Heuristic Evaluation

1. Validity of system status
   a. Since we will need to do page scraping to get information, we will need to have “loading functionality”. To communicate this to the user, we will have a dialog within the bounds of the module being loaded, but on top of any information.
   b. The aforementioned loading display would appear if the loading time eclipsed the standard 500ms, giving the user feedback.

2. Match between system and the real world
   a. Since we will be scraping information from other sources, this information may contain technical jargon. That said, our interface will use simple, accessible language.
   b. We will not use technology-specific terminology in our interface; instead, we will make use of task-oriented terms.

3. User control and freedom
   a. We will have a standard navigation toolbar. If at any point the user needs to go back, or go back to the start page, they will be able to do so.

4. Consistency and standards
   a. Actions like specifying module content, finding context-specific information, and searching will be implemented consistently throughout the interface.

5. Help users recognize, diagnose, and recover from errors
   a. Error messages will not contain error codes or jargon.
   b. They will explain what went wrong and a step-by-step procedure to remedy the problem.

6. Error prevention
   a. It will not be easy to make errors. For example, submission of POST requests will require a confirmation before being processed. GET requests will not be used for any actions that could cause state changes in the system. URLs of requests submitted using GET will be checked for invalid input.

7. Recognition rather than recall
   a. Overlays will be translucent so that background information or data is always visible. Important actions like navigation and “cancel” requests will never be blocked.

8. Flexibility and efficiency of use
   a. It will be possible to set up shortcuts for specific favorite commands to accommodate and retain advanced users. Users can also customize their interface to reflect their particular usage. If the user think they are being given too much information, they can simply choose to hide it.

9. Aesthetic and minimalist design?
   a. Our design will be spartan like Google’s, but we don’t plan on an “I’m feeling lucky” button.

10. Help and documentation
    a. Context-relevant help will always be available in addition to a main help section.

Cognitive Walkthrough

I want to find out if I have an assignment for my CS 465 class. I should be able to find out what the assignment is, when it’s due, if there is any help on the newsgroup, and if any e-mails have been sent out about it.

- A user goes to the project website, and must log in.
  - All logins are already saved, so once the user is logged into the project website, they do not need to perform any further logins.
- Once logged in, the user sees a page with an overview of all of the categories of information available to him or her plus his or her favorites.
- The user selects CS 465 through either his or her favorites or the “Classes” category (menu navigation).
- The user is brought to a summary page with the most recent and relevant information (according to their settings) for their CS 465 class.
- The user can select “Homework” from the predefined choices or do a search for “Homework” if the current assignment is not within the recent and relevant information.
- The current assignment can then be selected, and the user will be taken to a page with links to the relevant information. Some information may also be scraped from the pages and displayed directly to the user.
- The user now has access to the information he or she initially set out to find.
- When the user is finished and wishes to return to the home page, there will be several options available - such as an exit button, a back button, and relevant keyboard shortcuts.