Requirements Specification for Testing Center Scheduler

Diana Voss (DoIT) and Scott D. Stoller (Computer Science Dept)

version: 2015-08-31

1. Overview

A testing center was created in Frey Hall during its recent renovation. The University has been looking for software to schedule use of the testing center. Currently, all such software is either proprietary or not designed for a University testing center. This requirements document is based on the University's actual needs. Ideally, DoIT will obtain a deployable system from this course project (aside from some system integration issues).

The system serves two main purposes:

1. An instructor can submit a request to have their students take an online exam in the Testing Center. Instructors specify the duration of the exam and a time period during which students may take the exam.

2. A student on the roster for a course can schedule an appointment to take an exam for the course at any time during the time period specified by the instructor.

Note that the testing center (unlike a classroom) is not used by only one class at a time. Different students may concurrently take different exams for different classes.

2. Assumptions

Each course has a unique identifier. The system does not constrain the exact format of course identifiers, beyond assuming that they are short strings. At Stony Brook, course identifiers are 6-character strings, e.g., CSE308.

Each class has a unique identifier consisting of the course identifier, the section number, an underscore, and the term code. The system does not constrain the exact format of term codes, beyond assuming that they are short strings. At Stony Brook, term identifiers are 4-digit numbers; for example, Fall 2015 is 1158, Winter 2016 is 1161, Spring 2016 is 1164, Summer 2016 is 1166, and Fall 2016 is 1168. For example, the class identifier for CSE308 section 01 in Fall 2015 is CSE30801_1158.

Each exam has a unique identifier consisting of the class identifier, "_ex", and the exam number. For example, the exam identifier for the second exam in CSE30801_1158 is CSE30801_1158_ex2.

For each term, the testing center has specified hours, which specify a range of hours that the test center is open on each day of the week. At Stony Brook, the testing center's hours in Fall

2015 are Monday-Friday, 8am-10pm; Saturday and Sunday, closed. As exceptions to the standard hours, the testing center is closed during specified date ranges (e.g., Thanksgiving recess).

All appointment timeslots start on the hour or half-hour.

All seats in the testing center are equivalent. In other words, any exam can be taken at any seat. Seats are numbered (1,2, …), such that adjacent seats have consecutive numbers. This implies that seats with non-consecutive numbers are not adjacent. Note that seats with consecutive numbers are not necessarily adjacent, e.g., the last seat in one row and the first seat in the next row, but we do not model the room layout in sufficient detail to take this into account.

In the production version of this system, user account management and authentication will be handled primarily through integration with the University's Central Authentication Service (CAS). DoIT will handle that. For this project, we assume the system has access to a pre-existing user account database that contains each user's NetID, first name, last name, email address, hashed password, and roles (administrator, instructor, student). Note that a user may have multiple roles. Code that accesses the user account database should be encapsulated in one component, to facilitate replacing it with CAS client code.

3. Functionality for Administrators

1. Edit testing center information. This includes the:
    a. number of seats in the testing center. Stony Brook's testing center currently has 64 seats.
    b. number of set-aside seats. A set-aside seat is a seat that cannot be reserved directly by students. It can be reserved only by an administrator, making an appointment on behalf of a student. Set-aside seats give administrators flexibility to deal with situations such as malfunctioning computers and students late for appointments due to emergencies.
    c. hours of the testing center for any future term.
    d. date ranges that the testing center is closed during the current term or a future term.
    e. time periods during the current or a future term when the testing center is reserved for a non-Stony Brook exam (e.g., MCAT). The entire testing center is reserved during these time periods.
    f. gap time. The gap time is a lower bound on how long a seat must be unassigned between appointments. The gap time may range from 0 to 30 minutes. For example, if a student has a 9am appointment for an 80-minute exam, and the gap time is 10 minutes, then another student may be assigned the same seat for a 9:30am appointment. If the gap time is 30 minutes, then the seat cannot be assigned to another student until the 10am timeslot (recall that all timeslots start on the hour or half-hour). Changes to the gap time affect subsequent appointment requests but do not affect existing appointments.

g. reminder interval. For each appointment to take an exam, the system sends the student a reminder by email. The reminder interval specifies the number of minutes before the appointment when the reminder message is sent.

2. Import data. The user specifies a term and the name of a folder in the server's filesystem that contains files with course registration data for that term. The system discards existing data for that term and imports new data for that term from those files. The files should be named user.csv, class.csv, and roster.csv. They are in comma separated value (csv) format. Each line in user.csv has the format: FirstName, LastName, NetID, Email. Each line in class.csv has the format: ClassID, Subject, CatalogNumber, Section. Each line in roster.csv has the format: NetID, ClassID. ClassID is a unique identifier for a section of a course in the context of a specified term (e.g., the same ClassID may refer to CSE219.01 in one term and to CSE308.02 in another term). Subject and CatalogNumber together form the course identifier; for example, for CSE 308, the Subject is "CSE", and the Catalog Number is "308".

   Superfluous Appointments. If a student has an appointment for a course exam, and does not appear on the updated roster for the course, then the system marks the appointment as superfluous, releases the assigned seat, and sends email notifications of this to the student, instructor, and administrator. Conversely, if the database contains a superfluous appointment for a course exam, and the associated student appears on the updated roster for the course, then the system reinstates the appointment if possible (i.e., if a seat is available), and in any case sends email notifications to the student, instructor, and administrator. The main motivation for the latter requirement is to easily undo the damage if a bundled data import incorrectly causes appointments to be marked as superfluous.

3. Display the utilization of the testing center for each day in any specified date range. For a past or current day *d*, the system shows the actual utilization, defined by

   $$Utiliz_{actual}(d) \;=\; \Big(\sum_{a\,\in Appts(d)} duration(a)\Big) \;/\; (N_{seat} \,*\, OpenTime(d))$$

   where $Appts(d)$ is the set of appointments on day d, $duration(a)$ is the duration of appointment a including gap time after the appointment rounded up to the beginning of the next timeslot, N_{seat} is the number of seats in the testing center, and $OpenTime(d)$ is the duration that the testing center is open on day d. For a future day d, the system shows the expected utilization, defined by

   $$Utiliz_{exp}(d) \;=\; Utiliz_{actual}(d) + \sum_{e\,\in Exams(d)} (duration(e) + gap) \;*\; (N_{student}(e) - N_{appt}(e)) \,/\, N_{day}(e)$$

   where $Exams(d)$ is the set of exams whose date range includes d, $duration(e)$ is the duration of exam e, gap is the gap time, $N_{student}(e)$ is the number of students who need to take exam e, $N_{appt}(e)$ is the number of existing appointments to take exam e, and $N_{day}(e)$ is the number of days in the date range for exam *e.* For simplicity, this definition ignores the start and end times of the exam.

4. Approve or deny an exam scheduling request for the current or a future term.

   Schedulability. A request can be approved only if it is schedulable. A request is schedulable if, after it is added to the schedule, the testing center has enough seats for all students to take all of their scheduled exams during the specified time periods, taking into account both existing appointments and necessary future appointments. Note that schedulability varies over time, even if nothing else changes. For example, suppose an instructor schedules exam1 over a 3-day period, 200 students needs to take it, and at most 100 students can take it per day, based on the number of seats, the testing center hours, and the exam duration. Suppose no other exams are scheduled. Suppose another instructor wants to schedule exam2 on the last day of that 3-day period, 100 students need to take it, and at most 100 students can take it per day. If the request for exam2 is submitted before the 3-day period starts and before any students make appointments for exam1, then the request is schedulable, because it is possible for 200 students to take exam1 on the first 2 days of the 3-day period, and 100 students to take exam2 on the last day of that period. However, if the request is submitted partway into the 3-day period, and one or more timeslots have gone unused, then the request is not schedulable.

Utilization. This schedulability test does not ensure any scheduling flexibility for students: the last student to make an appointment may have only one timeslot to choose from. To help the administrator decide whether students will have sufficient scheduling flexibility, the system displays, for each day in the date range of the request, the utilization of the testing center based on existing reservations and the resulting utilization if the current request is approved.

5. Make an appointment on behalf of a student in the current term. The administrator indicates whether the seat being reserved is a set-aside seat, being used to deal with an exceptional situation, or a regular seat, being reserved on behalf of a walk-in, i.e., a student who happens to comes to the test center to take an exam without an appointment. [2015-08-31: added the following.] The system performs the same validity checks as when a student makes an appointment, except that, if a set-aside seat is being reserved, the availability test (item e) is modified to check whether a set-aside seat is available.
6. View appointments. The system displays all appointments and the number of available seats at the current time or a specified other time.
7. Cancel or modify any appointment in the system.
8. Check-in a student for an appointment. The system records that the student kept the appointment and displays the student's seat assignment. (Ideally, students would check in by swiping their ID card, but that is beyond the scope of this course project.)
9. Generate reports. For each report, a specified term can be a past, current, or future term. For web-based systems, each report can simply be displayed in the browser, and the browser's Save or Print function can be used to save it to a file. Non-web-based systems should also provide the ability to save a report to a file. Plain text format is acceptable, although better formatting (e.g., HTML) is preferable.
   a. For each day in a specified term, report the number of student appointments on that day.
   b. For each week in a specified term, report the number of student appointments that week and the course identifiers of courses associated with any of those appointments.
   c. For a specified term, report the courses that used the testing center in that term.
   d. For a specified range of terms, report the total number of student appointments in each term.

4. Functionality for Instructors

1. Request to schedule an exam in the testing center in the current or a future term.
   a. If the exam is for a course, the instructor provides the course identifier, section number, term, duration of the exam; the start date and start time, and the end date and end time For example, if the start date and time are August 25 at noon, and the end date and time are Aug 28 at 5pm, then all appointments to take that exam must be completely within that time period.
   b. If the exam is ad hoc (i.e., not for a course), for example a placement exam or language proficiency exam, the instructor provides the name of the exam, the duration, a list of students taking the exam, the start date and start time, and the end date and end time. The list of students is entered into a text box. Each line should have the format: NetID, LastName, FirstName.

   The system rejects un-schedulable requests. If the request is schedulable, the system displays, for each day in the date range of the request, the utilization of the testing center based on existing reservations and the resulting utilization if the current request and all pending requests to schedule exams are approved. The instructor can then choose whether to cancel or submit the request.
2. See a list of his or her exam scheduling requests in the current and future terms. The system should display some details of each request, including the current status (pending, approved, or denied) and, for approved requests, how many students should take the exam, and how many students have taken the exam.
3. Cancel a pending exam scheduling request.
4. See appointment and attendance details for a specified exam, including each student's appointment time (if any), assigned seat, and whether the student showed up for the appointment.

5. Functionality for Students

1. Make an appointment to take an exam. The system ensures:
   a. the student is enrolled in the course in the current term or is on the list of students who need to take the ad hoc exam.
   b. the student does not have an existing appointment for the same exam.
   c. the student does not have an appointment for a different exam in an overlapping timeslot.
   d. the appointment is entirely between the start date-time and end date-time of the exam.
   e. a non-set-aside seat is available at the requested time, taking into account the gap time between appointments.

   If these conditions are satisfied, the system assigns the student an available seat, trying to avoid placing students taking the same exam in adjacent seats. For simplicity, the system just tries to avoid placing students taking the same exam in consecutively numbered seats, ignoring the fact that some consecutively numbered seats are not adjacent.

2. Cancel an appointment. A student may cancel an appointment any time prior to 24 hours before the appointment, but not after that.
3. See a list of his or her appointments in a specified past, current, or future term.
4. Appointment reminder. For each appointment to take an exam, the system sends the student a reminder by email. The reminder interval specifies the number of minutes before the appointment when the reminder message is sent. If the appointment is made less than the reminder interval before the appointment time, no reminder is sent.

6. Other Requirements

Authentication: All access to the system requires authentication with a password.

Network Security: Communication is secured using HTTPS/SSL.

Application Security: The application should have reasonable protection against common vulnerabilities such as listed in the OWASP Top 10 **(**https://www.owasp.org/index.php/Top10). The application should not be vulnerable to URL replay attacks. The demo must include a test case that involves a URL replay attack. For example, an administrator logs in, navigates to the “View appointments” page, copies the URL, pastes it in another browser, and tries to view appointments in the second browser.

FERPA compliance. The system should be FERPA compliant. This implies that students can see only their own information. They cannot see any information about other students.

Concurrency: [2015-08-31: modified the following sentence to include all kinds of users.] Multiple users on different computers may use the system concurrently. Synchronization is used to ensure sensible behavior. The system may prevent the same user from having multiple concurrent login sessions; if it does not prevent this, it should ensure sensible behavior.

Client-Server Architecture: Clients may run concurrently on different hosts. All clients access the same database.

Programming language and Web development framework. The system may be implemented using any programming language and Web development framework. Stony Brook DoIT’s production environment uses Microsoft’s ASP.NET framework. Teams are encouraged to use that framework, if they are interested in seeing their system go into production use.

User friendliness. The system should have reasonable response time, display confirmation messages when requests have been processed successfully, and display informative error messages when a request cannot be completed or an error occurs.

8. Testing

Some test cases will require artificial manipulation of the system’s notion of the current date-time. For example, consider the scenario described in the paragraph about Schedulability in the description of the “Request to schedule an exam” function for instructors. Testing both variants of the scenario requires setting the current date-time to be before the 3-day period and setting it to be during the 3-day period. To facilitate this, the system should define its own function that returns the current date-time. This function has two modes: normal and test. In normal mode, it calls the execution platform’s current date-time function. In test mode, it returns a constant value provided by the user. The system should provide an interactive command for administrators that switches between modes and sets the value returned in test mode.