Magician Agent Initial Phase

Your neighbor has a business representing Magicians. She would like to run her business on her computer and has asked you to develop a program that will allow her to track what Magicians are booked for which Customers for specific Holidays. 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. Magician class, Holiday 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:

Book Customer Holiday
The customer will be assigned a magician for the requested holiday, if one is available. If one is not available, the customer will be put on a wait list for that holiday. The waiting list must be maintained in the order the customers where placed on the list. The magicians can be assigned in any order.

Status Holiday or Magician or Waiting List
The Status command for holiday will display the customers and their respective magicians for the requested holiday. The Status command for magician will display the customers and the holiday for the requested magician. The Status command for waiting list will display the waiting list of customers and the holiday for which they are waiting.

Database considerations:
The Magician Table should be preloaded with several magicians such as Merlin, Houdini, and Gandalf.
The Holiday Table should be preloaded with several holidays such as New Years, Halloween, and 4th of July.
The database tables should not contain redundant data, i.e. relevant data should only appear in one table except for foreign key fields.

GUI Guidelines:
The user should be required to enter only unknown data. Drop down lists of known data such as Magician names or Holiday names 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 without the user needing to check Status to see what was done.

Submission Guidelines:

Don't forget to submit your PROJECT files and your DATABASE files. The database files are probably located under "Users/USERNAME/.netbeansderby/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: When your database is created, create it with a username and password of java and java. The database when submitted for PA 6, should contain names of the magicians and the holidays. 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.**

Magician Agent Final Phase

Your neighbor has a business representing Magicians. She would like to run her business on her computer and has asked you to develop a program that will allow her to track what Magicians are booked for which Customers for specific Holidays. 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. Magician class, Holiday class..., etc. Your database accesses should be in the classes that correlate with the database tables not the GUI class.

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 Magician
When a new magician is added, the waiting list must be searched to see if any waiting customers can be scheduled. Any customers booked must be reported to the user.

Drop Magician
The Drop command must remove a magician from the application. Any customers the magician has booked must be rebooked with another magician if possible. If the customer cannot be rebooked, the customer must be put on the front of the waiting list so they will have the first opportunity to be rebooked.

Cancel Customer Holiday
The booked entry for that Customer and that Holiday must be removed from the magician's bookings or the waiting list. If the booked entry is removed from a Magicians bookings, the waiting list must be checked to determine if another customer can be booked with that magician for that holiday.

Add Holiday
Add a new Holiday to the system.

Database considerations:
The Magician and Holiday 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 except for foreign key fields.

GUI Guidelines:
The user should be required to enter only unknown data. Drop down lists of known data such as Magician names or Holiday names 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 without the user needing to check Status to see what was done.

Submission Guidelines:

Don't forget to submit your PROJECT files and your DATABASE files. The database files are probably located under "Users/USERNAME/.netbeansderby/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: When your database is created, create it with 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.**