# Naming Convention class

At the beginning of our work, we decided that we are going to need some kind of a naming convention interface that would be implemented by most of the classes in the project to prevent bugs, caused by spelling mistakes. Upon further consideration, however, we discovered that the Strings that we would need to store in this interface were mostly going to be used in the top layer of our program (the GUI) and that was a problem, because .jsp files cannot implement interfaces. They can, however, use the import statement to import classes. That is why we agreed that instead of an interface, we should use a plain old class for this purpose.

# Dummy Log-In

Since we were going to have at least two different types of users, using the product, and since developing a log-in system that actually checks for someone's username and password is a rather lengthy process, we decided that, at least until we're done with the more important user stories first, we would need some kind of a dummy log-in system that would easily allow us to use the log-in screen and test the system. The idea I came up with is the following: In order for us to log-in as an admin (dell guy), we would have to type "admin" in the username field and just some random characters in the password field. And in order for us to log-in as a partner of dell, we would just type random characters in both input fields and get in. It works the following way: First, we manually created the two different types of users that we would have and put them in the database using INSERT INTO queries. Then, we wrote the code, so that once the log-in button on the index page is pressed, the information in the input fields goes to the dummyLogIn() in the Servlet. There the method checks if the username parameter is "admin". If so, it starts the chain calling of the also dummy getAdmin() methods which lead all the way down to the dataSource layer. There, the previously mentioned admin user is retrieved and returned. The process for retrieving the partner user is identical, except it relies on the getPartner() methods.