**Assignment 10** (Due at the final exam)

Sleep deprivation may have serious health consequences. The reality is when we have too much sleep debt, we may not function well in our daily life; and much less enjoy it fully. In this assignment, you are to design and implement a sleep pattern monitoring system (or a “super alarm clock") for university studentsto help them manage their own sleep patterns. (As a byproduct, the system could also provide data to support a research inquiry into the extent of the problem in certain population). Here are the system requirements. The system shall:

* Allow a student to set an alarm to wake the student up. (THE TIME THEY WANT TO WAKE UP)
* Allow a student to set an alarm to remind the student to go to sleep.
* Record when a student tells the system that she is about to go to sleep.
* Record when a student tells the system that she has woken up, whether it is due to an alarm.
* Make recommendations as to when a student needs to go to sleep. This should include "yellow alerts" when the student will need sleep soon, and "red alerts" when they need to sleep now.
* Store the collected data in a text file for now. (Because a data container with database process capability could be implemented later, you should design a data access interface although you implement the interface is based on a text file.)
* Provide reports to students showing their sleep patterns over time, allowing them to see how often they have ignored alarms, and identify clusters of dangerous, or beneficial, sleep behavior (for instance, a report might display histograms showing number of yellow and red warnings student received each month during a given time period).

***Assignment Requirements:***

1. Design and implement this system.
2. Use 4+1 view model to provide architectural views of the system (including use case scenarios, activity, state, and dataflow diagrams).
3. Implement a GUI interface with a design of your own.
4. Saved or retrieve data from a text file.

**We will do presentation next week to look at each other’s design, so prepare the following:**

* Artifacts of the architectural views
* A use case scenario
* Skeleton code to test the design (without much logical detail implemented)

***A Technical Note:*** How can we track elapsed time (to run our own clock as opposed to a real-time clock)? The following loop simulates passage of a day, one hour at a time (you can also do one minute or even one second at a time with a thread to allow appropriate time gap needed for visualizing clock ticking).

*DateFormat df = DateFormat.getDateInstance(DateFormat.SHORT);*

*GregorianCalendar now = new GregorianCalendar();*

*System.out.println(df.format(now.getTime()));*

*GregorianCalendar oneDayLater = new GregorianCalendar();*

*oneDayLater.add(GregorianCalendar.DATE, 1);*

*System.out.println(df.format(oneDayLater.getTime()));*

*while( now.before(oneDayLater) ){*

*now.add(GregorianCalendar.MINUTE, 60);*

*System.out.println(now.getTime()); }*

You may find more information about Java Date() class and GregorianCalendar() class at (for example):

<http://www.javaworld.com/article/2076270/core-java/calculating-java-dates.html> <http://www.javaworld.com/article/2075174/core-java/working-in-java-time.html>