CIS*2430 (Fall 2010) Assignment Four

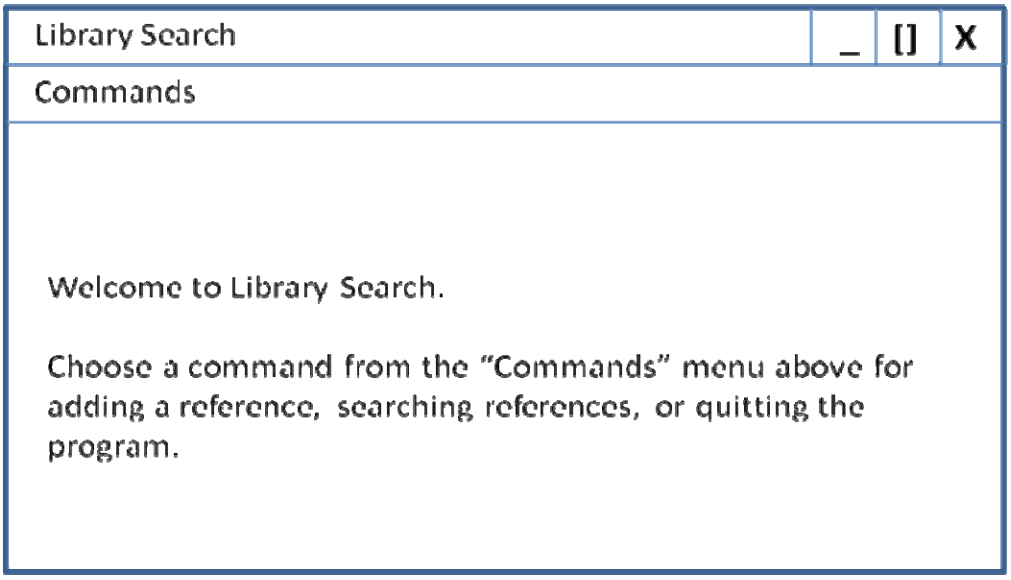
Instructor: F. Song

Due Time: December 6, 2010 by midnight.

In Assignment Three, you improved your design for four classes (Reference, Book, Journal, and LibrarySearch) and were able to read/write references from/to files and maintain a HashMap index to speed up the title search. In Assignment Four, you will add a GUI interface and exception handling so that the system will be more robust and user-friendly. There are also bonus marks for any efforts beyond the basic requirements to make your system even more useful and attractive.

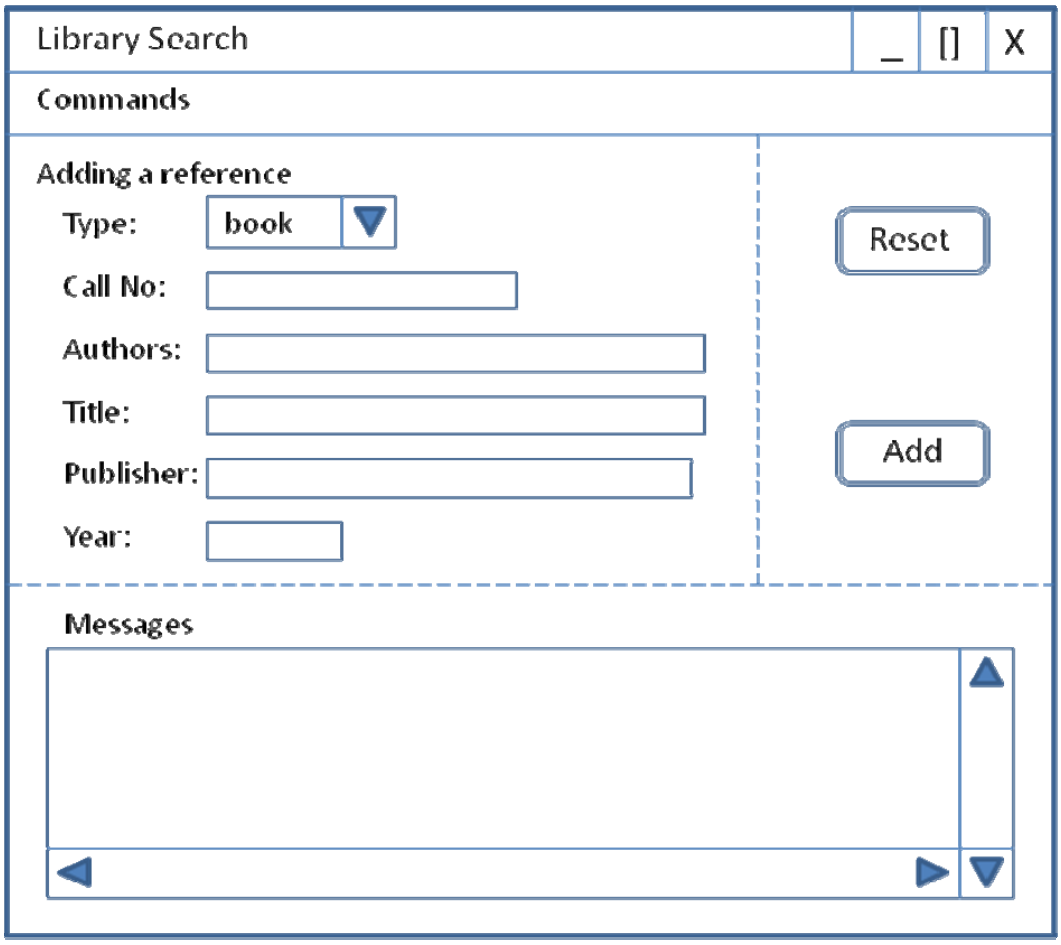
Specific Requirements for Assignment Four

(1) Create an initial interface that looks like the following. The content pane shows a welcome message and a brief instruction of how to use the system, and the "commands" menu has three choices: add, search, and quit. Choosing the "add" command will change the interface to the looks described in requirement (2); choosing the "search" command will change the interface to the look described in requirement (3); and choosing the "quit" command will simply terminate the program. Note that choosing the X button on the window frame will also terminate your application.

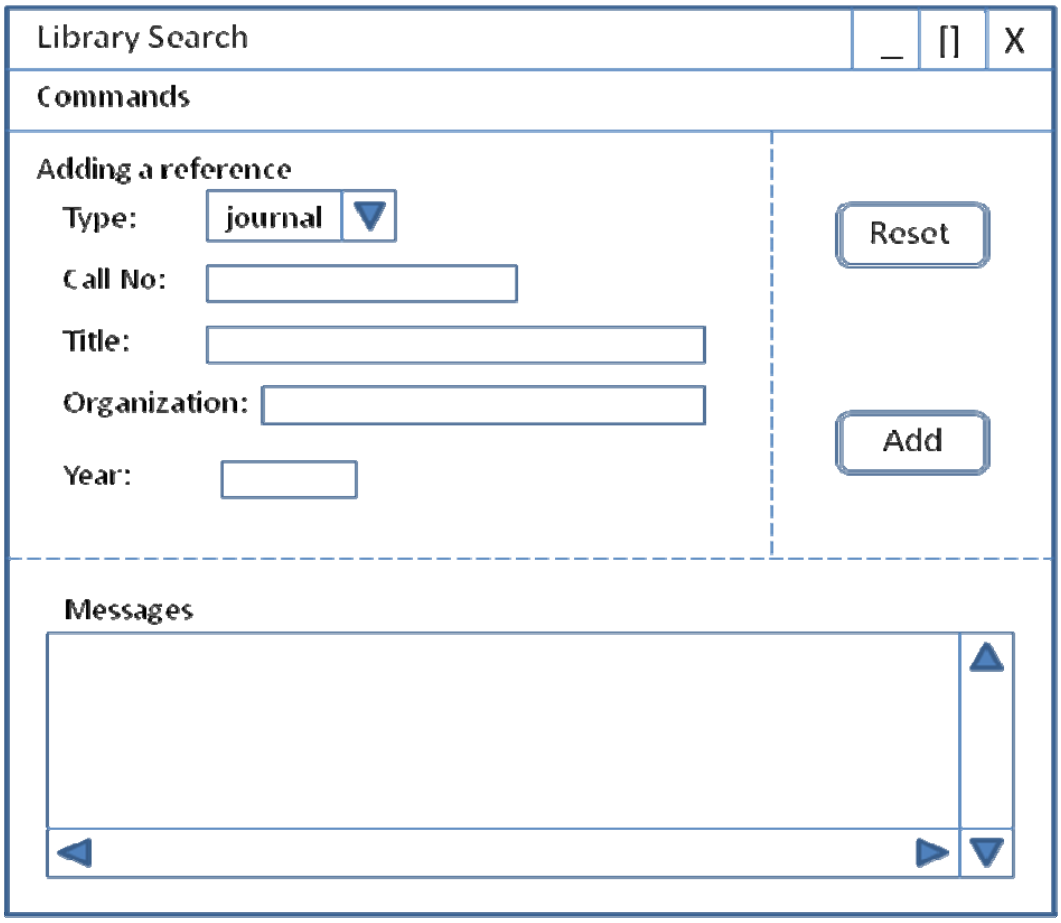


(2) Create the following interface for adding a reference. When the user selects the "add" command, the initial interface will be changed to the following. The type can be selected through a combo-box with two choices: book and journal. The default choice is "book", but the user can change it to another choice in the drop-down list. For the choice of "book", the

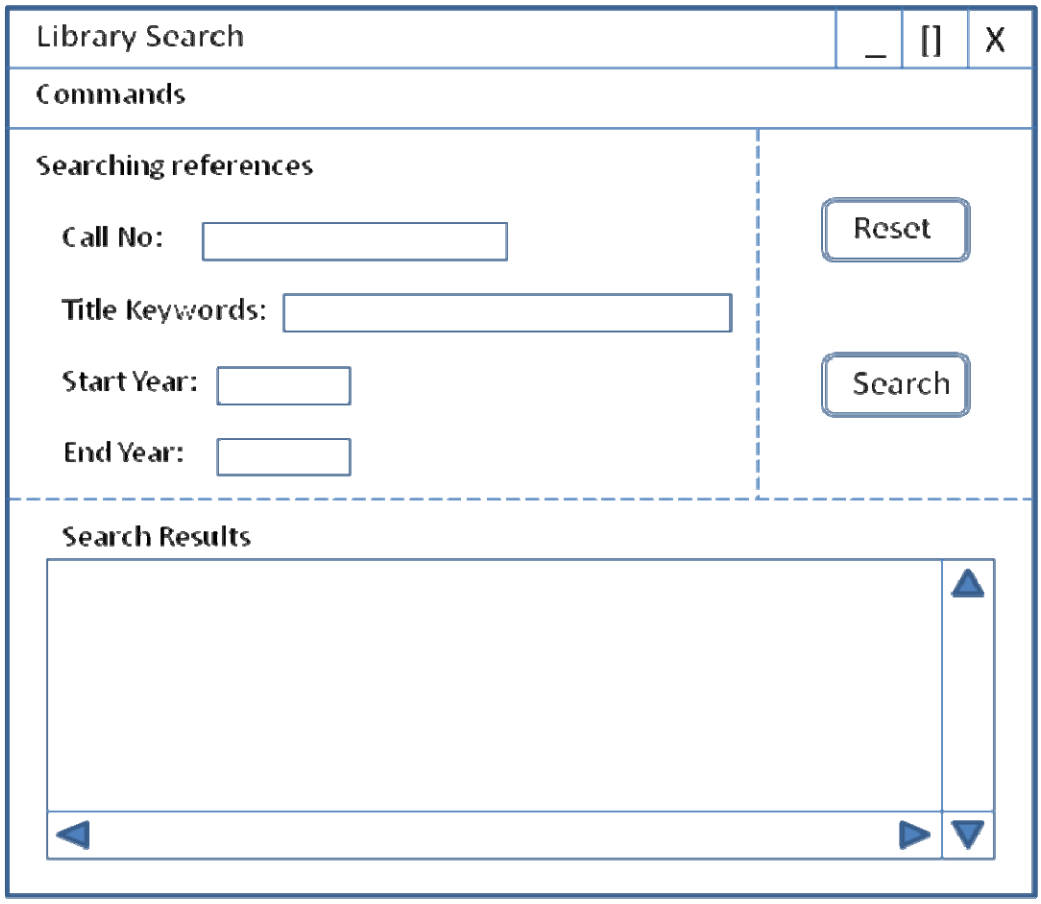
other five attributes: call number, authors, title, publisher, and year all get their values through textfields. There are two buttons on the righthand side: clicking the “reset” button will clear all the textfields and clicking the “add” button will try to create an object for the related reference. However, if there are any violations for the validity of the reference, an error message will be displayed in the text area at the bottom of the interface and the user is required to try again.



For the choice of "journal", the other four fields are call number, title, organization, and year, as shown in the following illustration. In implementing your interface for adding a reference, you can include all these textfields: call number, authors, title, publisher, organization, and year. For the choice of "book", you will make "organization" invisible, and for the choice of "journal", you will make "authors" and "publisher" invisible. As a result, one interface will be adequate for adding a reference, even though it has two different looks for adding a book and a journal.



(3) Create an interface for searching references. When the user selects the "search" command, the current interface will be changed to the following. The interface is almost the same as that for adding a reference, except that any or all of the textfields can have empty values. In addition, the text area for "Messages" is now replaced by one for "Search Results". Again, you will need to use scrollable text areas so that the user can see the full content if needed.



(4) Add exception handling to your program so that any errors or violations to certain conditions (such as class invariants) can be handled more elegantly. More specifically, you need to throw exceptions in constructors and mutator methods whenever the class invariants are violated. You should also use exception handling to check for incorrect input so that the user will be allowed to try again for correct values.

(5) Bonus (up to 30 marks). You are encouraged to add more extensions and improvements to the basic requirements specified above. This may require that you read the textbook or search the Internet for more examples and ideas. Please document it clearly in your "README" file so that the TA can examine any new features you may have made in your implementation.

Deliverables: Same as Assignment Two. However, if you use any additional packages other than the standard ones that come with Java 1.5 or higer, you need to include them with your submission so that the TA can properly test your implementation.