The OSCAR Redesign Project Part 2 Design Alternatives CS3750 User Interface Design

> The Grouches: Jared Salzmann Jack Gruendler Jason Tongco Carlos Rios

Table of Contents

Introduction	3
Requirements Summary	3
Design Space	3
Design Alternatives	4
Calendall	4
Rationale	4
Illustrations	5
Story Boards	8
Use Case Scenarios	13
Assessment of the Design	15
Action Counts	15
Pro/Con List	16
Class Basket	16
Rationale	16
Illustrations	17
Story Boards	20
Use Case Scenarios	22
Assessment of the Design	23
Action Counts	23
Pro/Con List	24
RequireGraph	25
Rationale	25
Illustrations	26
Story Boards	31
Use Case Scenarios	33
Assessment of the Design	35
Action Counts	35
Pro/Con List	36
Comparison Between Designs	37
Reflections	37

Introduction

This OSCAR redesign is, first and foremost, not an overhaul of the backend. Such changes are beyond the scope of this project. This redesign is focused upon the user experience and placing within easy reach of the user the most-used or most-requested functions.

Requirements Summary

First and foremost, the redesigned OSCAR should have easy access to the functions students use frequently. Specifically, looking up classes, registering classes, and viewing schedules should be accomplished with a minimum of mouse clicks. It should be easy to navigate for the first time without the help of FASET volunteers. It should integrate more powerful search features, such as searching by GPA, and display features, such as plotting classes on a campus map. Display of information should be better organized: the user shouldn't have to look at several different pages to find out everything about a class (in contrast to the current system, which requires the user to look at the search results, click on the CRN for the class, then click "View Catalog Entry" to see all information). It should be more customizable and intelligent, by having persistent user settings (for things like search term and campus), with reasonable defaults (such as "current semester" and "main Atlanta campus"). It should not break when pages are opened in multiple windows.

Design Space

The design spaces that we have created will attempt to make navigation through the OSCAR system and completing tasks with speed and ease. One such design aspect we have is to present as much information on a single page without it being cluttered this aspect will lessen the amount of pages needed to navigate through, such as information about a course. Another design aspect is to prioritize tasks by presenting the most important tasks (i.e. registration, permit overload, etc.) upfront so that the user will not have to waster precious time navigating the site to accomplish them. One of the most important aspects is to not have users be set up for a fall. One example is having the user able to select a class that the user can't register for because of a major restriction or prerequisite; our designs will alert the user before such a thing will happen.

With the task of designing an interface comes the difficulty realizing certain requirements. One requirement is that we most likely won't be able to display the amount of information we want to into a single page. Another is the lack of customizability in the layout of the interface, such as moving certain parts of the page into other areas. Another is the lack multiple methods in completing a task, such as registering for a class; doing so will most likely go against the rules made by Georgia Tech in how the class registration should be implemented (i.e. shortcuts that give some users an advantage in registering for classes much faster). There are several tradeoffs in designing this interface. The tradeoffs discussed were:

- More information without being cluttered
- Deep vs. shallow navigation
- Visual vs. text based
- Constrained vs. wide open (only show available classes vs. all classes)
- Modal vs. combined vs. simplified
- Dynamic vs. sequenced (Ajax vs. POSTS) (synchronous vs. asynchronous)

There are multiple methods in how our interface could support certain tasks better than the current OSCAR system. One of them, most importantly, is registering for available classes faster. It may not seem obvious, but minutes, even seconds, make the difference when registering for classes that a user wants. In order to make registering faster, in terms of getting the desired classes and completing the task overall, our design will alert what classes they must take, furthermore which ones will be available. Another task that can be done much better than the current OSCAR site is navigation. The current layout and hierarchal design of the webpage brings confusion and focuses on functionality rather than priority. Our design will focus on both to make users accomplish their tasks faster and with ease.

Design Alternatives

Calendall

Rationale

The goal of this design is to help the student organize their classes and other school activities in a visual way on a weekly schedule. Calendar systems are fairly common and most students are familiar with them. A visual calendar provides direct manipulation of classes as they relate to time. The calendar also allows excellent visibility because it shows the classes that are registered and the most common class conflict students have (time conflict). With the use of a simple fuzzy search instead of advanced search parameters, this design eliminates clutter. As we found in research, students spend a lot of time deciding which classes they want to take before they are aloud to register. To help the students in this situation, the calendar system has the ability to have multiple calendars to work with at once. Students can create multiple schedules that they would like to take. When it is time to register, the student can see which of the previously created schedules has all the classes open. The student then registers for all classes in the schedule with one click of a button.

Illustrations



Sketch 1







Sketch 3

Story Boards

Preference 1: Preference 2: All Classes After Available 10:05 AM	Preference 3: Last Resort	Name Year: Term Curren	: BURDELL, GEOI Senior Registration: Fall 20 nt Hours Registered	RGE 008 : 10 HRS	SCAR	v2.0
REGISTRATION MENU	Tîme/Day	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
SEARCH ENGINE SEARCH RESULTS ADD TRAVEL TIME ADD ACTIVITY	8:05-8:25 AM 9:05-9:25 AM 9:05-9:25 AM 9:05-9:55 AM 10:05-10:25 AM 10:05-10:25 AM 10:05-10:25 AM 10:05-11:25 AM 11:05-11:25 PM 1:05-12:25 PM 1:05-12:55 PM 1:05-12:55 PM	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443	ENGL 1101 Grayson, D Skiles 310 CS 1315 - A (Rec. Section) Potts, C Klaus 1402	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443 CHEM 1310 - G (Lab Section) Azizi, B CoC Lab 10	ENGL 1101 Grayson, D Skiles 310	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443

Standard Screen

Preference 1: Preference 2: All Classes After Available 10:05 AM	Preference 3:	Name: Year: 1 Term I Currer	BURDELL, GEO Senior Registration: Fall 2 11 Hours Registered	RGE 008 1: 10 HRS	SCAR	v2.0	
REGISTRATION MENU	Tîme/Day	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	
SEARCH ENGINE General Search Major Course Number Location Day Time Campus SEARCH RESULTS ADD TRAVEL TIME ADD ACTIVITY	8:05-8:25 AM 8:35-8:55 AM 9:05-9:25 AM 9:35-9:55 AM 10:05-10:25 AM 10:05-10:25 AM 11:05-11:25 AM 11:05-11:25 AM 11:05-11:25 AM	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443	ENGL 1101 Grayson, D Skiles 310	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443	ENGL 1101 Grayson, D Skiles 310	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443	
	12:35-12:55 PM 1:05-1:25 PM 1:35-1:55 PM		(Rec. Section) Potts, C Klaus 1402	(Lab Section) Azizi, B CoC Lab 10			F

User Clicks Search Engine

Preference 1: No CS 1050 Available Preference 2: All Classes After 10:05 AM	Preference 3:	Name Year: Term Curret	: BURDELL, GEO Senior Registration: Fall 20 nt Hours Registered	RGE 008 : 10 HRS	SCAR	v2.0
REGISTRATION MENU	Time/Day	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
SEARCH ENGINE General Search Major Course Number Location Day Time Campus SEARCH RESULTS ADD TRAVEL TIME ADD ACTIVITY	8:05-8:25 AM 8:35-8:55 AM 9:05-9:25 AM 9:35-9:55 AM 10:05-10:25 AM 10:05-10:25 AM 10:05-10:25 AM 11:05-11:25 AM 12:05-12:25 PM 1:05-12:25 PM 1:05-12:55 PM	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443	ENGL 1101 Grayson, D Skiles 310 CS 1315 - A (Rec. Section) Potts, C Klaus 1402	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443 CHEM 1310 - G (Lab Section) Azizi, B CoC Lab 10	ENGL 1101 Grayson, D Skiles 310	CHEM 1310 - G Azizi, B CoC 16 CS 1315 - A Potts, C Klaus 1443

User Types "calc 1"

Preference 1: Preference 2: All Classes After Available 10:05 AM	Preference 3: Last Resort	Name Year: Term Curre	: BURDELL, GEOI Senior Registration: Fall 20 nt Hours Registered	RGE 008 : 10 HRS	SCAR	v2.0
REGISTRATION MENU	Tîme/Day	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
SEARCH ENGINE SEARCH RESULTS MATH 1501 - A1 - Calculus I MATH 1501 - A2 - Calculus I MATH 1501 - A3 - Calculus I	8:05-8:25 AM 8:35-8:55 AM 9:05-9:25 AM 9:35-9:55 AM	CHEM 1310 - G Azizi, B CoC 16	ENGL 1101 Grayson, D Skiles 310	CHEM 1310 - G Azizi, B CoC 16	ENGL 1101 Grayson, D Skiles 310	CHEM 1310 - G Azizi, B CoC 16
MATH 1501 - A4 - Calculus I MATH 1501 - F1 - Calculus I MATH 1501 - F2 - Calculus I	10:05-10:25 AM 10:35-10:55 AM 11:05-11:25 AM 11:35-11:55 AM	CS 1315 - A Potts, C Klaus 1443		CS 1315 - A Potts, C Klaus 1443		CS 1315 - A Potts, C Klaus 1443
ADD TRAVEL TIME ADD ACTIVITY	12:05-12:25 PM 12:35-12:55 PM 1:05-1:25 PM 1:05-1:25 PM 1:35-1:55 PM		CS 1315 - A (Rec. Section) Potts, C Klaus 1402	CHEM 1310 - G (Lab Section) Azizi, B CoC Lab 10		

Search Results Are Populated



User Clicks on "Math 1501 – A2 – Calculus 1" and the description pops up

Preference 1: No CS 1050 Available Preference 2: All Classes After 10:05 AM	Preference 3:	Name Year: Term Curre	: BURDELL, GEOI Senior Registration: Fall 20 nt Hours Registered	RGE 008 : 14 HRS	SCAR	v2.0
REGISTRATION MENU	Time/Day	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
SEARCH ENGINE SEARCH RESULTS MATH 1501 - A1 - Calculus I MATH 1501 - A2 - Calculus I MATH 1501 - A3 - Calculus I	8:05-8:25 AM 8:35-8:55 AM 9:05-9:25 AM 9:35-9:55 AM	CHEM 1310 - G Azizi, B CoC 16	ENGL 1101 Grayson, D Skiles 310	CHEM 1310 - G Azizi, B CoC 16	ENGL 1101 Grayson, D Skiles 310	CHEM 1310 - G Azizi, B CoC 16
MATH 1501 - A4 - Calculus I MATH 1501 - F1 - Calculus I	10:05-10:25 AM	CS 1315 - A Potts, C Klaus 1443		CS 1315 - A Potts, C Klaus 1443		CS 1315 - A Potts, C Klaus 1443
MATH 1501 - F2 - Calculus I	11:05-11:25 AM	MATH 1501 Brakebill, L Howey Physics L1	MATH 1501 Brakebill, L Howey Physics L1	MATH 1501 Brakebill, L Howey Physics L1	MATH 1501 Brakebill, L Howey Physics L1	MATH 1501 Brakebill, L Howey Physics L1
ADD TRAVEL TIME	12:05-12:25 PM 12:35-12:55 PM 1:05-1:25 PM 1:35-1:55 PM		CS 1315 - A (Rec. Section) Potts, C Klaus 1402	CHEM 1310 - G (Lab Section) Azizi, B CoC Lab 10		

User Adds the Class to the Schedule

Use Case Scenarios

Name: The student adds Calculus 1 to the weekly schedule.

Stakeholders: The student

Preconditions: The student must be logged in and at the class registration page.

Post conditions: A Calculus 1 class is on the weekly schedule.

Main Success Scenario:

1) The student clicks in the "Class Search" text box field

2) The student types in "calc 1" and presses enter.

3) The area below the "Class Search" text box field expands to show the results of the search. "Math1501A", "Math1501B", and "Math1501C" are the search results. Limited class information is displayed in this area.

4) The student mouses over "Math1501A" and the "Math1501A" result is highlighted along with it's location in the weekly schedule—based upon the class time.

5) The student clicks on "Math1501A" and an overlay screen opens up with all of the detailed class information such as: prerequisites, location on campus map, seats remaining, etc.

6) The student closes the overlay screen.

7) The student double clicks the "Math1501A" search result and the class is now added to the weekly schedule.

Extensions and Alternative Flows:

If there are class conflicts with previously added classes, the color of the highlighting changes. Open Issues: None

Name: The student adds a class at 2pm MWF. Stakeholders: The student

Preconditions: The student must be logged in and at the class registration.

Post conditions: A class is registered at 2pm MWF.

Main Success Scenario:

1) The student puts his mouse on the 2pm block in the Monday column. The same time block in Wednesday and Friday are highlighted.

2) The student clicks once

3) The area below the "Class Search" text box field expands to show the results of the search. All classes that start at 2 or 2:30 pm are displayed including. "Math1501B" and "CS1371C" are the search results. Limited class information is displayed in this area.
4) The student mouses over "CS1371C" and the "CS1371C" result is highlighted along with it's location in the weekly schedule—based upon the class time.

5) The student clicks on "CS1371C" and an overlay screen opens up with all of the detailed class information such as: prerequisites, location on campus map, seats remaining, etc.

6) The student closes the overlay screen.

7) The student double clicks the "CS1371C" search result and the class is now added to the weekly schedule.

Extensions and Alternative Flows:

If there are class conflicts with previously added classes, the color of the highlighting changes.

Open Issues: None

Name: The student makes a copy of a weekly schedule.

Stakeholders: The student

Preconditions: The student must be logged in and at the class registration. The student must have a weekly schedule filled with desired classes.

Post conditions: Two weekly schedules are now available.

Main Success Scenario:

1) Clicks on the drop down arrow next to the "Schedule 1" tab at the top of the screen.

2) The student selects the "Duplicate Schedule" button

3) A new weekly schedule called "Schedule 2" is added next to the "Schedule 1" tab.

4) The student clicks on the "Schedule 2" tab and sees the same classes on that schedule as schedule 1.

Extensions and Alternative Flows:

The student can change each schedule independently. The student can add, delete, move, and rename schedules. Open Issues: None

Assessment of Design

This design is an excellent way to eliminate time conflicts and to create multiple versions of schedules so that the student can choose from one of the alternate schedules if one of his classes is full. The design is familiar to students who have used calendar programs. Searching is also familiar to students who use search engines that try to infer what you are looking for rather than a straight, comparison search. The use of highlighting, colors, and visual representation of classes on a weekly schedule makes this design have high observability. This design is flexible because it allows the user to maintain separate weekly schedules (multithreading) that are persistent between logins. It also allows the user to add classes in multiple ways: text search and time selection.

The design is limited because so much space is taken up by the weekly schedule. The search results cannot have all the information that students need. A possible work around would be to have the option to make the calendar small and the search results large. Students may not first realize that classes can be searched for by selecting a time on the calendar. The design may not be flexible enough to serve all users. For example, if a user is currently registering for classes, there is no need to create and save multiple weekly schedules.

Action Counts

Alice wants to register for all the CS classes she can take. The CS classes that she can take are: CS1371 and CS2110. There are two CS1371 sections, section A is at a time that is conflicting with POL1101—a class she must take. Action counts from only POL1101 registered, to the three classes registered.

- 1. Click "Search Engine"
- 2. Home to "General Search"
- 3. Type "CS1371", press enter (6 steps)
- 4. Hover over CS1371A, see there is a conflict
- 5. Hover over CS1371B, see no conflict
- 6. Double click CS1371B, see it added to schedule
- 7. Click "Search Engine"
- 8. Home to "General Search"
- 9. Type "CS2110", press enter (6 steps)
- 10. Hover over CS2110A, see no conflict
- 11. Double click CS2110A, see it added to schedule
- 12. Home to "Preference 1" schedule tab
- 13. Choose menu and select "Register All These Classes"

Total Steps: 23

Pro/Con List

Cale	ndall
Pro	Con
Good for planning alternate schedules	Class details are muted
Time Conflict Visibility/Resolution	Campus map isn't integrated
Calendar is common interaction	Not class centric
Could integrate with other calendar programs	No requirement consideration
Could show things other than classes in calendar	Advanced searches require extra step
Fuzzy search good for most searches	Not all users prepare in this way (create multiple schedules)
Register for all classes at once	

Class Basket

Rationale

This design allows for "quick and easy" mass registration. While this can lead to undesirable, automatically generated schedules, Class Basket allows for filters and search criteria to be applied pre- and post-search, helping the user to narrow the results to those which are relevant. The driving impetus for this design, however, is familiarity. Class Basket looks and operates similarly to shopping websites such as ebgames.com and newegg.com. It is this familiarity which will enable new users to quickly adapt to Class Basket. It leverages previous knowledge and experience to get the user quickly acclimated and searching and registering for classes.

Illustrations



Sketch 1

LOGIN



Sketch 2



Sketch 3

Keywords Select All Select None FILTER BY: MAJOR INSTRUCTOR DAY TIME LOCATION CAMPUS	LOGIN HOME STUDENT SERVICES	VIEW QUEUE	OSCA INVOICE		2 • D Help	Na Ye Te Cu	ame: BURDELL, ar: Senior rm Registration: I rrrent Hours Regi	GEORGE Fall 2008 stered: 10 HRS
Search MaJOR MAJOR Image: Compare Selected	Keywords Search Recet MAJOR Image: Construction of the sector of the	Select All FILTER BY: Select All Compare Selected	Select None MAJOR Select None	INSTRUCTOR	DAY	TIME	LOCATION	CAMPUS

Story Boards

Main screen

Select All Select None 4803 FILTER BY: MAJOR INSTRUCTOR DAY TIME LOCATION CAMPUS Search Rest MAJOR MAJOR INSTRUCTOR DAY TIME LOCATION CAMPUS MAJOR MAJOR MAJOR INSTRUCTOR DAY TIME LOCATION CAMPUS CAMPUS C CAMPUS C EE E	LOGIN HOME STUDENT SERVICES	VIEW QUEUE	OSCA INVOICE	R VE	2 • O Help	Nan Yeau Terr Cun	ne: BURDELL, r: Senior n Registration: rent Hours Regi	GEORGE Fall 2008 istered: 10 HRS
Select All Select None	Keywords 4803 Search Reset MAJOR Image: Comparison of the sector of the secto	Select All FILTER BY: Select All	Select None MAJOR Select None	INSTRUCTOR	DAY	TIME	LOCATION	CAMPUS

The user enters a keyword and a major.

LOGIN HOME STUDENT SERVICES	VIEW QUEUE INVOICE	EAR V	2 - 0 rs help	Nan Yeai Terr Curr	ne: BURDELL, r: Senior n Registration: I rent Hours Regi	GEORGE Fall 2008 stered: 10 HRS
Keywords	FILTER BY: MAJO	OR INSTRUCTOR	R DAY	TIME	LOCATION	CAMPUS
Search Reset	CS 4803 - A: Special Topics Ferguson, R 3 Credit Hours	MWF 9:05 - 9:55AM	Klaus 1447 Atlanta Campus	More Info	Add Queue	Register
CS INSTRUCTOR ▼	CS 4803 - B: Special Topics Doe, J 3 Credit Hours	MWF 3:05 - 3:55PM	Klaus 1428 Atlanta Campus	More Info	Add Queue	Register
DAY ▼ TIME ▼	CS 4803 - F: Special Topics Potts, C 3 Hours	MWF 1:05 - 1:55PM	Klaus 1447 Atlanta Campus	More Info	Add Queue	Register
	CS 4803 - J: Special Topics Soo, K 3 Credit Hours	TR 1:05 - 2:35PM	Klaus 1434 Atlanta Campus	More Info	Add Queue	Register
Search Reset						
Links Registrar www.registrar.gatech.edu						
Bursar www.bursar.gatech.edu						
Financial Aid www.finaid.gatech.edu	Select All Select N	ione				
	Compare Selected					

Search results are displayed

Use Case Scenarios

Persona: Joan is a 3rd year INTA major. She's already taking her required major classes, but she wants to try her hand at a CS course. She wants a CS course that has a minimum of programming, but is still interesting.

Name: Registering for one class (CS 4803LK) Stakeholders: Joan Precondition: Joan must be logged into OSCAR. Post condition: Joan is registered for the class she is looking for. Main Success Scenario: 1. Joan searches for CS 4803.

- 2. Sections of CS 4803 appear and are available for registration.
- 3. The student clicks on the "Register Now" button for the LK section.
- 4. The student is now registered for the class.

Extensions and Alternative Flows:

1. If the student does not meet the class prerequisites or if no sections matching search criteria have open seats, the "Register Now" button is grey is not clickable.

Open Issues: none

Name: Checking a class' prerequisite information Stakeholders: Joan Precondition: Joan must be logged into OSCAR. Postcondition: Joan has found the relevant information. Main Success Scenario:

1. Joan searches for CS 4803.

2. Sections of CS 4803 appear.

3. Joan finds the section she wants and clicks the "More Info" button.

4. A pop-up window containing the course catalog entry, required texts, instructor contacts, etc. appears.

Extensions and Alternative Flows:

1. If the class has only recently been offered, it may not have complete information available.

Open Issues: none

Assessment of Design

While sporting an interface immediately familiar to web-savvy students, Class Basket should be easy enough to be used by students who may not be as web-centric as their peers. The layout is such that one needs only glance to determine how a particular item should function. There are not many hidden functions, those being limited to the drop down boxes used in the initial search and the post-search filters available immediately above the search results. Functions for class-to-class comparisons and one-click registration further streamline students' decision-making process.

Class Basket's limitations are that there isn't a live-updating list of classes and there is no apparent calendar function. There is also no display of total and available seats in a class, though the system will disable registration for closed classes, pending overrides. The other potential problem is that a UI as extensive as Class Basket's may create undue slowdown during heavy periods of use.

Action Counts

Alice wants to register for all the CS classes she can take. The CS classes that she can take are: CS1371 and CS2110. There are two CS1371 sections, section A is at a time that is conflicting with POL1101—a class she must take. Action counts from only POL1101 registered, to the three classes registered.

Adding 1371: 1. change "major" drop box to "CS" (3) 2. type "1371" in keywords box (5) 3. click "search" (1) 4. sections A and B come up; A is unavailable ("register now" is grayed out) but B is available ("register now" is clickable) (2)5. click on section B's "register now" button (1)[subtotal: 12]

Adding 2110: 1. highlight "1371" in the keywords box and type "2110" (5) 2. click "search" (1) 3. see 2110A is available (1) 4. click on 2110A's "register now" (1) [subtotal: 8]

total: 20

Pro/Con List

Class E	Basket
Pro	Con
Fast way to develop schedules	Less direct control by users
Familiar system of shopping	NP-hard problem
System does the hard work	Lack of fine control of scheduling
Stores all desired classes	Lead to undesirable schedule
Register for all classes at once	No requirement consideration
	Campus map isn't integrated

RequireGraph

Rationale

We discovered in the requirements gathering phase that students often spend a lot of time deciding what classes to take, and that classes are often chosen based on whether they're required for the student's major or not. Therefore, presenting the classes to the user in a way related to major requirements makes sense because it provides a good visualization of progression through degree program requirements and prerequisites, and thus makes it easy to see what should be taken next. Conversely, it also reduces the number of irrelevant classes that clutter up the results page of the current system (when users don't bother to carefully craft their search criteria, as is often the case). Finally, because its hierarchy is based on the particular situation of the student rather than the organization of the Institute, it can be much quicker for the student to register for classes in different areas because they don't have to re-search.

Illustrations









View for a senior CS student



Story Boards



Initial view



Node for Math3012 selected



After expanding humanities courses and registering for LCC2500A.

Use Case Scenarios

Persona: Bob is a second-year CS student who has chosen the Computational Modeling and Platforms threads. He needs to take combinatorics (MATH 3012), a humanities elective (LCC 2500), systems and networks (CS 2200), his last lab science (PHYS 2212), and objects and design (CS 3240) this semester, but he does not know it yet.

Name: Bob registers for a class that he knows is required by name Stakeholders: Bob Preconditions: Bob must be logged in and at the class registration main menu Post conditions: Bob is registered for MATH 3012 Main Success Scenario: 1) Bob clicks on the item in the main menu labeled "Requirements Graph"

2) The graph of classes, pre-calculated to expand the ones most relevant to Bob, loads

3) Bob scans the picture, taking note of the bright blue nodes.

4) He glances up at the key, and sees that those blue nodes represent the classes he could choose to take

5) Among only the blue nodes, he looks for the one labeled "MATH 3012" and clicks on it

6) A description of the class, including the name of the professor, location, and meeting times for each section, appears in a pane of the web page below the graph7) Bob reads the description, picks the particular section he wants, and clicks the button labeled "Register" next to it.

8) The node labeled "MATH 3012" turns red to show that it's currently registered9) The "Register" button that he clicked becomes disabled and the list entry for that section will become highlighted

Extensions and Alternative Flows:

1) If all the sections of MATH 3012 are full, have a time conflict with Bob's existing schedule, or are otherwise unavailable, its node will be grayed out. If a section is unavailable, its list entry will be grayed out. If a section is unavailable for reasons other than a schedule conflict, the "Register" button on the description pane will be disabled.

2) If there is a schedule conflict:

 Bob can hover his cursor over the section (in the detailed description) he wants and the conflicting class in the graph will be highlighted in red.
 Bob can click the "Register" button for that section anyway, upon which a dialog box will prompt for confirmation that he wants to deregister the conflicting class

Open Issues: None

Name: Bob registers for an interesting humanities elective

Stakeholders: Bob

Preconditions: Bob must be logged in and at the class registration main menu Post conditions: Bob is registered for LCC 2500

Main Success Scenario:

Bob clicks on the item in the main menu labeled "Requirements Graph"
 The graph of classes, pre-calculated to expand the ones most relevant to Bob (which, since he is a CS major, do not include most humanities classes), loads
 Bob sees a node labeled "Humanities" and double-clicks on it to expand it
 Bob scans the blue nodes that have expanded from the "Humanities" node
 Bob finds one that looks interesting, in this case LCC 2500, and clicks on it
 A description of the class, including the name of the professor, location, and meeting times, appears in a pane of the web page below the graph
 Bob reads the description, and decides to take the course

8) Bob picks the particular section he wants, and clicks the button labeled "Register" next to it.

9) The node labeled "LCC 2500" turns red to show that it's currently registered 10) The "Register" button that he clicked becomes disabled and the list entry for that section will become highlighted

Extensions and Alternative Flows:

1) If all the sections of MATH 3012 are full, have a time conflict with Bob's existing schedule, or are otherwise unavailable, its node will be grayed out. If a

section is unavailable for reasons other than a schedule conflict, the "Register" button on the description pane will be grayed out.

2) If Bob decides, in step 7 above, not to take the course, he goes back to step 5. Open Issues: None

Assessment of Design

For its purpose -- figuring out graduation requirements, and registering for the needed classes -- RequireGraph excels. Classes can be found with significantly less effort than in traditional search-based interfaces, and prerequisites are expressed in a much more obvious and intuitive way.

However, it is not without limitations: first, the graph is a limited metaphor, and does not lend itself nearly as well to finding classes by other criteria (e.g. meeting times). When the number of valid options are high, as when selecting humanities electives, for example, the number of "relevant" nodes could be large enough to make the graph not fit on the page, both making panning necessary and (more importantly) overwhelming the user. It is also not particularly well-suited to resolving conflicts.

Action Counts

Alice wants to register for all the CS classes she can take. The CS classes that she can take are: CS1371 and CS2110. There are two CS1371 sections, section A is at a time that is conflicting with POL1101—a class she must take. Action counts from only POL1101 registered, to the three classes registered.

- 1. Find the node for CS 1371
- 2. Click on the node for CS 1371.
- 3. See that only section B is available because section A has a schedule conflict
- 4. Click on the "Register" button for section B
- 5. Find the node for CS 2110
- 6. Click on the node for CS 2110
- 7. Click on the "Register" button for the only section, section A

Total steps: 7

Pro/Con List

Requi	reGraph
Pro	Con
Visualization for what you need to take	Can't search/display results well
Eliminates superfluous classes	Color coding/naming/shapes requires a key
Directly manipulate visual objects	Can end up with too many objects on screen
User has control of what is visible	Position of object changes often
Can alter graphing options (sort by)	Browser running animated graphics
	Conflicts/scheduling doesn't fit the paradigm
	Lack of fine control of scheduling
	Could be CPU intensive
	Campus map isn't integrated

Comparison Between Designs

Criteria	Calendall	Class Basket	RequireGraph
Planning for alternate schedules	Very Good	Very Good	Very Poor
Time Conflict Visualization/Resolution	Very Good	Very Good	Neutral
Ease of search	Good	Neutral	Very Good
Result Relevance	Poor	Neutral	Very Good
Register for all classes at once	Very Good	Very Good	Very Poor
Class Detail Visibility	Poor	Very Good	Poor
Campus Map Integration	Poor	Neutral	Poor
Requirements/Prereq Visibility	Very Poor	Poor	Very Good
Speed To Register	Poor	Good	Very Good
Fine Control Of Schedule	Very Good	Neutral	Poor
No Clutter	Verv Good	Very Good	Verv Good
Direct Manipulation	Good	Neutral	Very Good
Predictability	Good	Good	Good
Synthesizability	Good	Good	Good
Familiarity Metaphor	Good	Very Good	Very Poor
Consistency	Neutral	Neutral	Good
Dialog Initiative	Good	Good	Neutral
Multithreading	Poor	Poor	Poor
Task Migratability	Neutral	Good	Very Good
Substitutivity	Good	Good	Very Poor
Customizability	Poor	Poor	Poor
Observability	Good	Neutral	Good
Doesn't set user up for fall	Good	Poor	Very Good
Recoverability	Poor	Good	Neutral
Task Conformance	Poor	Neutral	Good

Qualifiers: Very Poor - Poor - Neutral - Good - Very Good

Reflections

We were able to quickly brainstorm three ideas and were able to flesh out ideas. The process of drawing diagrams, providing rationale, assessment, etc took a lot longer. It took longer than expected. We probably started a to late and ended up staying up late. We had one person create the final illustrations for the poster but it was probably too much work for one person.