CSCI 2194 (Professional, Ethics, and Design Seminar), Spring 2011

Example Design Project: Requirements Analysis

1 Use Case Analysis

This system has two classes of users — advisors and advisees — so it seems to make sense to model each of them as an "actor" in use-case terminology. Use cases for the system include the following. ("Schedule" here will mean a list of time slots for advising appointments, each of which is either associated with an advisee or open/available.)

Create schedule. The advisor starts the application and supplies some sort of authentication (username and password?). The advisor creates a list of time slots (appointments, initially all open) and saves it. Exactly what the user interface should be is not clear at this point.

Modify schedule. The advisor starts the application and supplies some sort of authentication (username and password?). The advisor "opens" a previously-created schedule and modifies it as desired. The advisor should be able to change not only the list of time slots but the status of each (open or claimed by an advisee). Here too the details of the user interface are not completely specified.

View schedule. The advisor or advisee starts the application, possibly supplying authentication. The schedule (time slots and status of each) is displayed.

Sign up for advising time. The advisee starts the application and supplies some sort of authentication (username and password?). The schedule is displayed. If the advisee is already signed up for a time, an error occurs (and/or the system shifts to the "change advising time" use case). The advisee chooses an open time slot. The schedule is changed to reflect that the advisee has "claimed" that time.

Change advising time. The advisee starts the application and supplies some sort of authentication (username and password?). The schedule is displayed. If the advisee is not already signed up for a time, an error occurs (and/or the system shifts to the "sign up for advising time" use case). The advisee chooses an open time slot. The schedule is changed to reflect that the advisee has "claimed" that time, and the advisee's previous time slot becomes free.

It seems worth keeping in mind that if the system allows concurrent access by multiple users there needs to be some synchronization, for example to make sure two advisees do not claim the same time slot.

2 Use Case Diagram

The diagram in Figure 1 shows the use cases for the system.

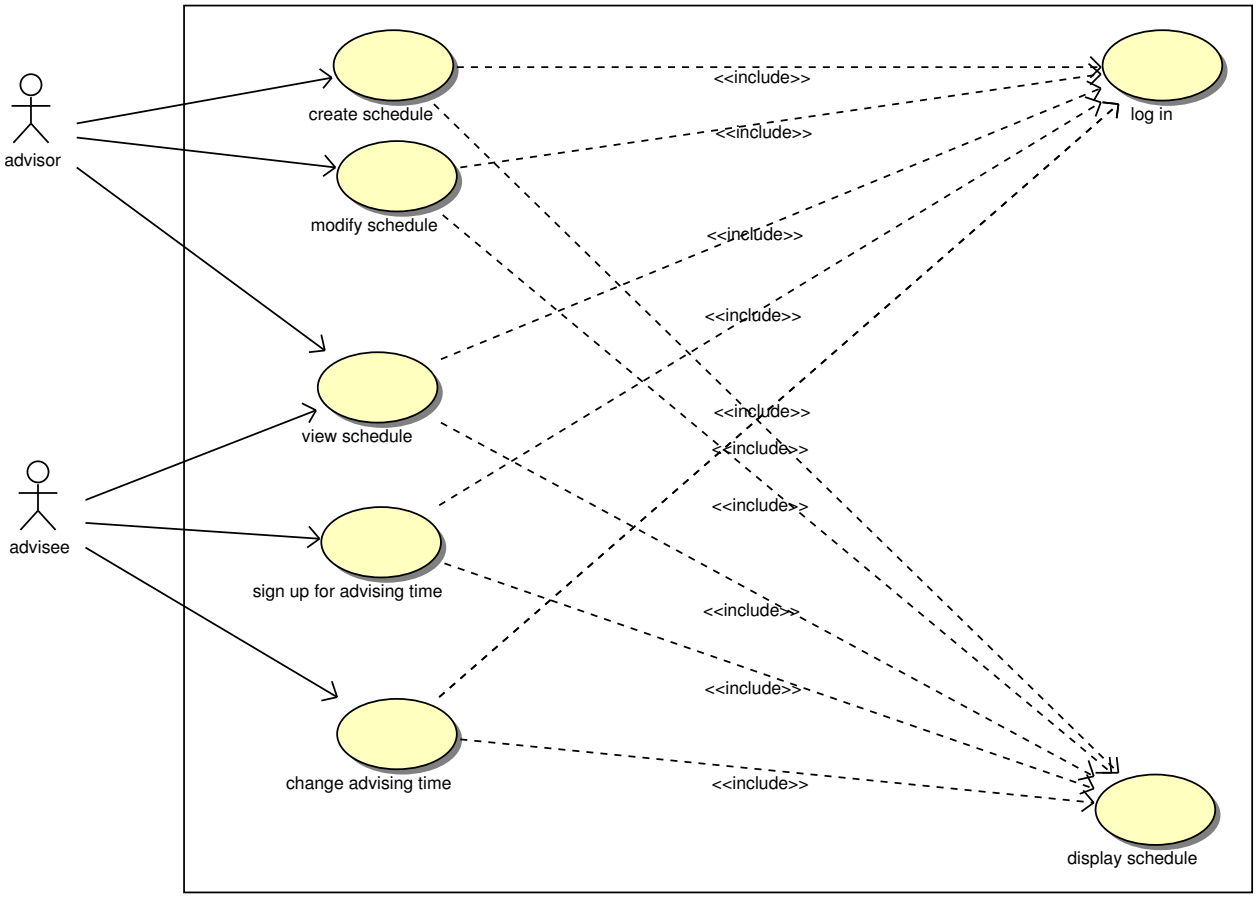

Figure 1: Use cases for the system. Notice use of "include" to factor out what will likely be common code.