Multithreading to implement the count down of the timer.

2 Detailed Instructions

- Create a Single View application project named <lastName><firstName>-HW6.
Create a **Timer** class with 3 properties: event, location and remaining time.

**The Main View Controller (Fig. 1(a)):**

- In the storyboard, select the Main View Controller and embed it in a navigation controller by "Editor - Embed in - Navigation Controller".
- Drag a table view into the Main View Controller and create a custom table view cell "TimerTableViewCell". There are 3 labels in a Timer Table View Cell, containing information of the event, the location and the remaining time of the timer.
- Create the ”+” button on the Navigation Bar of the Main View Controller.

**The Add Timer View Controller (Fig. 1(b)):**

- Ctrl+drag the ”+” button of Main View Controller to the Add Timer View Controller.
- Add 3 labels and 3 textfields for user input. When the ”save” button is clicked, a new timer should be created with information from the textfields.

**The Countdown View Controller (Fig. 1(c):**

- Ctrl+drag the timer table view cell to the Countdown View Controller.
- The event, location, remaining time of the selected timer should be shown here.
- In order to implement the counting down, we need Multithreading techniques. When the Countdown View Controller is shown, create a thread with a while loop running. Within the loop, the sleep() function is called and then decrease the remaining time of the timer. If the remaining time equals to zero, then jump out of the loop.
- **The user should be able to go back to the Main View Controller anytime and the countdown should stop.** In this case, add some code in the viewWillAppear() function to help stop the countdown. The remaining time of the selected timer in the table view of the Main View Controller should be updated with the new value.

You can use any techniques to implement data passing (not limited to Delegates/Protocols). For example, you can create a global array to contain all the timers that every View Controller has access to. This way of data sharing is not safe and not encouraged but you can do it for this homework.

### 3 Grading criteria

1. You have UI components as defined. (20%)
2. The Main View Controller is implemented as well as the Timer Table View Cell. (30%)
3. The Add View Controller is implemented such that you can add new timers and the Main View Controller should update corresponding to the new timer. (20%)
4. In the Countdown View Controller, the value of remaining time is decreased by 1 in every second. (20%)
5. When the user goes back to the Main View Controller from the Countdown View Controller, the timer should be stopped and the remaining time shown in the Main View Controller should update correspondingly. (10%)
6. **Note that if the app does not build and run, ZERO points will be given.**
7. The Coding Standard is followed. One point deducted for each violation.
4 General criteria

1. I will be looking for good documentation, descriptive variable names, clean logical structure, and adherence to all coding conventions expected of an experienced programmer, as well as those outlined in the Coding Standard document. There will be penalties for failure to meet these standards.

2. Your code must compile and run before submission.

3. Xcode will automatically generate standard headers to your .swift files. Add two lines to each Swift file that list your EID and the course number, so that the header looks like the following:

```
//
// Filename
// LastnameFirstname-HW5
// EID: xxxxxx
// Course: CS371L
//
// Created by xxxxxx on x/xx/18.
// Copyright 2018 xxxxxx. All rights reserved.
//
```