Project Description

CS3910: Spring 2018

1. Introduction

The Department of Computer Science is currently seeking industry funding to support an individualized tutoring service for students enrolled in our three introductory classes: CS1020, CS1110, and CS1210. The idea is to hire tutors from among our own juniors and seniors. Each will be paid an hourly fee to staff a fixed number of half-hour help sessions with individual students, where a help session will focus on either clarifying material from the lecture, or working on a programming problem.[1]

Your job is to design and implement a system to support this new tutoring service as described here. Note that the system is quite extensive and featureful: part of your task is to prioritize the different functional aspects of the systems so as to ensure that you have at least a minimal system completed by the end of term. You must decide what to do first and how much to add to your minimal working system.

2. Definitions

Students enrolled in participating classes will be given a *budget* expressed in terms of 20-minute tutoring sessions, which must last them for the entire semester (*n.b.*, while each class may have a different budget, every student in a class gets the same allocation). They will use your system you are designing to schedule their sessions with an available tutor. They may elect to schedule their sessions early in the semester, or they may elect to wait until they feel the sessions will be most useful. Once they have used up their allocated budget, they can no longer schedule additional sessions.

Your system must thus support several types of agents: *students*, *tutors*, and *faculty*. In addition, there should be an *administrator* (or *root* account) to manage the system. Each actor will have a different view of the system and should be able to complete different tasks.

2.1. Administrator

The administrator is responsible for setting up the system at the beginning of the term. They are responsible for hiring tutors, creating accounts for each tutor and also for each participating faculty member.

2.2. Faculty

Each faculty member is responsible for creating and managing their course's student accounts. Ideally, a faculty member should be able to upload a list of student account identifiers (*e.g.*, a *hawkid*) to establish the initial accounts. The system should generate random account passwords and email those passwords to the student's UIowa email address (note that each hawkid has a corresponding *hawkid@iowa.uiowa.edu* email address). Note that faculty should also have the ability to later add/delete individual accounts for students who add/drop the class.

Faculty are also responsible for designing and uploading sample problems suitable for a tutoring session. For example, as the CS1210 instructor, I might upload a *library* of supplemental programming problems similar to those covered in discussion section or even problems from old exams. This problem library then

[1] Note that tutors are not permitted to help with course homework assignments; rather, the tutoring session is to complement and supplement course material. Students should continue to see their instructors and/or TAs for help with course work.

forms the basis for a tutoring session; in this fashion, tutors can study and prepare sample solutions prior to working with an individual student.

2.3. Tutors

Each tutor is hired to provide a specific number of tutoring hours per week; individual tutors may be authorized to tutor for either all or just a subset of the participating courses. Once their account is established by the administrator (presumably in a similar fashion to how a faculty member establishes a student account), tutors will access a calendar-like schedule to establish their open tutoring slots for the rest of the term (each slot is a 20 minute block). The system should ensure that they not exceed their allocated weekly work hours; moreover, tutors should be allowed to make schedule changes, adding and deleting (open) slots, as the term progresses. In addition to their schedule, tutors should have access to their respective assigned classes' problem libraries.

2.4. Students

Once logged in, each student should (*i*) be able to see the problem library for their respective class, (*ii*) be able to see how many help sessions remain in their individual budgets, (*iii*) be able to access the calendar showing open tutoring slots (identified by tutor name), (*iv*) reserve an open slot, and (*v*) cancel a previous reservation at least 48 hours in advance.

3. Additional Features

Once you have implemented basic system functions, there are many additional features you may choose to add. What follows are simply some examples; you should feel free to propose your own extensions.

(1) You might elect to add icalendar or Google calendar functionality by generating appropriate email messages to students and tutors for scheduled or canceled appointments.

(2) You might give faculty a dashboard showing their allocated budget, appointments completed, appointments scheduled, and open slots remaining. Such a dashboard might also permit faculty to rebudget unused slots by allocating additional slots to students. In this way, a faculty member might elect to only allocate half their budget before the midterm and allocate the remaining budget after the midterm.

(3) You might add bulletin board like discussion capabilities to each problem in the problem library. In this fashion, students could use the library as supplemental study material even without scheduling a tutoring appointment.