

Developing a Computer Lab booking system for Commack High School

Anthony Bisulco

Table of Contents

ANALYSIS.....	3
DESIGN	5

My topic is: Developing a system to allow users to book computers labs

My end user is: Mr. Keltos/ Commack High School

Analysis

Description of the project: Currently in Commack High School, there are six computer labs in which teachers can book their class to go to for class work. In order for a teacher to get a spot in the computer lab, he or she must go to the computer lab he or she would want and then would have to sign up a period and date to be in there. The system, though, is time consuming because if a spot is not available for the teacher, he or she must go to another computer lab to look at the book and try and get a spot there. The whole process winds up with a teacher walking all around the school to book a computer lab.

The goal of this investigation is to solve this problem above by making a software application where a user can book a computer lab online. This would solve the above problem in which a teacher would not have to run around the school, but rather he or she could just go online and see what labs were available and book the lab they want. One requirement that a user has is a teacher needs to be able to log in to a form, external to the program or internally based on the circumstances, and put in a set of credentials in order to prevent student use. A teacher also must be able to book any of the six computer labs in the school on a specified date and period. If a teacher does not want that date, he or she must be able to delete it in order for other teachers to have access to the computer lab that period. Another requirement that teachers need is to be able to find what dates that they have booked. For example, if a teacher booked computer lab 1 on October 12 2012, they must be able to search for all the dates they have reserved. Finally, a teacher must also be able to search a specified date for any computer lab availability.

One different set of requirements than the teachers is the admin. The admin must be able to remove a teacher from the computer lab for any reason possible. Also the admin must be able to auto book dates. For example, if there is an assembly period 1-3, he or she must be able to

Id	Requirements	Description	Optional
1	User Input	User must be able to input period, date and time that they will be using the computer lab	N
		Appropriate security measures will be taken if they cannot be handled externally from the program	N
2	GUI	A visual implementation of the programs features making it more accessible to users	Y
3	Find Feature	Users must be able to find what labs they have already booked	N
4	Multiple computer labs	User must be able to book and view the usage of the six computer labs in the school	N
5	Web interface	User must be able to access this applet from the internet	Y
6	Administrative privileges	Admin must be able to remove teachers from computer labs	N
		Admin must also be able to view what periods a teacher is booked in a computer lab Admin will also have the ability to schedule computer labs to prevent teacher from booking possibly during a school event	N 3

book those three periods in order to not allow teachers to book them. Also the admin must be able to search and see what periods a teacher is using the computer lab. For example, a search for Mr. Smith would tell the admin period 1, 2 and 3 on October 12 2012 in computer lab 2. All together the admin will have supreme authority over the program and can edit anything if necessary.

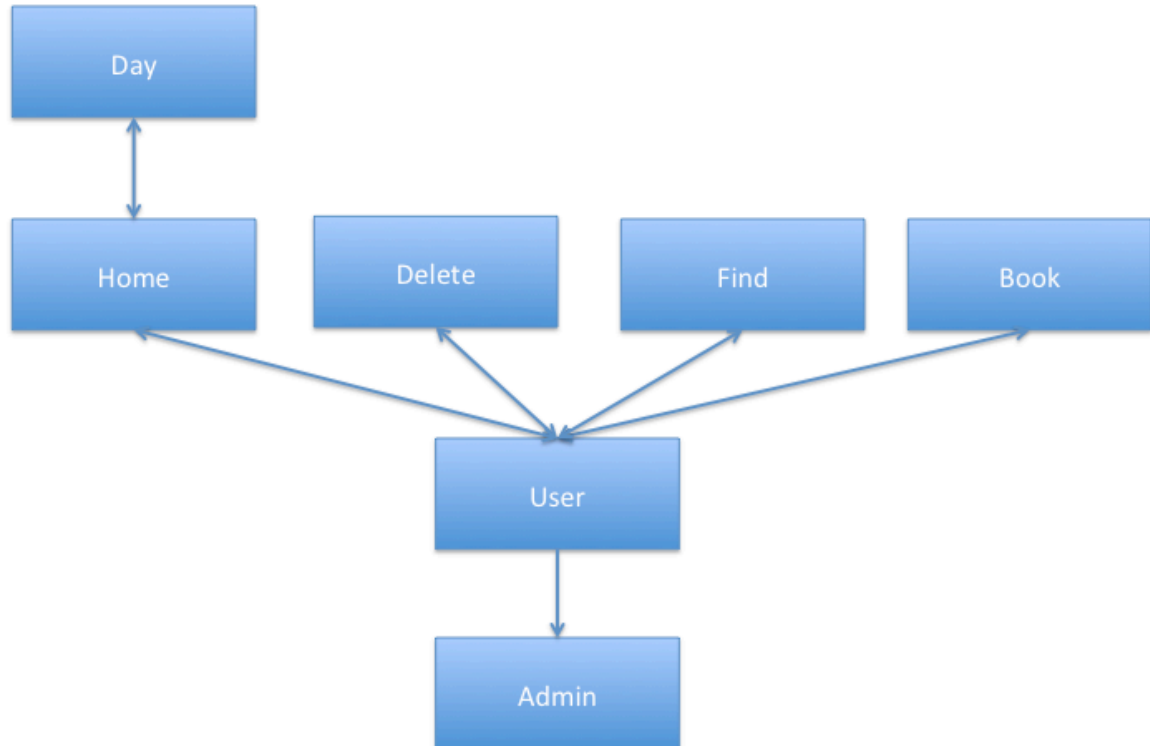
Resources required to complete project:

- School calendar
- Intranet access
- Access to school file system

Project Deliverables: At the end of the programming of this project, user documentation, training sessions and program implementation will be provided. The user documentation will be a screen shot tutorial of how to use the application and a text instruction on how the application works and how to use it. As for training sessions, Mr. Keltos will be taught how to use the program along with any other teacher asking for specific help on how to use the program.

Project Success Criteria: My user and I will consider this project a success if teachers can view computer lab usage online, book a computer labs online and teachers can log onto the program.

Design



Class Breakdown

The overall design of a computer lab booking system for Commack High School is broken down into two classes. There is a user and an administrator, each with a different set of properties. A user overall has five functions. They can book, delete, find what day they are in a lab, log off and browse through a list of days in order to find a day. An administrator overall has all these same method except an administrator can override any booking that a teacher books. Overall one can see that an administrator is a specific kind of user in which an administrator will extend from a user.

User Class Modular Design

The user has five functions which are the methods for the user class. When a user first opens the program they will come to a menu which will allow them to choose from one of these five functions in order to do their specified task. Book, which will book a day in the computer lab for the teacher, will work by storing that booking for that teacher so no one else can take it. Booking will consist of two fields pertaining to the booking in which one is the period of booking and the other is the date of booking. Once a teacher has filled out these fields they can hit the book button in which a label at the bottom of the screen will either say booked or not available based on if this spot is taken.

User Class Algorithm

In order to see if this spot is taken the program will go to a three dimensional array which will have computer lab, period and date as its three dimensions and will check based on the parameters that were obtained in the field, what computer lab tab one is under and who is logged in if that spot in the array is filled. If it is not filled the teachers name and department will be

filled in and booked will appear on the label. Though if the spot is filled then the label will read back not available and not store any data in the array. Once a user had completed their objective they can then hit the back button in order to be moved back to the main screen where they can also delete, find, log off and add additional bookings.

Delete Class Modular Design

The next function a user has is delete, in which a teacher must be able to give up his or her computer lab time for other teachers if he or she cannot use that time. A user would first navigate to the delete button on the menu, which will bring them to a list of all their current bookings with period number, date and computer lab number. A user then comes to the screen and will type in the number they would like to delete. If a user would like to delete multiple numbers, they will complete this process for each number they would like to delete. Once finished deleting one entry from the list of found bookings the program will refresh and display a new list of found bookings. The user once done with all their deletions will then press the back button to go back to the menu.

Delete Class Algorithm

This found bookings list is found by looping through the entire three dimensional array and finding where that teacher's name comes up in the array slots. When the teachers name is found, a new array with all the teachers' bookings is created and position in the array. This array is then displayed on the screen numbered sequentially based on the number found. The program once it has obtained the number the user would like to delete will go back to the array with all the bookings and obtain the position element from the booking array by that teacher. With this number the program will then go to the three dimensional array with that position element and delete the content inside the element but not the element itself.

Find Modular Design

Another function a user can do is to find all their bookings they have made, which is done by navigating to the find button on the menu. Once a computer is done looking through their bookings they then hit the back button where they go back to the menu to accomplish their next task.

Find Algorithm

The same a algorithm to find the bookings of a teacher in the delete method was used.

Browsing Functionality in Home Class

A user may want to also try and browse through the computer lab days and find if within a day a certain computer lab is open. This is accomplished right on the menu a list of the current listing of all the teachers that are booked for today is listed with the ability to change the day to find other days with open periods. If a teacher would like to look for a certain day they will type in MM/DD/YY and press enter, which will bring up a listing of all the spots for that day in the computer lab which tab you are under. A user can then repeat the process as many times as they would like and then go on to their next task.

Browsing Functionality Algorithm

This list is accomplished by going to the three dimensional array and looking through the current date typed in and the computer lab tab one is under with the period number one through nine looping through storing data in a new array with the contents of the three dimensional array. This new array is then displayed on the screen numbered sequential and if there is no teacher booked for that period a blank space appears after that number.

Log Out functionality in Home class

When a user is done with all their specified tasks they need to do, he or she can go to the log out button. The log out button will close the program and will log the teacher back to the log in interface. If they would like to log back on, he or she will go to the log in interface and log back on which will open the program to the user. This log in interface will be handled externally to the program which will protect the program from student use. Overall, a teacher now has their computer lab date and now can use their computer lab time if they successfully booked a computer lab.

Day Class Algorithm

Another algorithmic process used in this program was that in order to find out if its an even or odd day. The Commack High School keeps an up to date calendar on their website that contains whether it is an even or odd day so I used this information in order to determine whether it was an even or odd day. If this information was not obtained this way rather by just saying every other day is odd or even this method would be susceptible to error due to closing of school or other event. So in order to first find whether its an odd or even day we have to create an http client in order to connect to the calendar and get the webpage's text which contains whether its an odd or even day. The http client request the url [http://www.commackcalendar.org/calendar_events.cfm?ListEvents=1&printpage=0&&cat=6633.2994,2748,2991&location=1081&dir=&themonth="+month+"&theyear="+year+"&buildit=0.961211055903&AddSportingEvents=0](http://www.commackcalendar.org/calendar_events.cfm?ListEvents=1&printpage=0&&cat=6633.2994,2748,2991&location=1081&dir=&themonth=) where there is a interchangeable month and year inserted in order to get the different calendar for different months and years. After the client has obtained the html file a buffered readers reads the html files tags into a string. Now this string must be converted from html to the raw text on the webpage which is done by using a package known as Jsoup. Jsoup allows one to parse an html file into raw text. So using jsoup we then get the raw text file by using its method Jsoup.parse(html).text(). After this we can trim of the header on each of the calendars which contains information like school number and location. This is done by taking the substring from 442 on of the parsed html page. After this the day is checked if it's a summer date where there is no school where the immediate output of the method is no school. Next if the above condition is not true the method continues by checking if the text no school is seen in the parsed html file by checking if the index of the date plus the text SCHOOL CLOSED is seen. Though this is then rechecked in order to see if the string also contains SCHOOL CLOSED K-5. If this is true then the method continued unless if school closed is just true where the immediate output is school closed. Finally if there is no school the method searches though the parsed html file in order to find the date searched for. When this is found then the closest occurrence of CHS/CMS Day DAY is found. Since this is the closest occurrence this will work for weekdays but not for weekends where the day would be skewed in order to fix this problem the data is then rechecked for day of the week. Then based off of a fixed value known between the words CHS/CMS Day DAY and the day number the substring is taken of the parsed html file containing the day number. Then in order to check day of the week in order to confirm the day is a weekday a date object is created with day of the week. We then can use the method getday which returns a number 0-6 corresponding to the day of the week. This number is then checked to be a 0 or 6(Saturday or Sunday) if this is true nothing is returned. Otherwise no conditions have been flagged so the correct day number is returned.

Administrative Overrides

An administrator gets all the same methods as a user except that he or she can override specific actions. For example, if an administrator knows there isn't any assembly 3rd period for all grades, he can then book a slot in the computer lab for third period. If a teacher already has

time booked that period, the administrator then has the ability to book over that teacher. This is one change to the book method which will now go to the period booked in the three dimensional array and erase the contents and replace it with administrator. Another action an administrator can do is to delete a teacher from a computer lab. This is a change to the delete method, in which instead of a list of found booking three fields come up in order for the admin to delete a teacher. The admin then types in the first field period, the second field date and the third field computer lab number. The program then goes to this specific number in the array and deletes the contents of that element. The final change to the user class that an admin has is instead on find dates booked of the user he can find when certain teachers are booked. This is a change to the find class by using one field in order to obtain teacher name searched. Once this teachers name is obtained it is searched for throughout the three dimensional array and if the name is found anywhere it is stored in a new array with date of booking, period number and computer lab. This array is then displayed to the administrator where he can do his analysis. Overall there are only three changes from a user to an administrator which allows for the program to function smoothly.

File Handler Algorithm in User Class

The data stored in the run time array of bookings is written using a file writer by looping through each element of the array. These elements then are stored in the text file to be read by the file reader. The file reader reads the text file by looping through each line and sets it equal to the corresponding point in the array.

GUI Building

The netbean's GUI builder was used in order to create the GUI for the book, delete, find and home class. The GUI builder was used to draw the screens and add the components as for using it for coding aside from the drawing of the screen that code was written by myself.

Data Structures

	1	2	3	4	5	6	Cl
1							
2			Holmes Math				
3						Dineen English	
4							
5							
6							
7				Solomon Science			
8							
9		Saturner Social Studies					
Period							

The data of the computer labs bookings is stored during run time in a 3 dimensional array while not in run time it is stored in a text file. The 3 dimensional array is made up of one dimension as periods which has 9 elements for the 9 different periods. Along with periods as a dimension another dimension is computer lab which has 7 elements for the 7 different computer labs.. Also the last dimension of the array is date which is 364 elements corresponding closely to days in a year. Now based on date, period and computer lab number the teachers name is then stored into that array element. The date element of the booking is actually converted to days over the year in order to be able to store the date as a number(181) rather than 12/12/23. After the program is finished running a text file is written by looping through the array with the current values of the array elements. During run time the program will then read these values into the array by looping through the contents of the array.

Justification for the choice of data type

Overall, in order to store the period, date, and computer lab number for a Computer Lab booking system for Commack High School, I used a three dimensional array. This will allow for an ease of storage because an array is a singular unit as compared to if this program was to use object storage in which there would be 200 different objects each with its own information. Compared to an object storage arrays overall contain the 200 different objects in an easy interface accessible to all users like a database. Instead of storing data in our own object memory bank we store it to a shared inquiry database with everyone's information. The use of an array is though sometimes not ideal for this situation. For example, if I need to find a teachers period in computer lab I have to loop through the array and find the teacher while if object storage was to be used I could call a method like get All Periods Available, though overall arrays allow for an easy change to the program. Take for example if you wanted to browse through the periods available if you were to use object storage each user would have to contact an object and get its data for that period. As for an array one could just loop through the array and find the periods, dates and periods available. Overall any array allows for a clear concise shared inquiry of all users of the data.

Documentation

Handling Errors

In this project errors such as File not found, IO exception and Number format exception were caught from each input and output of the program. When the program starts a file is read in with the bookings stored locally so when this file cannot be read an error is caught and a error message pops up. This is an IO exception because the file cannot be read. Another error that can occur when reading in the file is the file is not found which would occur if the file name used to find the file was not located. This error is caught using a try and a catch loop in order to allow the function to still run. Along with errors within file IO this project also dealt with error when inputting data into text fields. In order to prevent the program from using incorrect data such as putting in the wrong date a Number format exception is used to catch any error in putting in any strings. If a string is put into a text field then a error message will pop up. Along with errors in formatting if the dates put in are to large then another error message pops up with an error saying to large or small a number inputted.

Mastery Element Locations:

- Arrays used in User class labeled location 1
- User Defined Object in Home class labeled location 2
- Object as data record in Log In class labeled location 3
- Complex selection in Home class labeled location 4
- Simple selection in Home class labeled location 5
- Loop in Home class labeled location 6
- User defined method with parameters in User class location 7
- User defined method with return value in User class location 8
- User defined in User class location 9
- Search in User class location 10
- File IO in User class location 11
- Use of additional library used in day class location 12

Evaluating Solution

In this program one thing that I would change in order to make the program better is to use object storage rather than storing the data straight into the array. The benefit of storing the data in object is a method such as getClass or getTeacher would allow for the maximum amount of data to be stored rather than just storing the teacher name. With the current data storage system the data was stored as a string saying teacher name who booked that class. Now if an object were to be used many more things could be stored rather than just teacher name.