Room Scheduler Initial Phase - Programming Assignment 6

You have been asked to develop a Room Scheduling application for Tiny College. This college has only one building in which to schedule rooms. The Tiny College can reserve one room per Date per faculty, for each Date the rooms are available. The rooms are reserved by Faculty name. Faculty can request a room reservation for a specific Date based on the number of seats they require. The room will be assigned by the program, faculty do not get to request a specific room. Each Date is just a specific Date. Faculty are identified by a single name.

This application should have a very nice GUI interface and will be a database driven application. The database used will be Derby. This application must use good Object-Oriented Design and Programming. The database must use good Object-Oriented Design and Programming. There is a very close correlation between Object-Oriented Design and Database Design. Your application design should include at least four classes besides the main GUI class, e.g. Faculty class, Room class, Date class..., Reservations class, etc. Your database accesses should be in the classes that correlate with the database tables.

This assignment is the first half of the final project and will be submitted as Programming Assignment 6. This phase of the project will implement the following user commands:

Add Faculty

A faculty member is added to the database. The faculty member is identified by only one name.

Reserve Faculty Date Seats

The faculty member will be assigned a room for the requested Date, if there are seats available. The rooms will be assigned in a best fit manner. The faculty member should be assigned the smallest room that has enough seats for their request. If there are no rooms available or no rooms with enough seats available, the faculty member will be put on the wait list for that Date. The waiting list must be maintained in the order the faculty members tried to reserve their rooms.

Status Reservations by Date

The Status command by Date will display the faculty members that have rooms reserved on that Date.

Status Waiting List

The Status command for the Waitlist will display all the Faculty members waiting for rooms. It will be displayed in Date order and then in the order of when the reservation was requested.

Testing scenario:

A testing scenario will be provided to assist you in testing this application.

Database considerations:

The Rooms Table should be preloaded with several Rooms such as 101, 102 and the number of seats in the room. You should have a different number of seats per room so that you can test the proper reservation of rooms for different class sizes. You will be shown how to preload tables with values.

The Date Table should be preloaded with several Dates of your choice. These are just normal Dates.

The database tables should not contain redundant data, i.e. relevant data should only appear in one table.

GUI Guidelines:

The user should be required to enter only unknown data. Drop down lists of known data such as Faculty member names, Room names, or Dates should be displayed for the user to select. Combo Boxes should be used for the drop-down lists on the form. When information is requested to be displayed e.g. for a Status command, all of the requested information must be displayed. When a command is performed, the results of that command should be displayed to the user on the same display without the user needing to check Status to see what was done.

For this assignment, you may, and I recommend, the use of the GUI designer in NetBeans.

Submission Guidelines:

Don't forget to submit your PROJECT files and your DATABASE files. The database folder is probably located under "Users/USERNAME/.netbeans-derby/NAME_OF_DATABASE".
Zip the ENTIRE database folder and the ENTIRE project folder and submit the two zipped files in the dropbox under one submission.

Note:
Your project must be created with the name 'RoomSchedulerNameID', where Name is your name and ID is your psu account id, e.g. xxx1234. The database must be created with the name 'RoomSchedulerDBNameID', and a username and password

of java and java. The database when submitted for PA 6, should contain data for the Rooms and Dates tables. All other tables should be empty.

Grading Criteria:

In this project I will be looking for good OO design practices and this includes:

- **Use of getter and setter methods for class variables**
- **Good naming of your classes, methods and variables**
- **Correct use of static and non-static methods**
- **The way you split this project into classes.**
- **All of your updates to the database must be done using SQL statements, do not use ResultSetTableModels to update the database.**
- **If a SQL statement to update the database needs to contain a variable, then you must use PreparedStatements, do not use concatenation of strings to create the SQL statement.**

Note: A good example to use for how to program this assignment is contained in the programming examples in Figures 24.30, 24.31 and 24.32 in Chapter 24 from our book.

Room Scheduler Final Phase

You have been asked to develop a Room Scheduling application for Tiny College. This college has only one building in which to schedule rooms. The Tiny College reserves one room per Date per faculty, for each Date the rooms are available. The rooms are reserved by Faculty name. Faculty can request a room reservation for a specific Date based on the number of seats they require. The room will be assigned by the program, faculty do not get to request a specific room. Each Date is just a specific Date. Faculty are identified by a single name.

This application should have a very nice GUI interface and will be a database driven application. The database used will be Derby. This application must use good Object-Oriented Design and Programming. The database must use good Object-Oriented Design and Programming. There is a very close correlation between Object-Oriented Design and Database Design. Your application design should include at least four classes besides the main GUI class, e.g. Faculty member class, Room class, Date class..., Reservations class etc. Your database accesses should be in the classes that correlate with the database tables.

This assignment is the final half of the project. The final project is a continuation of Programming Assignment 6. This phase of the project will additionally implement the following user commands:

Add Room Seats

Add a new Room to the system. The Room name is a string and Seats is the number of seats in the Room. When a room is added, the wait list must be searched for any faculty waiting for a room for all Dates that rooms are reserved and reserve the room for them and remove them from the waiting list.

Add Date

Add a new Date that rooms can be reserved to the system.

Cancel Reservation Faculty Date

The reservation for that Faculty member and Date must be removed from the Room's reservations or the waiting list. If the reservation is removed from a room, the waiting list must be checked to determine if another Faculty member can be booked for that room for that Date, checking the number of seats requirement, of course.

Status Faculty member

The Status command for a Faculty member will display the Room and Date for each room the Faculty member has reserved and/or is waitlisted for.

Drop Room

The Drop command must remove a room from the application. Any faculty members that had this room reserved for any Date will get another room reserved for that Date if possible, in priority sequence, and the new reservation reported to the user. If the faculty member cannot get a new room reserved, the user is informed that the faculty member could not get a new room reserved and that they have been placed on the waitlist.

Testing scenario:

A testing scenario will be provided to assist you in testing this application.

Database considerations:

The Room and Date Tables no longer need to be preloaded with values. When submitted, all database tables should be empty.

The database tables should not contain redundant data, i.e. relevant data should only appear in one table.

GUI Guidelines:

The user should be required to enter only unknown data. Drop down lists of known data such as Faculty, Rooms or Dates should be displayed for the user to select from. Combo Boxes should be used to categorize data on the form. When information is requested to be displayed e.g. for a Status command, all of the requested information must be displayed. When a command is performed, the results of that command should be displayed to the user on the same display without the user needing to check Status to see what was done.

For this assignment, you may, and I recommend, the use of the GUI designer in NetBeans.

Submission Guidelines:

Don't forget to submit your PROJECT files and your DATABASE files. The database folder is probably located under "Users/USERNAME/.netbeans-derby/NAME_OF_DATABASE".
Zip the ENTIRE database folder and the ENTIRE project folder and submit the two zipped files in the dropbox under one submission.

Note:
Your project must be created with the name 'RoomSchedulerNameID', where Name is your name and ID is your psu account id, e.g. xxx1234. The database must be

created with the name ‘RoomSchedulerDBNameID’, and a username and password of java and java. When your database is submitted, all tables should be empty.

Grading Criteria:

In this project I will be looking for good OO design practices and this includes:

- **Use of getter and setter methods for data**
- **Good naming of your classes, methods and variables**
- **Correct use of static and non-static methods**
- **The way you split this project into classes**
- **All of your updates to the database must be done using SQL statements, do not use ResultSetTableModels to update the database.**
- **If a SQL statement to update the database needs to contain a variable, then you must use PreparedStatements, do not use concatenation of strings to create the SQL statement.**