Welcome to the Degree Audit Interface Design! The following slides and accompanying text tell the story of how we arrived at our final design, through user interviews, personas and scenarios, paper-prototyping, interactive prototypes, and usability studies. All the while, we will refer to deliverables we produced which are located on our team website.
Phase 1: Proposal

- What is a Degree Audit?
  - Tool for viewing progress toward a degree
- Goals:
  - Create something useful for Olin
  - Design functional system

A degree audit system is a tool for viewing someone’s progress toward a degree. Here at Olin it’s used by students, faculty, and staff to track individual and collective student progress toward graduation.

Going into this project, we had two concrete goals. We wanted to choose an interface to design that would be useful for the Olin community, something that is pertinent to our lives and would be beneficial for people if done right. The degree audit system clearly fills that aim. Likewise, we ambitiously wanted to create a functional system, not just an interface. So we only looked at projects that lent themselves to the possibility of actually designing a complete back-end as well. Not surprisingly, we were not able to do this with our degree audit system, but it is important to note our ambitions going into the project.

More information about our driving forces and initial thoughts can be found in the Project Proposal under Phase 1.
Phase 1: Current System

This is a screen shot from the current degree audit system implemented on Student Information Systems (SIS). It is still in beta form because it cannot do AHS concentrations yet (and some other functions). It’s not a terrible system, but it’s far from perfect. It is not intuitive to read and can be quickly overwhelming. More about this and other existing systems can be found in our Comparative Analysis section under Phase 3.
In keeping with the principles of user-centric design, our first task was to locate and interview our users. We interviewed two staff members, two faculty members, and four students, one from each class. We then tried to concentrate the lessons we learned there from the interviews into the formation of three design personas, Audrey, Neil and Sophia. We intentionally created our student personas, Neil and Audrey, to represent two extreme user types, with Sophia being an extremely organized student who plans ahead, and needs a tool to help her with that planning, and Neil being a much more haphazard student, who chooses courses that he’s interested in, and who needs a tool that will help him analyze his courses and find a degree. Sophia, our faculty persona, is an extremely capable advisor who’s interested in her students, but who is busy with both teaching duties and her position as the ECE program chair. She needs a tool that helps her keep in touch with her students, even if she only has 5 minutes to use it.
Once we had a set of personas to consider, we built a task matrix to represent the things that we think they might use a degree audit system for. It includes who we think would perform each task, how important it is to them, and how frequently we think they would perform it.
At this point we took a step back and asked ourselves what the actual goal of our program was. Where should it fit into the lives of our users and personas? Ultimately, we agreed that the degree audit process was part of the overarching goal of deciding what to do with one’s life. At the top were the high level things like choosing a career, grad school, and other things our faculty users had all expressed the wish that they had more time to deal with. We decided to have our program take care of other tasks and free up the time for our users to have those one-on-one conversations.

A bit further down the list was information gathering, or deciding what specific major and graduation requirements should be and publishing the information so everyone can access it. This is already covered by course catalogs. At the bottom was making the actual choice of what a student would be doing at the next particular point in time, which we decided to leave to the students.

There was still one part missing, though. There were systems in place for choosing specific activities for the next semester (student registration) and finding a list of requirements a student must fulfill (course catalogs), but there was nothing to bridge the gap and compare next semester’s activities to the progress towards fulfilling requirements. This part of the process was being done by the students and advisors by hand. People aren’t very good at lots of automated math, but computers are. If our program could free up the time faculty and students used to spend number-crunching and allow them to have conversations and make important decisions instead, we’d consider our work a success. Further discussion of this decision can be found in the Design Space section in Phase 3 of our website.
After clearly defining our design space, we brainstormed many possible design approaches. This slide shows a sampling of some of our ideas, ranging from an iTunes-like interface to a pie chart to a bar graph. Several of these designs made it to our paper prototyping phase. More on these designs and their inspirations can be found in the Initial Design Ideas section of Phase 3.
After brainstorming we settled on three ideas to pursue in parallel design for paper prototyping, which we have called iAudit, Auditime, and AuditBar. Some of their features and user reactions may be seen in the slide. In short, users were familiar and comfortable with the layout of iAudit and Auditime because they were similar to Excel or iTunes, had re-sortable columns, expand and collapse buttons, check boxes, and other common and proven interface features. But users were intrigued by the different look and feel of AuditBar, which was a graphic-based bar chart. Although it made some initially uncomfortable, most said they would be willing to take the time to learn it in order to use its innovative features and concept model.

After some discussion, we decided that while iAudit or Auditime might be the simplest and most convenient way to track progress toward a degree, AuditBar offered a new and challenging approach that lent itself well to the goals we were trying to accomplish in this interface design course. Thus, we chose to continue to develop AuditBar as our interface prototype. More about the procedure, design, and feedback results of our Low Fidelity Prototype can be found in the Phase 4 section of our website.
For our first prototype, we decided to implement in Flash, which we felt was appropriate for a highly interactive graphical design. Notable changes include a modification to the color code, in which we used a gradient to add information about when courses were taken and an easier to read key. We also implemented cumulative statistics that track how many credits are needed to fulfill general degree requirements, as we had run into problems with our “finish-line” ideas in paper prototyping. Another significant change involved reorganizing how we displayed courses; this allowed us to remove redundancy (the same course appearing multiple times) which had confused users. The idea of display options was borrowed from iAudit, but only received nominal implementation. One of our biggest changes was to ensure consistency in the interface; our paper prototype involved varying results from the same action (i.e., clicking on two boxes that looked the same had very different results).
Phase 6: 2nd Interactive Prototype

- Changes:
  - Horizontal
  - Coloring!
  - Finish line
  - More display options
  - Returned groupings
  - One engineering column
  - All courses interactive
  - More degrees to audit

After finishing the 1st interactive prototype, we received our heuristic evaluation from other class members. This evaluation told us some things we already knew and some things we didn’t and was very important in our revision process for the 2nd interactive prototype which appears on the slide. We basically redesigned the whole layout, the most significant change being a switch from vertical to horizontal bars. This used the available screen space much more effectively. We also returned to grouping Math and Science together and AHS and E! together based on heuristic evaluation feedback, and brought back credit requirement finish lines, which we had discussed initially but left out in the first interface. We redid the coloring scheme and made it possible to color by year, grade, or course type. We were very happy with this second interface and thought that it came much closer to accomplishing some of our original goals with the interface. A more detailed description of our changes and our responses to the heuristic evaluation can be found in the Phase 6 section of our website.
With our updated prototype in hand, we returned to our users and performed 4 final interviews; two with users that had seen the paper prototype and 2 that had been interviewed but not used the system. The graphical layout received an overwhelmingly positive reaction, with users appreciating the easy access to information, and the different modes of information coloring that allowed them to check up on data at a glance. They also gave several helpful suggestions for the final design. Credit requirement finish lines made sense to some users and not to others, so we attempted to clarify this through coloring and adding text labels. More on-screen documentation was added in the form of labeling axes and making controls more visible.

The largest issue addressed was the overarching inconsistency of our course block grouping, which we discovered when our users asked us about which classes went in which bar. Our final design has a completely unambiguous layout; the top section includes only courses taken and the bottom section includes only courses not taken, with the bottom section divided further into general requirements (would have to complete regardless of which degree you choose) and degree-specific classes. Requests also included things like the ability to upload your custom-made degree requirements for those who created their own majors, the ability to dynamically prototype one’s future hypothetical schedule, and search functionality. We weren’t able to address these in the final prototype due to time constraints, but would like to see them in future versions of the program. Further discussion of the final prototype can be seen in the deliverables for Phase 7.
Well, as they say, hindsight is 20/20, and we’ve certainly learned some lessons from this process. The most important of these, we feel, is that as Cookie Monster would put it, “Goals aren’t sometimes foods.” At the beginning of the project, our primary goals were to create something that people at Olin could use, and to make it actually function. Neither of these are things that we feel we fully accomplished. When we looked at what would make a system useful for Olin, we decided that one of the most important needs is the ability to customize the system to deal with the exceptions that often seem to make up the Olin curriculum. Over time, though, we tended to collapse the system down, with the idea that it would work for most people, and we didn’t have time to implement all of the customization features. This ties into our other more notable lesson, that it is very easy to get in over your head at the beginning of a project. Inevitably, things will take longer than it seems they should, and when you end up behind on the project, you rarely put a lot of thought into what you leave out. This ties into our final lesson, which is that it’s very easy to let exceptions and loose ends be forgotten when you’re creating a system, but that the ability to deal with those exceptions is what makes the system actually good.