**ITESM CAMPUS QUERÉTARO**

# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

## DEPARTAMENT OF COMPUTER SCIENCE

## Software Architecture

# Assignment for Partial 3 To-do List with built in timer

The last part of the course is going to be more practical and involve a programming assignment. The purpose is to use this project to learn and practice design patterns in addition the assignment code can be used to Refactoring.

Every student have to create an application that is task list manager (To Do List) that will be integrated with a small timer application must have the functionality of a **stop watch** and a **countdown timer**. The application must be a standalone application that can run on any windows desktop. The design must use the **MVC** architectural pattern (or a variance of the MVC pattern). The idea is that every student will try to use as many of the design patterns in the application as possible.

Any development language of preference can be used.

**Deadline: 7:00 PM - Friday Nov 18.**

**Functional Requirements.  
  
Tasks**

1. A list of task must be displayed. This list can be loaded from a local file.
2. The tasks must have a priority (High, medium, Low) that will determine the order.
3. Tasks can be add, edit, delete and mark complete.
4. Tasks must have two timer properties.
   1. Total running time. This will be the total time that you worked on the task for the day.
   2. Interval / task time. This will be the time the timer takes before it sets an alarm.
5. Tasks must be saved to a file.
6. A task will set off an alarm when the time is finished. This could be a sound, dialog box, both or other.

**Timer**

1. Must have a **stop watch** that counts time upwards with the option
   1. Unlimited - Continue to count upwards until stop.
   2. Upper limit that it will get from the selected task.   
      For example it task #2 is selected and it has the Interval / task time set to 10 minutes then it must stop at this point and set the alarm.
2. Must have **countdown timer.** 
   1. **It will get the time from the selected task that will be the starting point.**
   2. It will set off an alarm when it hits zero.
3. Must have controls to set time, reset, ect.

**Technical Requirements.**

1. The application must be a standalone application that can run on a windows desktop.
2. The **MVC pattern** must be used to as the overall architecture.
3. You can chose any object orientated programming language but **strictly NO frameworks are allowed.**
4. There must be a settings or properties file that can contain the preferences. A Property Manager Class / object must be a **Singleton**.
5. The GUI must use the **Observer** and **Command pattern**s for user events.
6. The GUI must use the pattern for
7. A **Factory Pattern** must be used to create new tasks.
8. The Controller must be an isolated component meaning that it must not have code of the GUI. This is so that in future any GUI can use it. The GUI will interact with the Model calling methods and using the **Observer pattern**.
9. The **Iterator pattern** must be implemented to run through the tasks.
10. The **Strategy pattern** must be used to assign the ‘alarm’ to a task.
11. Students must also use and implement any of the other patterns where identified.

Deliverables

1. Each student must provide a running application with a video demo of the application and the design patterns used.
2. The source code must also be submitted via email to complete this assignment.
3. Documentation must be added to provide details of where the patterns was used.