CS3560 Homework 3

Be sure to read this document carefully. You are responsible to read and understand all of these instructions. If you have questions, be sure to ask, either in class, by email, or at office hours.

Description

Homework 3 is an extended group project. The class will be divided into groups of about 4 people. The smallest sized group would be 3 people, the largest about 6 people. There are several phases to this homework, and it is expected that everyone in the group participates and contributes to the work.

Your group will be designing a PSS, which is described below. You will need to do the following things:

1. During today's class, your group will be doing the CRC card design for the system, determining all of the classes you will need for the project, and assigning their responsibilities (both what they know and what they do), and also list the collaborators. You need to present this before leaving today, so we can do a sanity check on your design.
2. Your group will then produce a Class Diagram for your system. In addition to listing all of the classes and their relationships, you will also list all the essential attributes of each class and also show the methods, or actions, that each class performs.
3. For any class that performs complex actions, and you should have at least two or three that do, you need to provide a state diagram showing how the object will perform that complex action.
4. Next you will make some Use Cases documenting the various types of interaction that the user can have with PSS.
5. For each Use Case, draw some Sequence Models showing typical scenarios using PSS.
6. For any complex operations, draw Activity Diagrams showing how these operations are implemented through the actions of various objects.
7. Homework 3 consists of all of these Diagrams. Homework 3 is due on Apr 13, 2021.

So what is PSS?

PSS is a tool that will assist the user to schedule his/her activities. It will take various "tasks" as input and schedule them according to the user's needs. Typical tasks would be *attending class*, *studying*, *working on assignment*, and so on. Typical outputs will be daily, weekly, or monthly schedules. PSS also has commands for storing the list of tasks to a data file, or to read those tasks from a data file.

The user will interact with PSS to enter a new task to the system. There are different types of tasks (*see below*), but each task will have a start time and a duration. The times and durations should be rounded to the nearest 15 minutes. If the user attempts to create a task that overlaps an existing task, PSS will report the overlap and will not create the new task.

Some of the tasks are *recurring tasks*. These tasks occur on a repeating basis, from a particular start date to a given end date. For example, one task might be for one hour and 15 minutes, every Tuesday evening at 7:00 p.m., from January 28th to May 5th.

Another type of task is a *transient task*, which only occurs one time.

A third type of task is an *anti-task*, which cancels out one particular occurence of a recurring task. For example, an anti-task might be set for February 25th, for an hour and 15 minutes starting at 7:00 p.m. This task would need to refer to the recurring task. *Note that if an anti-task removes one instance of a recurring task, then a transient task could be scheduled at that same time*.

Recurring tasks can be further subdivided into *Course*, *Study*, *Sleep*, *Exercise*, *Work*, and *Meal*. Your group can suggest other types of recurring tasks.

Transient tasks can be further subdivided into *Visit*, *Shopping*, and *Appointment*. You may also add other types of recurring tasks.

Design

Your PSS should use the *Model-View-Controller* pattern, which we haven't talked about yet, but here is a brief description. Your implementation will have three primary categories of objects: a Model, a Viewer, and a Controller.

- The Model manages the data, which in this case is the list of all of the tasks. The model can be asked to return various lists of tasks, and can be asked to create, edit, or delete tasks. The model makes sure that there are no errors, such as overlapping tasks or tasks that are too large or too small.
- The Viewer (and you may have multiple objects that are Viewers) are used to display some information. This is how the user can see a schedule or a part of a schedule.
- The Controller (and again there might be multiple of these) is the main object that interacts with the user. The controller decides what actions to take.

You will have other classes as well, such as the hierarchy of tasks supported by the system. You may, if you wish, have a class for writing the data to a file, and another class for reading the data. Alternatively, you can have your primary Model class perform these activities.

Collaboration vs Cheating

Recall that Cal Poly's Academic Integrity policy states that all homework should be your own work. You should not turn in someone else's work with your name on it. However, this is a group project, you will work on the assignment together, turning in joint work.

Grading

This homework is worth 5 points (5% of your grade for the course). Your score will be computed as follows:

- **1 point:** Do you have a comprehensive set of Classes?
- **1 point:** Do your State Diagrams adequately describe the behavior of the complex routines?
- **1 point:** Does your Interaction Modelling show how PSS interacts with the user?
- **1 point:** Have you covered all of the requirements?
- **1 point:** Did you follow all of the instructions?

Turning In Homework

You will submit your homework to Blackboard.

Due Date

The homework is due on **Apr 13**, by the end of the day.