Schedule Booklet Revamped
Project Report

Team BDI

Brice Truong
Baboya Choch
Ryan Bourdais
Mark Abrahams
Katherine Perez

Table of Contents

Introduction

Schedule Booklet Revamped is a scheduling tool for LSU students. These students require a simple scheduling tool that can make scheduling courses easy and prevent students from making costly mistakes when scheduling courses. This tool was built as a complete replacement for two current LSU systems, the Schedule Booklet and the Course Catalog, and to take on part of the functionals of the current Personal Request

Our Schedule Booklet Revamped addresses multiple issues with LSU's current course scheduling system. One of the issues that we address is that the current system requires you to have both the schedule request page, course offerings/schedule booklet page, and general catalog page up on three separate tabs. We solved this problem with the Interactive Table which combines these systems into one. Another issue that we are addressing is that there is no way to search or filter specific courses in the current schedule booklet. We address this issue by providing three features to our interactive table which are the sortable headers, a filter sidebar, and a search bar for locating specific courses.

Our Schedule Booklet Revamped was designed to be intuitive, easily navigable, clean, and efficient. To achieve our design goals we added several features to serve as the core of our application like an interactive table, filter sidebar, schedule modal, and addition of much needed feedback all over the site. To make our product easy to learn, we used a similar design to the LSU's schedule booklet. Through making a new system that is familiar to current students we believe that transitioning to this new system would be easy and pain-free for current students to adopt, as well as intuitive to incoming students.

Final Product Design

By marrying all of our requirements, user experience goals, and design principles we were able to come up with an initial design from our vision. We presented this vision very early on in the form of a low fidelity prototype. However, the design fell very short in numerous ways. The filtering system was too confusing, too many features were not easily understandable (and worse not easily discoverable), feedback was lacking for too many actions leading to confusion for our users, the schedule modal was not fleshed out yet, and so much more. The result, a very surgical revisitation on our design choices which brought us to where the final product is today. Below we will break down the core components of this final product (searching, feedback, filtering, the interactive table, and schedule modal) while diving deeper into and explaining what interaction design principles influenced and contributed to the final result.

Searching

To begin, once on the site, the site loads completely blank. This forces the user to find out what actions need to be taken to display the necessary data. Because the site is so minimalistic at first, the interface easily nails discoverability and understanding. The simplicity makes it easy to discover the main mode of interaction, the two autocomplete search areas and a button. This then leads to an understanding of how to access the required info, course availability information. As mentioned before, we wanted to maintain familiarity, so this system is taken from the older design that is the current schedule booklet. Moreover, what we improved on from the old system here is that searching was allowed without the user even providing any data. This action is no longer allowed, we have created an interlock by forcing the users to make a “Semester/Year”’ and “Department” selection before being allowed to search. Another improvement we made is by replacing the current dropdowns, whose affordances and signifiers are not clear, with text areas with autocomplete. We felt autocomplete text areas, search boxes, make its affordances (primarily typing more clear), in addition to being better overall for the user to interact with when it comes to searching, an interaction type we wanted to nail. This core feature is one that we felt we nailed very early on and did not change very much in our design process throughout the project.

Feedback

When interacting with sites it is virtually a lost cause to expect all of our users to be able to perfectly utilize all of our components without issue. For this reason we made implementing feedback into our site a very important core feature. We decided on a few different ways to give feedback in our interactable components. For our dropdown and sorting, we decided to use an icon badge, such as an arrow to communicate if a dropdown can be opened or closed, and an up or down arrow for sorting. The use of icon badges would make discoverability and understanding of the interface easier for the user. To show if there is an error, such as if the user is looking at a semester that has already passed, we decided to utilize tooltips. We warn the user using a red info icon which first notifies them there is an issue and when hovered over gives a detailed explanation. This would help prevent the user from making errors by anticipating possible errors and warning the user about them. This design choice specifically was one made later on once we realized through our interviews that we were confusing our users about the reasoning behind the absence of certain actions . Moreover, To indicate if a button or box is clickable, we decided to highlight the clickable components and disable, gray out, non clickable components. This would give the user hints on what they can or cannot interact with. As mentioned before, a lot of the feedback mentioned above was the result of interviewing our users. However, very early on we noticed the old system never actually provided the most basic of feedback when loading content on the page. We decided to solve this issue with the use of a skeleton that appears on “Search Courses” click. We used a skeleton screen during loading for the user to visualize the page before the loading is completed, letting them know their request is being processed—leaving no room for confusion on whether or not anything is happening. We designed our components initially with only functionality in mind. However, very quickly we realized where confusion may arise, so we added feedback in the form of skeleton animations, tooltips, helper/error text, and played around with a lot of colors to let it be known that there are errors, red, or the absence of something, gray. The result, a site with interactive components that meet our user experience goals of being easy to understand and navigate.

Filtering

To add functionality to the design of our main mode of interaction, the table, we decided on providing the user a number of filters to get all the classes they want according to their needs. These filters are located on the left side of the screen. These filters help the user to narrow down classes by showing them classes based on the course time, the course level, such as thousand or two thousand level classes, if the classes are full or not, and as well as the day in which these classes will be going on for. Our usability goal for the filter side bar was to provide a clear and efficient way to find classes. We seeked to achieve this goal by providing the aforementioned filtering options and implementing them as checkboxes when multiple options could be selected, radio buttons when only one option, and an interactive clock to select the time to select the classes. The interaction clock uses faded colors to show times that are not available, another form of feedback. Finally, to ensure that the users are always in control and to establish confidence, we provided a submit filters button so the changes selected will filter the table only when the user is finished applying all the filters they want.

Interactive Table

One of the biggest shortcomings of the old system was its interface for actually providing course data/information was very un-interactive because it was too simple. As a result, we decided to stick with a similar system, but added interactivity when necessary because one of our design goals was to stay familiar to the old system. To achieve this we stuck both with and closer to the idea of the table as a conceptual model. By following this conceptual model of the table, in web interfaces, we are able to build upon the old system by utilizing familiar and common interactions. Firstly, we implemented sortable headers as a mode to easily order entries in columns, either ascending or descending. However, while implementing this feature, we ran into issues with the discoverability of the feature itself because we found it challenging to make the affordances of the header, clicking, obvious. In addition to sortable headers, we had to solve the problem of finding a way to make more information (description, prerequisites, and more) available to users. To solve this problem we once more looked at the conceptual model of a table. This once more led us to another idea, utilizing collapsibles. By utilizing collapsibles, we were able to expose more information without also overwhelming them and while simultaneously staying true to the table conceptual model.

Moreover, although we had all these new ideas of how to make the table provide more without making it overwhelming by following the already established model of table, we had features that required us to think differently. This became evident when it came to finding an implementation for how to allow our users to add a course. In finding a solution we wanted to maintain simplicity and stay within the table, so to speak. That was to add an additional row for “Actions”, alongside the others with course information, that made available actions, via buttons, for adding a course and another that both reveals and hides the collapsible area we earlier spoke about. Our features mean nothing if the discoverability of them is terrible. Our goals to keep discoverability high made it an easy decision to choose to implement our “add class”(serves the purpose of the current schedule request page) and “more info”(serves the purpose of the current general catalog) actions as an extension of the table itself.

All in all, keeping with the idea of the conceptual model of the table and extending it with the addition of an extra column for actions allowed us to stick with our user experience goals of staying familiar, simple, and clean in addition to maintaining high discoverability for the new features. We achieved this all while most importantly still being able to meet requirements for combining the schedule request and general catalog into our new application without selling ourselves short when it came to meeting our overall experience goals.

Schedule Modal

The core function of this application is being able to build a schedule, as easily as possible, so it is important to keep the schedule itself at the front to make it both very accessible and discoverable. We brought the schedule to the literal front functionally to meet these experience goals and design principles. To do that we implemented the schedule as something similar to a modeless dialog box, a collapsable modeless box per se. A modeless box allows us to keep the schedule at the literal front, foreground, of our application while also not drawing away focus from the rest of our features on the screen. The result of this implementation is that we make it very easy to discover where one's schedule is and, to begin with, provide a mode for being able to get feedback on your schedule as you build it and manage it, drop and remove courses. Hence, we are able to meet both our functional requirements (provide schedule management like the schedule request page) while keeping alive our design principles and user experience goals of staying easily navigable, discoverable, clean, and simple.

Evaluation Methodology

For our evaluation we used the "Controlled Experiment" approach. We interviewed 4 participants in total, all of them are students at LSU. 2 are incoming freshmen, 1 is a senior majoring in business and the last is a graduate student. We gave each participant access to the website and allowed them to interact with the web page giving them one single instruction to get started : try to schedule some classes.

We collected the data by observing each user while they were performing the task. Throughout the experiment we asked questions in an attempt to try and understand why they did what they did, and provided aid whenever it was absolutely necessary to move on. We conducted an interview afterward to talk about their experience in depth, maybe rehashing some previously asked questions. We asked them a series of questions, why did they do what they did? What they thought a component should have done? What could be done better? How does it compare to the current scheduling system?

We compared the data from each interview and analyzed it by looking at similar problems each user ran into or what questions they had about certain components. We also compared these findings to our previous feedback from earlier interviews with different groups to see if we had improved.

Results

From this evaluation we were able to get some valuable feedback that showed us issues we had previously missed. After beginning their search for available courses the users all shared that the sizing of some of the elements in the courses table looked inconsistent. Some users shared that when the courses were added to the schedule modal they became small and hard to read. One user thought that the webpage could benefit from more feedback saying that, "the load is too instant", when searching the courses table. Only one participant in our evaluation noticed and interacted with the sortable header feature of the courses table. This is an issue that we had encountered in previous interviews.

Overall, when asked about how this scheduling method holds up against the one currently used at LSU, all participants agreed that it was more convenient and easier to navigate than the current booklet.

Discussion

The results of our evaluation have taught us many lessons about our system. For starters, our system was built without responsive web design in mind. As a result, differing resolutions and devices were difficult to support (once we get to smaller than tablets) and, resultantly, are not up to the standard of our primary designs. Another issue that came to light was inconsistencies with respect to sizing components, going forward we would definitely make sure to add more styling information to Figma and ensure that we stuck to the original design's styling as a team and communicated where issues arose. Sortable headers are functional for our system, but we had issues with making their functionality obvious. Going forward the sortable headers feature would be a piece that we would focus on to make sure it is more apparent to users what is done. Other changes that we would like to make long term are quality of life changes, such as increasing personalization for users dependent on their major and year; as well as making the autocompleted boxes more obvious so users can spend less time filling them out.

Overall, we were successful in meeting our target goals. We developed an interactive table that contains all of the information necessary to schedule courses, as well as implemented a filtering and searching system to make finding the proper courses easier. This system would fulfill our requirements well and greatly increase students' quality of life when it comes to scheduling courses for their semester. We believe that we were successful because of forethought. We planned every piece of our system thoroughly ahead of time and had most of our design ironed out before we began coding. Our organization and teamwork truly shined in this project and allowed us to make a working system, with little to no hiccups, on time.

Our advice to give someone building a similar project is to organize yourself early and plan every step of the way. Most projects fall flat or outright fail because of skipped steps, but because we made sure to do everything sequentially and not jump the gun at any point we were successful in this project.

The most unexpected part of this project was data collection. We scraped the original course offerings page from LSU's site and discovered that it would be a monumental task to clean up and make ready for our site. We discovered that currently the LSU course catalog receives a massive html file with every database call. This html file includes the entire catalog for the selected semester as one large string with "\n" and dashes to denote different lines. In order to make this data usable we first had to scrape this massive string, parse it into a csv file, then manually edit specific cases that could not be done programmatically, such as research courses or seminars.