Problem Statement

ISB is suffering from organizational inefficiency in storing and calculating appropriate awards for athletes at the end of each year.

Word Count: 20

Description of Scenario

As learned in my interview with Mr. Quinet, the athletics director, currently, all team rosters are in paper, stored in physical binders held in the athletics office (Appendix A, Q.1). In order to determine what awards each athlete should receive at the end of the year awards ceremony, the athletics office must sift through paper after paper, listing accolades and years in a painstaking and an extremely inefficient manner. As such, Mr. Quinet has said he is in need of a way to easily store and access the team rosters, and to then be able to easily see what awards players should, or shouldn't, receive at the end of each year (Appendix A, Q.2).
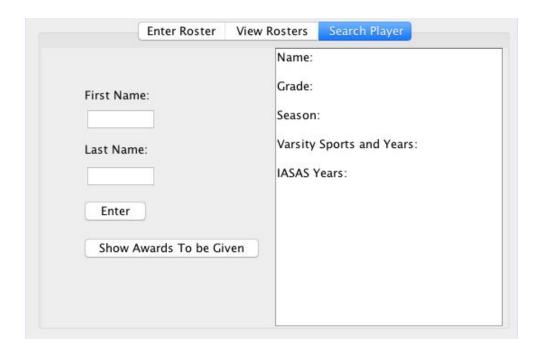
Word Count: 108
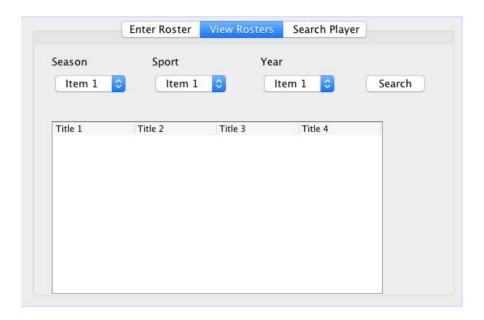
Rationale for the Proposed Product

Mr. Quinet explained that he needs a means of inputting various attributes, including Player names, positions, year, and sport (Appendix A, Q.3). A GUI database through the Netbeans IDE will therefore be made, allowing for a simple to use user-interface. He of course also needs to be able to search for any player in the system (or find that that player is not yet in the system), and to then see what accolades that player holds, as he requested (Appendix A, Q.3). To this end, a GUI with a simple search and display function will be able to search through the stored rosters, and output whether the name searched was found, and what awards should be given. Java is an effective, and for me, familiar, means in searching and sorting data, thus enabling me to create an effective solution to the outlined problem.

Word Count: 137

Prototype 1

Criteria for Success:

Function

1. Client can search for a name, roster, sport, and year.
    a. The client will be able to separately search within these parameters.
2. Program displays the history of that name in ISB athletics, and what awards they deserve
3. Client can enter in a team roster that saves names individually
4. Client can enter the sport, date, and position of a player
5. Client can view all saved rosters in a table
6. Client can edit/modify rosters that have already been entered
7. Names and all according data is saved in a database
8. Program may indicate whether the player deserves a new IASAS bag
9. Saves database to a local file on the computer.
10. Client will be able to edit rosters already saved in the database, in case an error was made, or a player's attributes must be changed

Usability

11. Help button
    a. Will inform the user how to use each function
        i. Providing an example.
        ii. Explanation may be opened in a new window
12. Combo boxes will be used as a GUI element, making data input easier.

13. Text fields will clear after each entry has been submitted
14. For important functions, a prompt will be opened to confirm that the user does indeed wish to carry out the following action.

   Error Handling

15. When names must be inputted, it will be verified whether the user has entered a String.
    a. If not, a JOptionpane will be outputted, asking the user to input the correct data type.
    b. The same will be done for required integer inputs.
16. If the user searches for an element that does not exist in the database, a JOptionpane will be outputted, notifying the user.
17. If the user attempts to search or submit data when all necessary fields or selections have not been filled or made, a JOptionpane will be outputted, notifying the user, and halting any processes that would lead to an error.