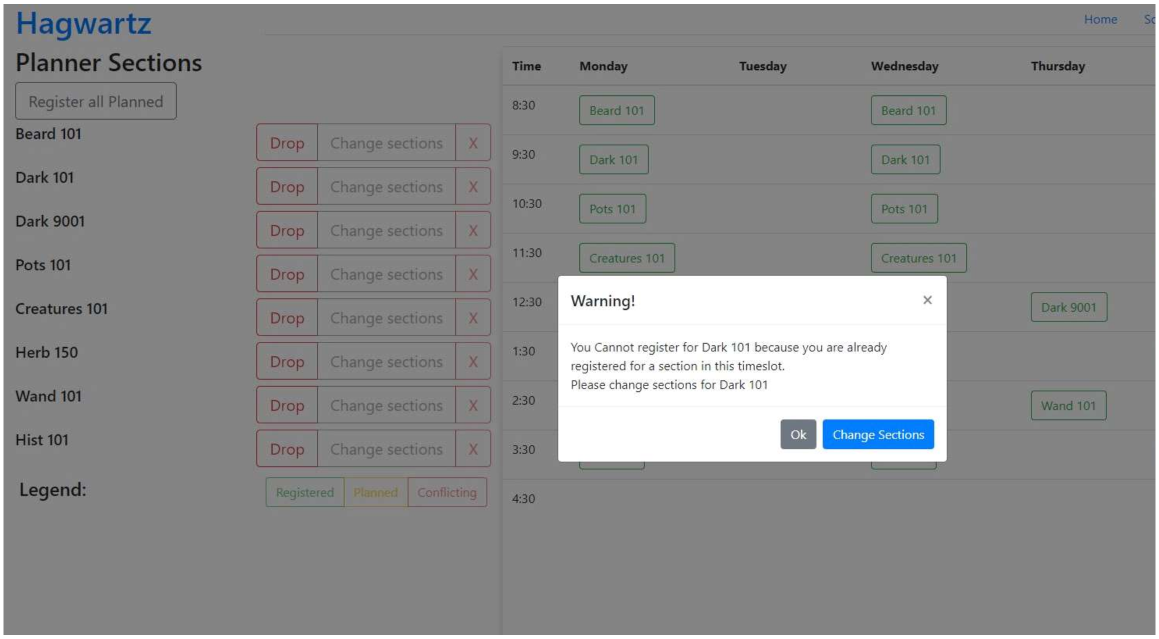
Our heuristic evaluation method consisted of three evaluators which is each person in our group. The procedure of our heuristic evaluation method was to take each heuristic from Nielsen's Heuristics and compare them to each page in our course registration system. Individually, we found as many heuristics as we could that matched what was seen in our interface, then we would compare them as a group. Next, we ranked each heuristic from most significant to least significant based on how much of the given heuristic was seen in our interface. The data below is presented with these heuristics ranked in order of significance and the heuristics that weren't seen in our interface are absent. Also, the heuristics of Help Users Recognize, Diagnose, Recover from Errors and Error Prevention were combined since the problems reflected in our system were so closely related. For each heuristic, there will be a paragraph of the problems that we have seen reflected in our system. Following, there will be a paragraph indicating some possible solution we could implement in the next iteration of our interface. Depending how well the problem can be visually captured within our system, we will include a screenshot that can better indicate the problem at hand.

Help Users Recognize, Diagnose, Recover from Errors + Error and Prevention



Beard 101
Dark 101
Dark 9001
Pots 101
Creatures 101
Herb 150
Wand 101
Hist 101

Time	Monday	Tuesday	Wednesday	Thursday
8:30	Beard 101		Beard 101	
9:30	Dark 101		Dark 101	
10:30	Pots 101		Pots 101	
11:30	Creatures 101		Creatures 101	
12:30				Dark 9001
1:30				
2:30				Wand 101
3:30				
4:30				

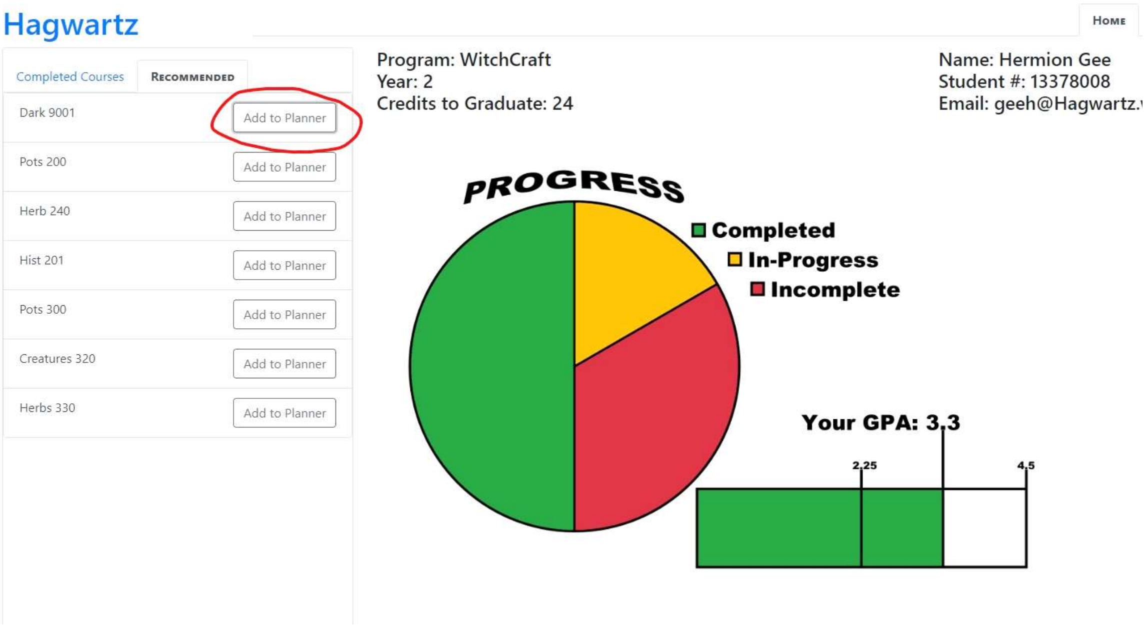
Warning!

You Cannot register for Dark 101 because you are already registered for a section in this timeslot.
Please change sections for Dark 101

Throughout our system, the heuristics of help Users Recognize, Diagnose, Recover from Errors and Error Prevention were violated a few times. The main issues with the error heuristics was with our "register all planned" button. If the system runs into an issue, it outputs an error message to help the user fix the issue, but if it encounters multiple issues it only outputs the error message from the first issue. This means our message which is made to help the user, only helps the user recognize, diagnose and recover from one issue instead of all of them. This makes error prevention more difficult and makes the registration page fail these heuristics. Another violation of these error heuristics is in the compare page of the course catalogue. If a user removes a course from the compare list there is no way for them to undo this action, making errors easily unrecoverable. Similarly, there is the inability to undo adding courses to the planner from the recommended tab in homepage, thus not giving users the ability to recover from errors.

To solve our first issue, we could have the option to implement one of two fixes. First would be to make the error message include the names of all the courses with conflicts and give the user the ability to change all their sections and fix the conflicts. The other fix would be to make multiple error messages appear and once the user resolves or passes on the first issue the next message would appear. A solution to one of our other violations is to simply add an undo button to the compare function in the course catalogue which would add back removed courses. This would allow the user to recover from error easily and make the compare function pass the error heuristics. For our last violation, we can solve our issue of not being able to undo adding courses to the planner by making the "add to planner" button change to "remove from planner" if the course is already in the planner, just like we have done in our course catalogue search page. With these changes we can help protect our users from making unrecoverable errors.

Consistency and Standards



Program: WitchCraft
Year: 2
Credits to Graduate: 24

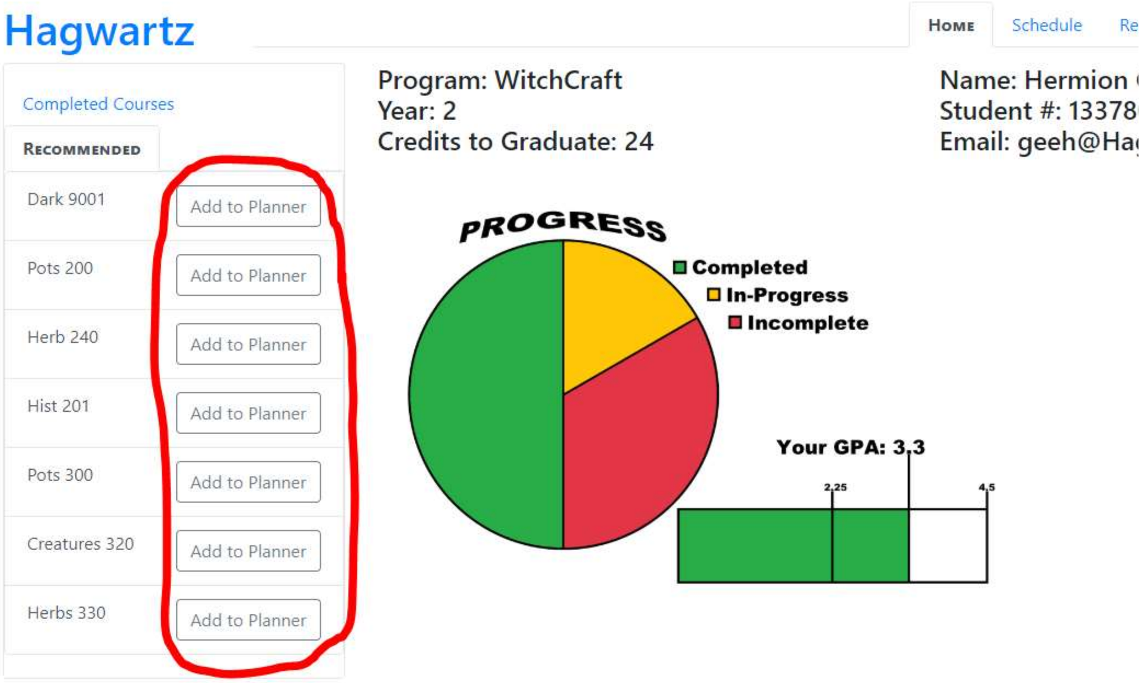
Name: Hermion Gee
Student #: 13378008
Email: geeh@Hagwartz.

Your GPA: 3.3

The Nielsen heuristic of consistency and standard was not completely followed in our interface. For example, on our homepage we have the "add to planner" button, and unlike similar buttons across our interface, it does not change from "Add to Planner" to "Remove from Planner" when the user adds or removes the course. This violates the heuristic by creating a lack of consistency across our interface. Also, the homepage lacks in this heuristic where the courses in the "Recommended Courses" tab are missing their short names which would link to the info on the course catalogue search page. This results in an inconsistency among similar actions over the interface. One last example of the violation of this heuristic is the fact that the list in the search function appears to be alphabetical. However, if you scroll all the way down you see that it isn't. This is breaking the consistency and standards heuristic by making a list seem to follow a general standard where lists are presented in alphabetical order, but actually not.

To fix this heuristic it is relatively easy. To start we should add the course names on the "Recommended Courses" tab of the homepage which would link the user to the information page on the course catalogue, as it is done on other pages. Next, we should make the buttons in this tab switch from "Add to Planner" to "Remove from Planner". These two changes make the homepage consistent with the other pages and therefore fixes their violations of the heuristic. The final fix we would need to implement would be to make the course list in the search part of the course catalogue alphabetical.

Visibility of System Status:



Program: WitchCraft
Year: 2
Credits to Graduate: 24

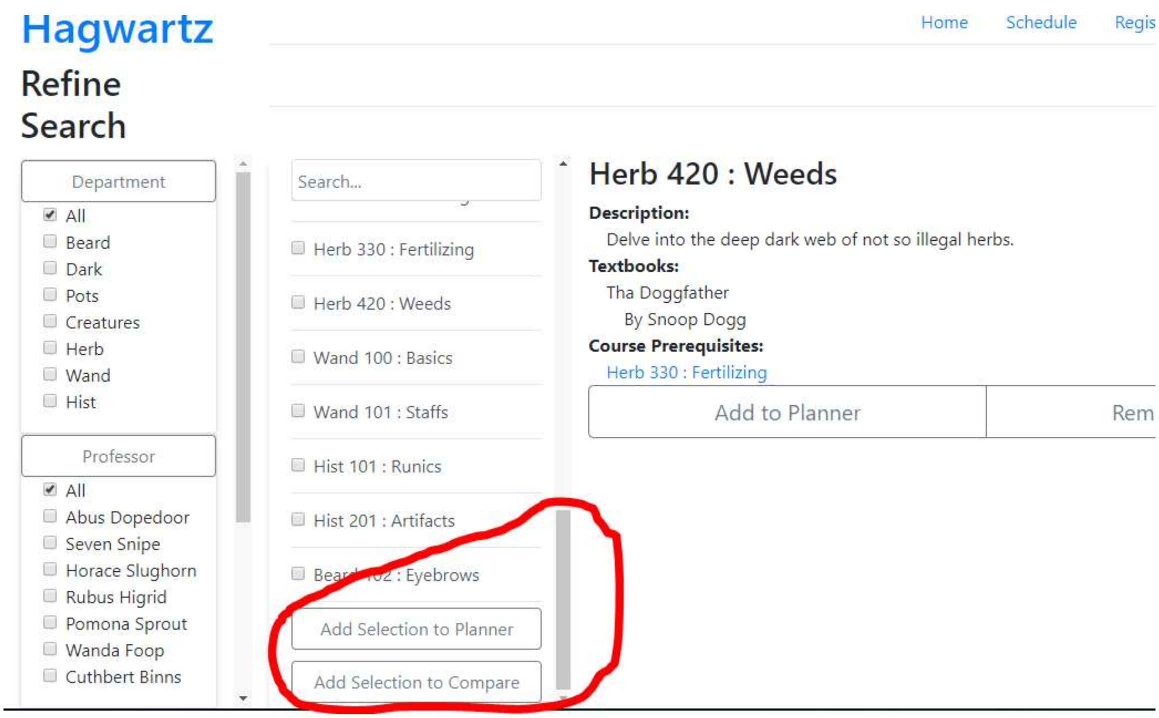
Name: Hermion
Student #: 13378
Email: geeh@Ha

Dark 9001
Pots 200
Herb 240
Hist 201
Pots 300
Creatures 320
Herbs 330

For the Visibility of System Status heuristic, we found that there was a lack of system feedback for the user in more than one area. On the homepage the [Add to Planner] buttons did not indicate courses that were already in the planner nor if a course had been added to the planner. On the Registration page there was no clear indication as to what the [X] buttons in the course list would do when pressed. And when the user clicked to register in a course it would not give any indication of success other than changing the colour of the element in the schedule outline.

A potential overarching solution for all these issues would be to have a small notification pop up in the bottom corner of the screen telling the user whether the operation was successful (A modal with no need for further user input). As such for the homepage a simple '<Course Name > Added to planner!' would suffice. For the registration page the messages could range from 'Successfully Registered in <Course Name>' to some other errors that may occur during registration (incomplete prerequisites, holds on the account, waitlist information...etc.) and information on how to address those issues.

Recognition over Recall:



Herb 420 : Weeds

Description:
Delve into the deep dark web of not so illegal herbs.

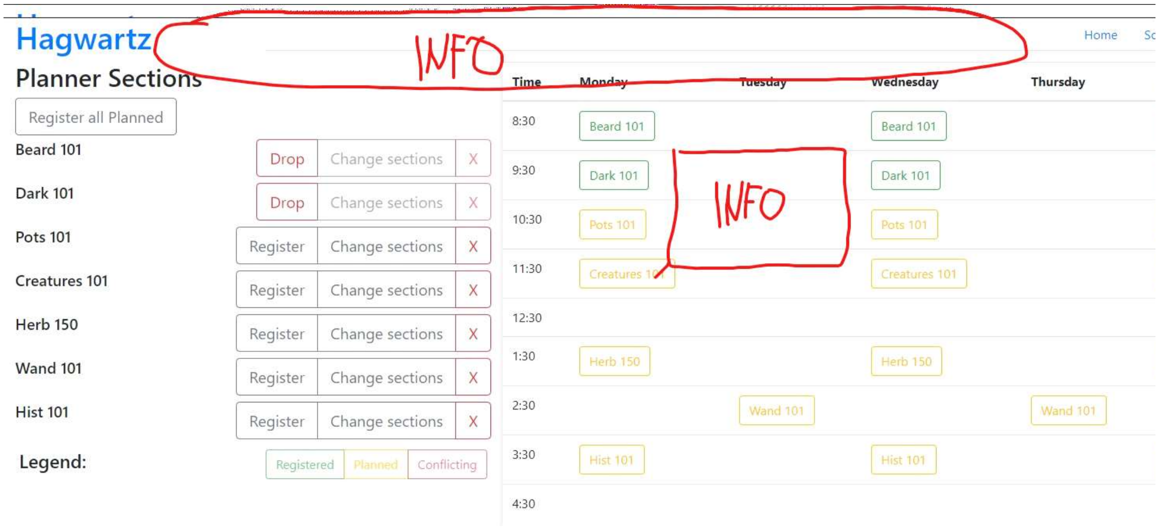
Textbooks:
Tha Doggfather
By Snoop Dogg

Course Prerequisites:
Herb 330 : Fertilizing

As for the Recognition over Recall heuristic, we found that on the search page the two buttons to add the selection to planner and compare were at the bottom of the list of courses. This forces users to remember that when selecting courses, they must scroll to the bottom of the list in order to actually perform some action on those boxes that are checked.

Thankfully there is a simple solution that would be to bring the buttons out of their parent container and sit them below the scrollable list, so they are always visible to the user. This would allow the users to recognize the potential for mass additions to the two areas for further analysis and action.

Help and Documentation



Planner Sections

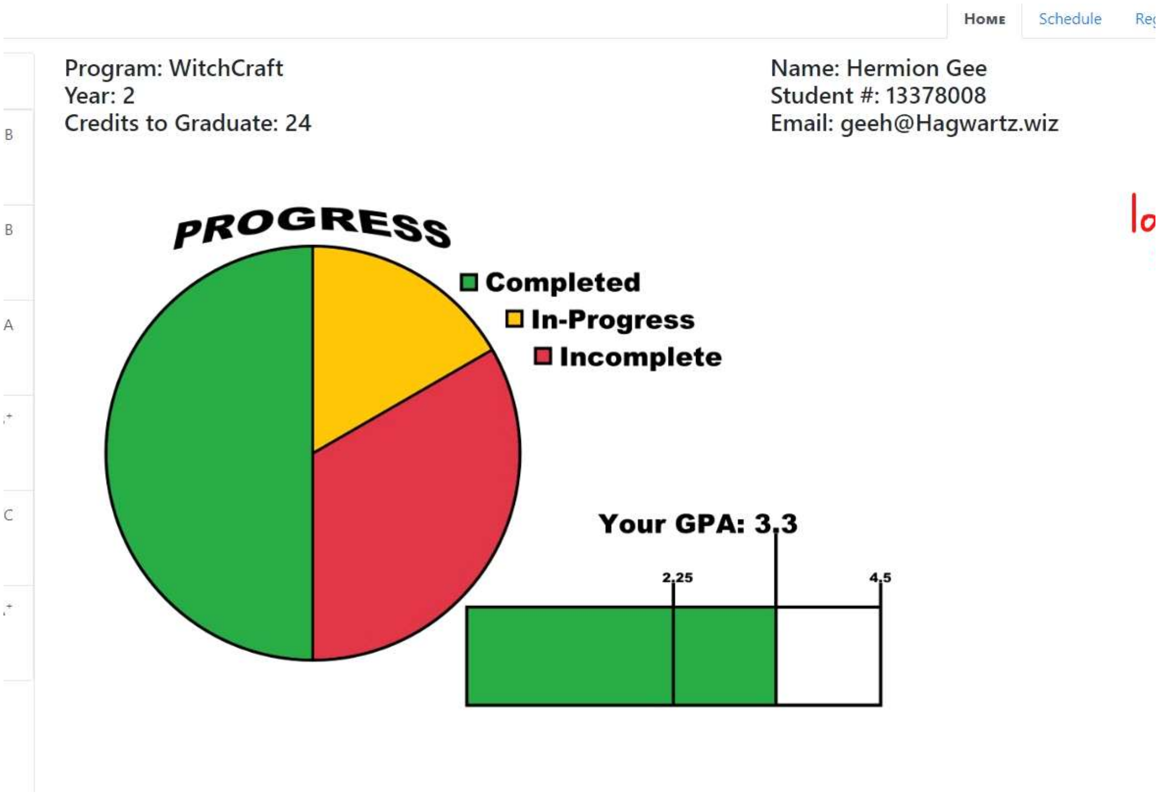
Beard 101	Drop	Change sections	X
Dark 101	Drop	Change sections	X
Pots 101	Register	Change sections	X
Creatures 101	Register	Change sections	X
Herb 150	Register	Change sections	X
Wand 101	Register	Change sections	X
Hist 101	Register	Change sections	X

Time	Monday	Tuesday	Wednesday	Thursday
8:30	Beard 101		Beard 101	
9:30	Dark 101		Dark 101	
10:30	Pots 101		Pots 101	
11:30	Creatures 10		Creatures 101	
12:30				
1:30	Herb 150		Herb 150	
2:30		Wand 101		Wand 101
3:30	Hist 101		Hist 101	
4:30				

Throughout our entire interface, we implement many unique aspects that may be not known to most users. Aspects such as, having a course planner and distinguishing between "Add Course to Planner" and registering for a course may be confusing. We should have some way of showing users the very basics of our interface and what they can expect by pressing specific buttons. At the current state of our interface, there is no way of giving help to the user if they do not understand the concept of our system.

To solve this problem for the next iteration of our interface, we can introduce some tooltips on each page. Depending on which button the user hovers over with their mouse, the interface shows a bit of text indicating what that specific button will do. Also, we could add some text at the beginning of each page indicating the purpose of that specific page. This can give the user an idea if they are at the right place to complete their task within our interface.

User Control and Freedom



Program: WitchCraft
Year: 2
Credits to Graduate: 24

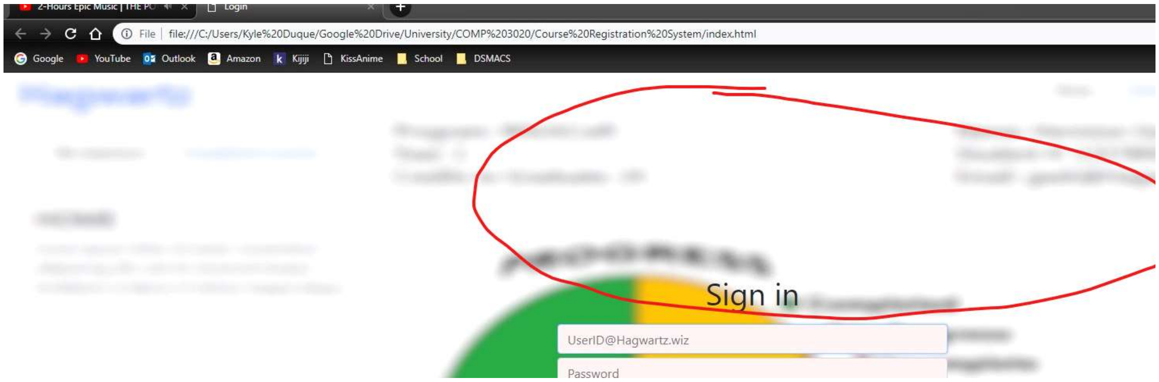
Name: Hermion Gee
Student #: 13378008
Email: geeh@Hagwartz.wiz

Your GPA: 3.3

While navigating any of our pages within our course registration system you'll notice that there is no way of logging out of the account that you're in. This limits the control and freedom of our users because they must exit our webpage and re-open it to be able to switch users. This is a round about solution that can be easily solved. This problem will be even more predominant in a household that only has one device for many users of our course registration system. Depending on the settings of their browser, they might return to the previous user's information posing a security risk.

The simple solution of solving this problem is making sure there is an option to logout on every page of our interface. By leveraging a user's prior knowledge of how a website, we can include the logout button on the top right-hand corner for easy memorability. Also, for extra security we can make sure the browser does not return to the previous user's data by forcing a logout after they exit the window.

Aesthetic and Minimalist Design


2-Hours Epic Music | THE PO
Login
File | file:///C:/Users/Kyle%20Duque/Google%20Drive/University/COMP%203020/Course%20Registration%20System/index.html
Google
YouTube
Outlook
Amazon
Kijiji
KissAnime
School
DSMACS
Sign in
UserID@Hagwartz.wiz
Password

A small aesthetic flaw that can be seen on the homepage of our registration system is that there's no indication of where you're logging into. Since this is the landing page for our interface, it is important for users to be able to distinguish which website they're on. Of course, the user can check the URL, but in the case of a visually impaired user, having written text for the browser to be able to read aloud is important.

Solving this problem can be done by adding a title and possibly some other details about our interface. Thinking about the different types of users, such as the visually impaired, we can add these aspects to shed light on what our interface offers.

Flexibility and Efficiency of Use

In our course registration system, we believe there is little room for accelerators since the process of adding and dropping classes cannot be broken down into luxuries such as keyboard macros. Also, it is difficult to speed up the process of adding or dropping a course since this is not like a workflow tool such as MS Word. However, speeding up the process of logging in by caching a user's information can act as an accelerator. It is a simple method of speeding up the time it takes a user to complete their tasks, but it can be effective especially if this interface is visited frequently.

To solve this problem, we can introduce a caching system in the next iteration of our interface. Whenever a user would start typing their username, it would auto-complete the field. Also, we could incorporate an option where the user can specify if they're on a private device which would allow saving their password as well.