[IS 4300] T8: Team Assignment #8 - Final Report and Interface
Finding Your Way

Erica Yee: yee.er@northeastern.edu
Maria Isabel Morel: morel.m@husky.neu.edu
Prisca Joseph: joseph.pr@husky.neu.edu
Project website: https://github.com/ericayee/hci/
Prototype: https://ericayee.com/hci/

Problem
Some of the most frustrating interface experiences are the ones that have no alternatives—so the user is forced to use them. Rather than create a brand new product, we wanted to address a user interface most students must use regularly: Northeastern's degree audit system, which displays a student's progress on the completion of their degree requirements. Each of us have had personal frustrating experiences with the degree audit system. Anecdotally from our friends and classmates, no one seemed to have overall positive experiences using the tool, though they could generally get the desired information with effort.

The current system is unintuitive because all the information is presented at once making it difficult to find information. For example, it's hard to navigate because without information hierarchies or clear section breaks, the text cannot easily be scanned. No instructions are built into the system; instead the "Audit Help" option takes users to a seperate FAQ site.

Users
The two groups that use the audit system are students and academic advisors. Students often use the system to plan their classes. Some students, particularly first-years or undecided majors, use the audit system to see the effects of changing their major or catalog year. Academic advisors work with students and the audit system to create a plan of study. Both students and advisors use the system to see the effects of changing a student's pattern of attendance, as some classes are only offered in certain semesters. Academic advisors and older students also use the system to make sure that a student is on track to graduate.

User goals are predominantly information retrieval tasks rather than information input. The tool to explore different majors does require user input, but it is not freeform. As such, we focused on how to best present the information in a way that is intuitive to navigate to facilitate tasks related to planning out course schedules for student users. Our personas of primary users included a freshman student who came in undecided and is trying to figure out which major she wants to pursue. A main task would be creating potential audits to see classes are required for different majors she's interested in. Another primary user was an upperclassmen with two semesters left before graduation. This user might need to determine what required courses he has left to take.

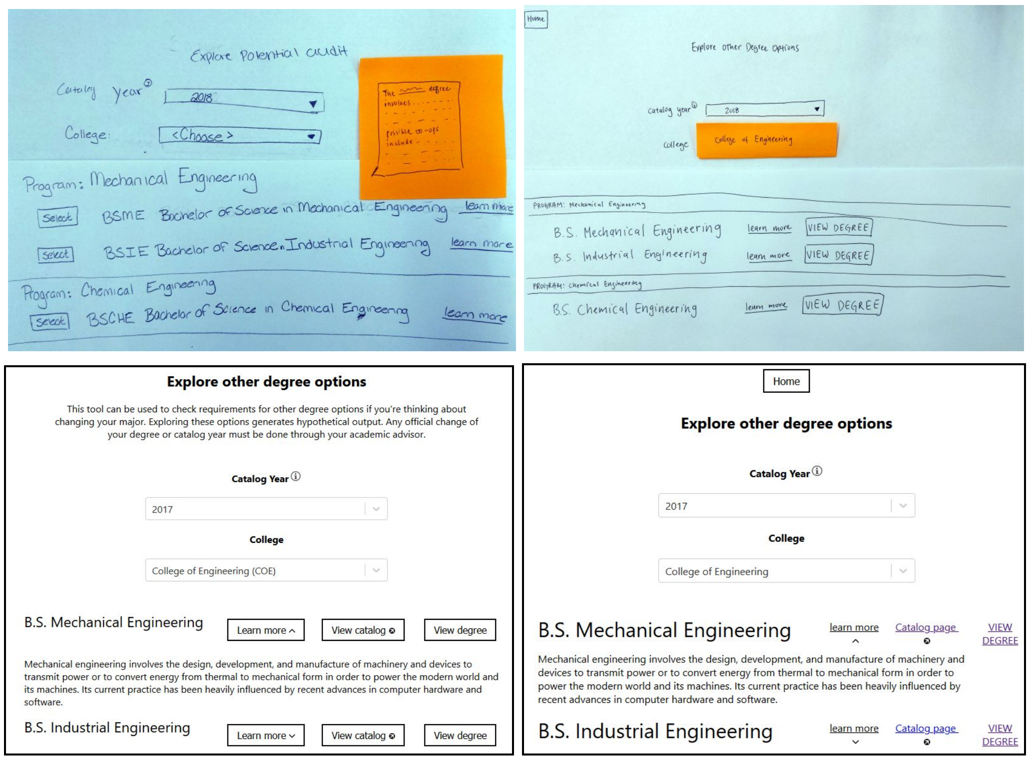
Design
Our final interface design has three main pages: Landing, Degree, and Explore.

When first opening the interface the user is met with a landing page. Below the introductory text are two buttons, one to view the current user's degree and the other to explore other possible degree programs. As a team we were adamant for keeping the page simple to prevent user errors and to keep what was absolutely needed for the system.

One of the main functions of the interface is to allow users to view their progress towards their current degree on the Degree page The page is broken up into a summary dashboard on top with information such as degree program, GPA, credits and current classes. When the user scrolls down or uses the side navigation menu, there are sections of tables indicating which requirements have been and still need to be fulfilled.

Finally, the “Explore other degree options” page presents the user with two dropdown menus through which to navigate lists of every degree program. Once a user chooses a college and possibly changes their catalog year, the listed programs each have three options: “Learn more” to expand/collapse a brief description, “Catalog page” that links to external school catalog, and “View degree” to see requirements. The first two options use arrow symbology to help users understand what will happen when they are clicked.

<u>Paper prototyping</u>


Explore Potential audit
Catalog year
2018
College:
<Choose>
The degree involves
possible co-ops include
Program: Mechanical Engineering
Select
BSME Bachelor of Science in Mechanical Engineering
learn more
Select
BSIE Bachelor of Science in Industrial Engineering
learn more
Program: Chemical Engineering
Select
BSCHE Bachelor of Science in Chemical Engineering
learn more
Home
Explore other Degree Options
catalog year
2018
college
College of Engineering
PROGRAM: Mechanical Engineering
B.S. Mechanical Engineering
learn more
VIEW DEGREE
B.S. Industrial Engineering
learn more
VIEW DEGREE
PROGRAM: Chemical Engineering
B.S. Chemical Engineering
learn more
VIEW DEGREE
Explore other degree options
Catalog Year
2017
College
College of Engineering (COE)
B.S. Mechanical Engineering
Learn more
View catalog
View degree
B.S. Industrial Engineering
Learn more
View catalog
View degree
Home
Explore other degree options
Catalog Year
2017
College
College of Engineering
B.S. Mechanical Engineering
learn more
Catalog page
VIEW DEGREE
B.S. Industrial Engineering
learn more
Catalog page
VIEW DEGREE

This tool can be used to check requirements for other degree options if you're thinking about changing your major. Exploring these options generates hypothetical output. Any official change of your degree or catalog year must be done through your academic advisor.

Mechanical engineering involves the design, development, and manufacture of machinery and devices to transmit power or to convert energy from thermal to mechanical form in order to power the modern world and its machines. Its current practice has been heavily influenced by recent advances in computer hardware and software.

Mechanical engineering involves the design, development, and manufacture of machinery and devices to transmit power or to convert energy from thermal to mechanical form in order to power the modern world and its machines. Its current practice has been heavily influenced by recent advances in computer hardware and software.

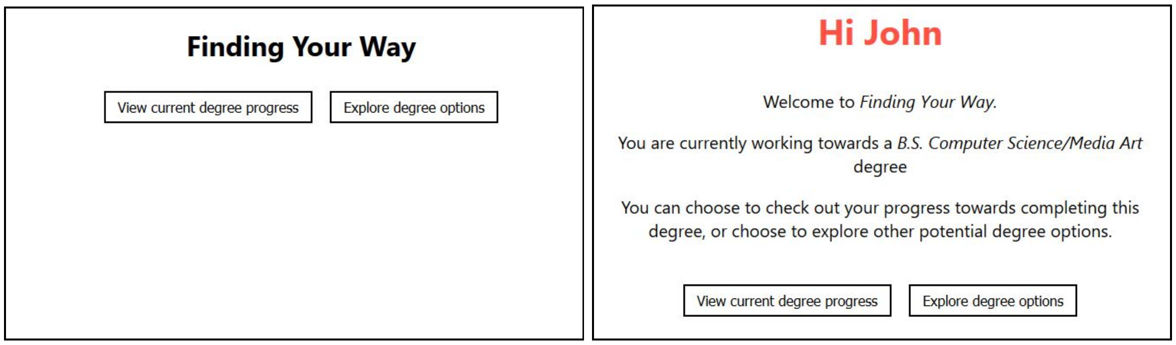
Iterations of Explore page throughout the design process (clockwise starting from the top left)

The initial in-class paper prototyping was enlightening because we coincidentally teamed up with a group of graduate students. As undergraduates, we assumed every Northeastern student had at least used the system with an advisor, so would come to our redesign from that baseline. However, the graduate students do not use the same system. Because they came to our interface with no familiarity, we quickly learned that even the word “audit” was confusing. Our task wording in the briefing also led them to believe that they were using the system to search for classes in general, like one would in the university-wide course catalog. This feedback led us to spend significant time rethinking the language of the system, not just layout. If our interface was implemented for the school, then new first-year undergraduate students would not immediately know what “audit” meant either unless told by an advisor. Consequently, we decided to use the more colloquial terms “degree progress” and “course requirements”.

Our second round of paper prototyping elicited feedback that our updated designs were helpful. Participants generally liked the interactivity of our system, such as the button navigation, drop down menus and tooltips. Still, there were consistent asks for more definitions of school terms, which we addressed with tooltips. We worried about the presence of too much text, but the computer prototype evaluators affirmed that they liked having all of these definitions. Tooltips are unobtrusive, but easily triggered to most users without leaving their current page.

One paper prototype evaluator wanted to use a back button from the Degree page. He said he had a mental diagram of pages he had already visited and where he wanted to return to. Still, he was able to find and successfully use the “Home” button to complete his task. If that user was testing our computer prototype, he would’ve been able to use the browser back button, so we did not implement a redundant back button. Based on feedback from a heuristic evaluator, however, we made sure the “Home” buttons were consistently located on the Degree and Explore pages.

<u>Heuristic evaluation</u>
One heuristic evaluator thought the Landing page was unnecessary and added to user memory load because it didn’t present any information other than the two buttons. He recommended adding a default state of either of the other two pages and then linking to the other option. We decided to keep the landing page but have it orient users more so they could make an informed choice of which page to navigate to. Thus, we added some welcome language with the user’s name and an explanation of the two options.


Finding Your Way

View current degree progress | Explore degree options

Hi John

Welcome to *Finding Your Way.*

You are currently working towards a *B.S. Computer Science/Media Art* degree

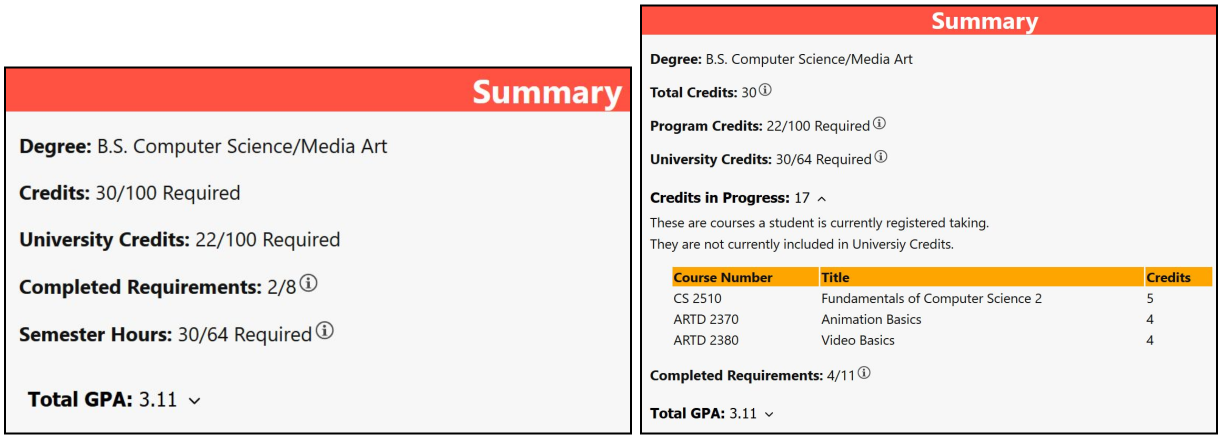
You can choose to check out your progress towards completing this degree, or choose to explore other potential degree options.

View current degree progress | Explore degree options

Landing page before and after feedback from heuristic evaluation

In the design sketches stage, we had two main debates for the degree page: whether to include a fixed side navigation menu, and whether the sections should be expandable/collapsible. We decided to include the menu for efficient navigation because the user is always one click away from any category. In paper prototyping, we noticed that most of our participants never touched the menu even though the section titles matched the task they were executing. This could’ve been because the menu did not look “clickable” on paper. However, we got positive feedback about the menu in the computer prototype. One evaluator said it “can help users go to the target sections quickly,” which validated our rationale for keeping it. We decided to not implement expandable/collapsible sections because of the extra clicking it would necessitate. Even though all the information is on a single long page, no evaluator noted that there was too much scrolling.

The summary dashboard also required iteration. We decided in the design sketches stage to display it prominently at the top, but decided against making it fixed. Both the paper prototype and heuristic evaluators were with the placement of the information, but consistently needed more help and documentation. We added more tool-tips that explained entries including “Completed Requirements” and “Semester Hours.” One heuristic evaluator pointed out the major issue of not being able to easily complete the task of seeing how many credits he currently is taking—he had to sum his current courses himself—so we added a “Credits in progress” entry. We clarified the difference between "Credits" and "University credits" using tooltips. Lastly on the Degree page, we implemented sorting of requirements by completion status on each table based on feedback.



Summary

Degree: B.S. Computer Science/Media Art

Credits: 30/100 Required

University Credits: 22/100 Required

Completed Requirements: 2/8 ⓘ

Semester Hours: 30/64 Required ⓘ

Total GPA: 3.11 ⌄

Summary

Degree: B.S. Computer Science/Media Art

Total Credits: 30 ⓘ

Program Credits: 22/100 Required ⓘ

University Credits: 30/64 Required ⓘ

Credits in Progress: 17 ⌃

These are courses a student is currently registered taking.
They are not currently included in Universiy Credits.

Course Number	Title	Credits
CS 2510	Fundamentals of Computer Science 2	5
ARTD 2370	Animation Basics	4
ARTD 2380	Video Basics	4

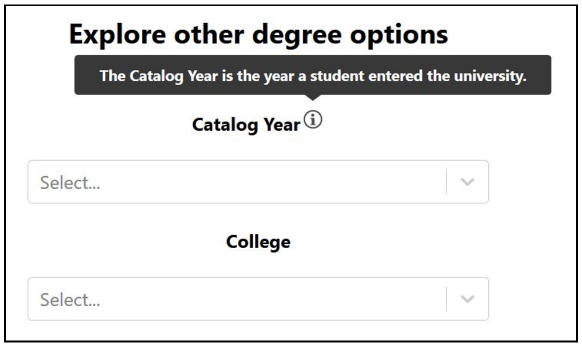
Completed Requirements: 4/11 ⓘ

Total GPA: 3.11 ⌄

Summary dashboard on Degree page before and after feedback from heuristic evaluation

The lack of help and documentation was also a theme in the evaluations of the Explore page. To address these issues, we added a brief explanation above the initial dropdown menus of the page’s purpose. We heeded a suggestion from a heuristic evaluator to minimize user memory load and clicking by defaulting the “Catalog year” dropdown to the most common use case, which is the year a student entered the university. Evaluators were confused about why a student would want to choose a different catalog year than the default. There are legitimate reasons to explore other catalog years, which we clarified in the tooltip.

An idea for a minor, yet surprising, change came from a heuristic evaluator who liked how he could search by college acronym in the drop down menu. This was not actually the case in the computer prototype; it only seemed to work because of a coincidence with the evaluator’s search term. However, adding the acronyms (i.e. CAMD) made searching by the familiar college names fully possible, so that the system matched the real world and spoke the user’s language.



Explore other degree options

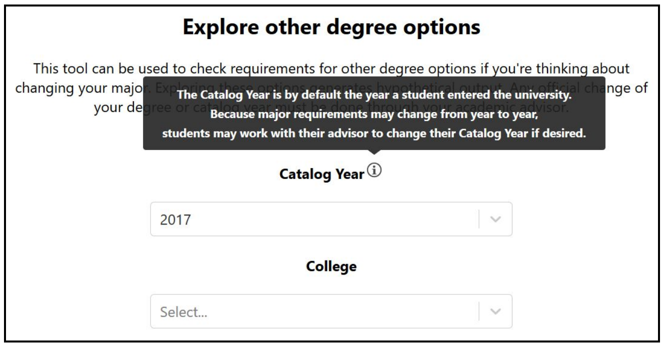
The Catalog Year is the year a student entered the university.

Catalog Year

Select...

College

Select...



Explore other degree options

This tool can be used to check requirements for other degree options if you're thinking about changing your major. Exploring these options generates hypothetical output. Any official change of your degree or catalog year must be done through your academic advisor.

The Catalog Year is by default the year a student entered the university.
Because major requirements may change from year to year,
students may work with their advisor to change their Catalog Year if desired.

Catalog Year

2017

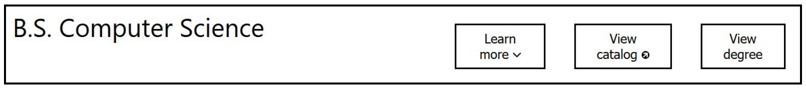
College

Select...

Explore page description and tooltip before and after feedback from heuristic evaluation

One evaluator wanted the degree program descriptions visible by default with the option to close them. However, we decided to keep them closed by default with the reasoning that if the system was showing every possible degree choice for that college, a page with every description open would be too crowded. Our prototype only shows a few example listings, but a fully implemented system might show dozens of degrees for each college.

<u>Implementation</u>
We implemented our prototype in React. We chose this JavaScript library because it is component-based, in which each component manages its own states and can render differently depending on the current state. Because each component can be as small as a button, it's possible to customize reusable components as building blocks to make up the total UI. For example, when a user chooses a college from the dropdown menu on the Explore page, the page will render components for every program in that college based on a JSON file.



An example of a rendered DegreeListing component

One catastrophic problem pointed out by a heuristic evaluator was that refreshing the browser tab crashes the site if not on the home page. React is usually used for developing single-page applications, which our host GitHub Pages doesn't natively support. Our workaround was to add redirects for subpages (i.e. the Degree and Explore pages) using a combination of the React Router library and these scripts. This is not the best way to implement React applications, but short of switching hosting platforms in the middle of the project, we deemed it best in this case. With this workaround we were able to mitigate feedback delays. Users should not notice anything amiss beyond the page quickly blinking in and out on refresh.

<u>Evaluation</u>
Through class assignments, we received feedback from two types of evaluations: participant-based user testing at the paper prototype stage and expert-based (if classmates count) usability inspection at the computer prototyping stage. Given more time and resources, our next step would be to continue user testing with the updated computer prototype. Cooperative evaluation or co-discovery could be particularly helpful here since all undergraduate students must use this system. Students already informally teach each other how to use school systems based on personal experience, so observing pairs interacting with

the prototype together may elicit feedback we would not get from solo interviews. Additionally, we would have liked to have academic advisors test the interface, as their task needs could differ from students.

This user testing would revolve around usability requirements we outlined in T2, along with some additional metrics. Our initial measures were efficiency—that a user will be able to find a desired requirement section in under 10 seconds from opening the application—and learnability — that after completing an initial standardized task, a user would be able to complete the same task with a different input without errors. The latter relates to the usability objective of a system meeting needs for infrequent or intermittent use, which is often how students use the degree audit system in real life. Because all our evaluations with experts and participants were conducted in single sessions, it's difficult to know how they would use the interface when coming back to it after some time away. Maybe they wouldn't need all the same tooltips now that they have internalized the term definitions.

Drawing on what we've learned in the readings and our design process, we would also now add some kind of survey to administer after user testing sessions. The survey would include rating scales for satisfaction, which measures overall usability, and ease of learning, which measures learnability. This quantitative feedback would complement insights gleaned from observing and interviewing participants.

Finally, we've brainstormed at least one way in which the third evaluation type of data analytics could be useful for our project. Since this is a school system, we're not interested in search engine optimization or referral methods. However, it would be useful to know what devices and browsers students use to ensure the system is optimized for all users. Analytics could also help us learn if, for example, students don't seem to be spending time on the Explore page. Do they have no need for it, or is the design confusing and thus turns them off? More than automated analytics, though, it is clear from anecdotal experience in this class and as Northeastern students that our peers have opinions about required school systems. To continually evaluate how useful these systems are and iterate on them in the future, there must be some kind of persistent outlet, such as a feedback hub, for comments on the deployed app.

Reflection

It may have been helpful to create and test a variety of paper prototypes. After the design sketches stage, we stuck to one main layout and rearranged elements within that framework based on feedback. However, it is difficult to know if that layout was the way to go. If we could've shown testers completely different layout designs, we might've been able to make a more informed choice. Additionally, we may have wanted to create at least one hi-fi prototype using wireframing software, Figma.com, Sketch, etc, instead of jumping straight from paper to computer prototyping. A hi-fi version would help ensure that all group members envisioned and agreed on design elements and functionality that were hard to capture on paper, such as navigation menus and the scroll experience.

Ultimately, the design process and feedback from evaluators helped us rethink a current system from scratch. Even if it's never officially implemented, we hope our prototype demonstrates—as we learned—how any interface can be improved with iteration.