Guelph Course Notifier: Report

Joshua Samuels (0960244)

School of Computer Science, University of Guelph

CIS 4300: Human and Computer Interaction

Joshua Samuels

Dr. Neil Bruce

December 18, 2020

Perceptual Considerations

Colour

Colour choice plays an important role in website design and using the "wrong" colours can hinder the usability and likeability of a website. The initial design for my website had a "dark-mode" colour-scheme, a popular choice among many frequent computer users. After further consideration, the dark-mode design was a great idea, however, inconsistent with the majority of websites and Webadvisor. Instead I decided to make the background stark white with a black text and medium blue title bar. Black text on a white background is the most legible colour-scheme and was complimented very well with the blue title bar. There are issues with a heavy reliance on blue, however, I do not believe the minority of users who cannot view blue properly will have any issues with the website due to the strong contrast between the colours.

Gestalt Principles

The gestalt principles have been applied in many aspects of web design. The gestalt principles are several principles about the human perception that help create intuitive designs. Although I did not utilize all the gestalt principles, I will explain the principles I used in my design.

Law of Symmetry. The law of symmetry is a gestalt principle that states "a visual object appears incomplete if it is not balanced or symmetrical" (Bruce, 2020). In my website design, I made sure all of the components were symmetrical. A lack of symmetry is one of the most difficult issues for a user to discover, however, causes frustration with a system.

Law of Proximity. The law of proximity states "Items placed near each other will be perceived as a group" (Bruce, 2020). There are many groups of information in the website I have created, including in the notification management dashboard, the information to register for a course, and the course information. Too many physical groupings (e.g. in a box) would cause the interface to feel extremely cluttered so I rely on the law of proximity for the items to be recognized as a group.

Attention

Humans have a limited ability to process information, called their attention. The two main forms of attention are passive, when an environmental change shifts a person's attention, and active, which is when a user is concentrating on a task. Although passive attention grabs are distracting there are several alerts built into the website when the user performs an invalid action or provides invalid data. An example lives in the Sign In button: When attempting to sign in with an SSO provider there is a dismissable alert notifying the user that SSO providers are not supported. The alert also informs the user to please use the guest sign in option.

I used several techniques to keep my user's attention focused on the website during their interaction. Firstly, I ensured to keep the website's design as minimalistic as possible so the user does not get overwhelmed with where they should spend their attention. The website attempts to abstract as much information from screens as possible, only providing the information required for a given task. Finally, I used neutral colours which are relatively standard on the web to prevent the user from subconsciously wanting to leave the website before they have lost their attention.

Memory

One of the most important things to remember when creating a website is that viewers can recognize and remember patterns better than they can recall specific details, commonly simplified to recognition over recall (Bruce, 2020). This principle explains why graphical interfaces are much easier to use than a command-line and why computers have become so easy to use. There are many recognizable actions on my website, such as the SSO via providers options, the title bar, and clickable application name in the top right to send the user home. When building my website, I used Miller's theory on short-term memory to ensure no groups on my website contained more than 7 actions (Bruce, 2020). Not only does Miller's theory improve the overall user experience but it also significantly speeds up the learning curve as well.

Goals

Instead of a "standard" home screen, which explains an overview of the website to the user I opted to make the home screen the course searching page. This decision improved learnability since the pages to register and manage notifications are dependent on the user finding classes they would like notifications for. Additionally, signing in to the application does not benefit the user until they have searched for a class and registered for notifications for that class. The intention was to start the user with a simple/error-prone and recognizable task, searching for a class, and gradually introduce the user to the website's features as they progress through the website, a design which also significantly improved the memorability of the application. Since new tasks were taught to the user gradually as they learned how the application worked, the website became much easier to learn. The website was also designed based on "industry-standard" where applicable, such as the program name in the top left sending the user back to the main page, a menu bar with actions and a single sign on flow common with many popular applications.

My primary user experience goal was usability of the system. The University of Guelph has thousands of classes, each of which containing information about scheduling, professors, exams, etc. To ensure the website was not overwhelming to the users I simplified the searching either by course or by name - ignoring all other fields for that section. The screen to register notifications was split in half to ensure a visual divide between the course information and notifications for the website.

It was also important for the website to be enjoyable and aesthetically pleasing, two goals in a convenience tool that go hand-in-hand. Material-UI helped turn my low-fidelity prototype into an aesthetically pleasing website by providing modern generic components that follow core design principles discovered at Google. In turn, I spent most of my time optimizing the website for user-experience instead of tinkering with CSS settings, which hopefully will create an enjoyable experience for the users of the application.

Users

The primary users of the website are students at the University of Guelph who are looking to be notified about certain course information. A student waiting for new slots to open in a course would use the system infrequently - going to the website to opt-in to notifications and then again to opt-out of notifications once they are no longer needed. Many students take 4-5 courses per semester and have some required meaning that these students would only watch and unwatch notifications a handful of times every semester. Since some courses post the exam information once course selection has occurred, diligent students who want to ensure there are no exam conflicts would also be a primary user of the website.

The secondary users of the website would be friends of the students who sign up for notifications that do not want email spam and will rely on their friends to inform them of the information. They will not be using the site directly, but may benefit from the information their friends are receiving through the website.

Tertiary users would include administrators at the University of Guelph who might reexamine their course selection windows based on the introduction of the application. Students may have also created simple applications similar to this one but may choose to use this website instead of maintaining their own application or may choose to create a competing application to this one.

Requirements

The first step I would take to gather requirements for the website is interviewing the primary users of the application. I would start with an unstructured interview of several friends of mine to gauge the general direction I should take when building the searching and notification options in the website. With a brief idea of the website's functionality I would create a semi-structured interview with random students (on Zoom or once the campus reopens) to gauge the needs of students around course selection while reducing the bias from my social circles.

The second step to gather requirements would be through a questionnaire targeted at a large group of students. I could reach a large number of students by posting the questionnaire in university Facebook groups and emailing the classlist of the classes I am in (and several of my friends). As an incentive to complete the survey I could raffle 5-10 $5 Amazon gift cards to students who complete the survey - a relatively low price to pay for requirements gathering. The questions asked would be broad but only in multiple choice format to ensure good data is retrieved. The questions would help keep the website simple while still ensuring it solves a problem students have.

Finally, once a working prototype of the application has been developed I would observe users interacting with the website. Since a dozen multiple choice questions would just be used to guide my design, observing how prospective users use my system would help validate the assumptions I made from the survey results and create a better system for the users.

Although in a real-life situation I would obtain some requirements from users, I have created a list of assumed requirements as the basis of my design:

- Searching should be done by course code or name
    - Most students can recall at least one of the course code or course na,e
- Students are most concerned with the following course notifications
    - A course they are interested in taking fills up (moves to 0 available slots)
    - A course they are interested in taking opens up (moves above 0 available slots)
    - The university of Guelph updates exam information for a course
- Students want full control over the notifications they receive
    - They do not want to feel as though they are being spammed
- For this website to be useful users must feel they are notified in real-time
    - Notifications must be sent within 2 minutes of the change

Tasks

What different tasks does the system perform and what scenarios have you considered?

Hierarchical Task Analysis

Creating a notification

0. To create a notification for CIS*1050

1. Select the course CIS*1050

    1.1. Select the Course Code dropdown and type "CIS*1050" and select the option

    1.2. Select the Course Name dropdown and type "Web Design & Dev" and select the option

2. Select the "Search button

3. Check the boxes for the relevant information

    3.1. Check the box to retrieve notifications when a course becomes available

    3.2. Check the boxes to retrieve notifications when a course becomes unavailable

    3.3. Check the boxes to retrieve notifications when a course's exam information is updated

4. Enter your email address

5. Click the Title in the top right corner

6. Select Sign Up for notifications

Plan 0:

- Do 1-2-3-4-6 in that order
- Do 5 to search for a new course without saving notifications

Plan 1:

- Do 1.1 or 1.2 depending on if the course code or name is easier to recall

Plan 3:

- Do 3.1, 3.2, 3.3 in any order depending on the notifications wanted
-

Toggling off a notification

0. Toggling off a notification

1. Click the sign in button

2. Sign in as Guest

3. enter the email address with saved notifications

4. enter the verification code sent to your email (anything that is non-empty for the prototype)

5. Select the Manage Notifications option from the title bar

6. Locate the notification to toggle off

7. Toggle the switch from on to off

Plan 0:

- Do 1-2-3-4-5-6-7 in that order

Scenario

Josh is a 5th year Software Engineering student at the University of Guelph. His course selection window opened at 8am but since he did not get up until 11:00am was unable to enroll in an elective he had wanted to take his entire undergraduate degree. Josh observed that there were only 10 sections in the course, a very small number for any university. Josh discovered there is no option for webadvisor to notify him when his course becomes available and has class all-day so he is unable to be constantly refreshing Webadvisor. Josh does a quick google search and comes across "Guelph Course Notifier" an application promising to email him when his course becomes available within 5 minutes. Josh is intrigued so he finds his course by course code in the drop down and selects the search button. He signs up for notifications when the course becomes available and sees the option to be notified when exam information is updated so opts-in to those notifications as well.

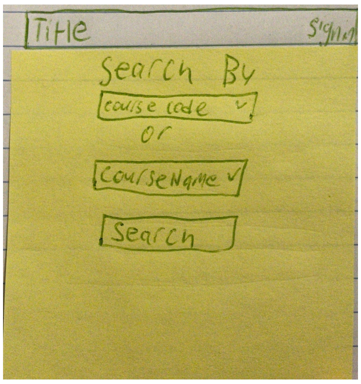
Josh was skeptical at first but eventually got stuck in a debugging rabbit-hole and forgot about his course worries. Later that day, Josh got a gmail notification from his phone and it was none-other than Guelph Course Notifier notifying him that there are 4 available spots in his course. He logs into webadvisor and sees there is 1 spot remaining and registers for the course. Josh is happy and refers Guelph Course Notifier to all his classmates.

Prototyping

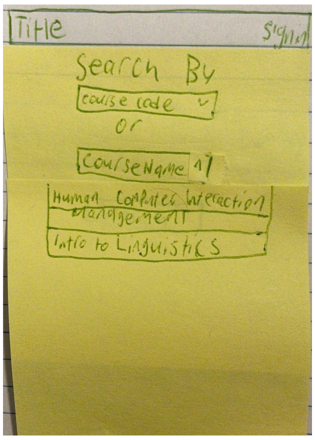
I used low-fidelity prototypes before developing my website. I have had ideas for this website for a few months so I saw a high-fidelity prototype as a barrier to my coding of the website - especially once I completed a low-fidelity prototype. The low fidelity prototype allowed me to iterate quickly and once I decided on a layout I hopped into coding (with a few *small* changes).

Searching for a course

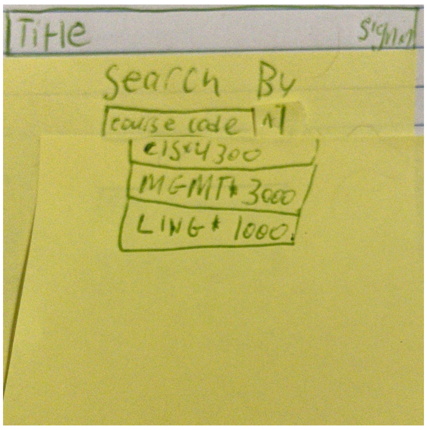
Homepage



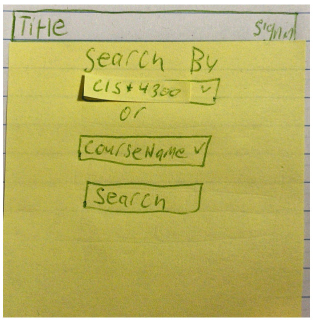
Search by Course Name



Search by Course Code

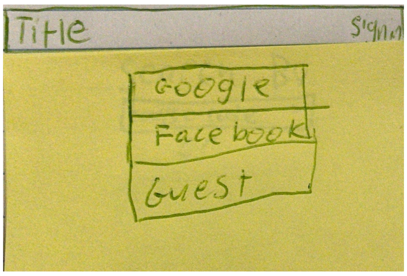


Select CIS*4300 from dropdown

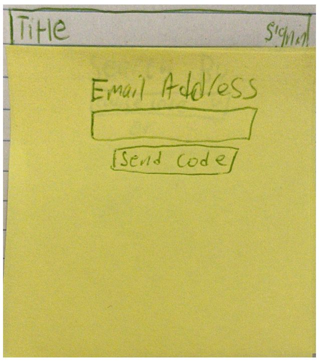


Sign In

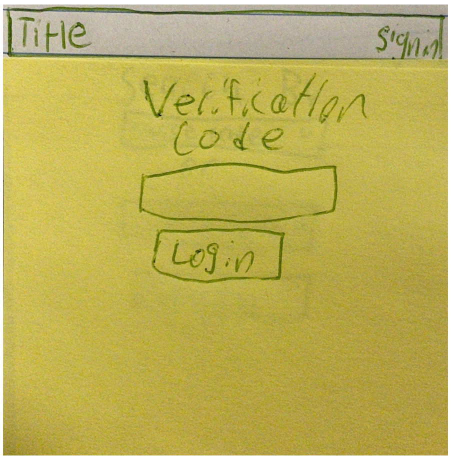
SSO Provider pop-up



Enter email address (to sign in as Guest)



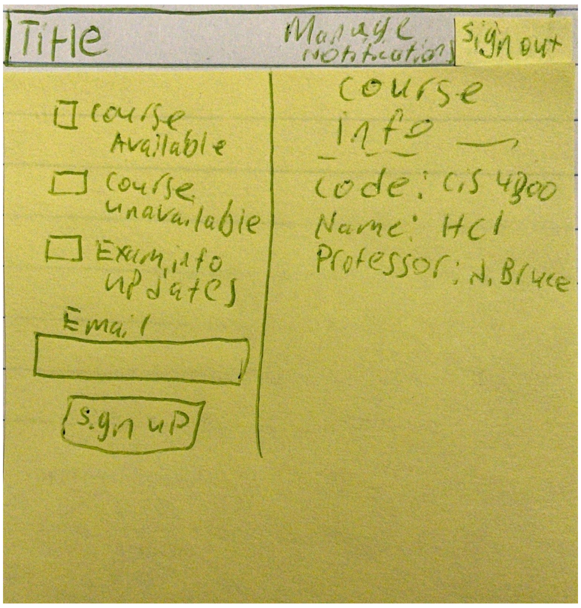
Enter the verification code sent to the email address



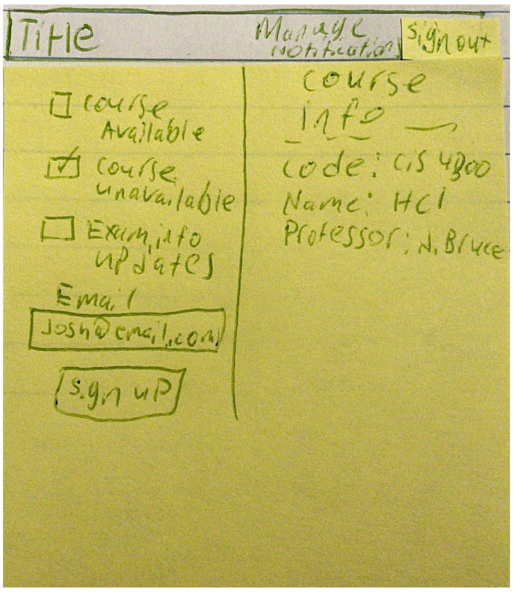
Register for course

Note: Search for course must be executed first

Register Dashboard

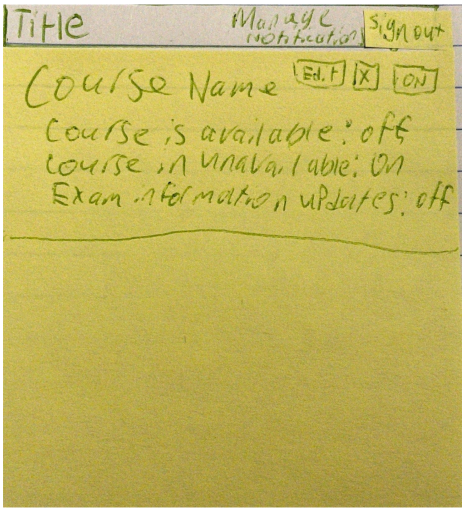


Select notifications and enter email



Course Management

View existing notifications



Course is available: off
course in unavailable: On
Exam information updates: off

Layout

I used a hierarchical layout within the webpage to ensure the title bar is always at the top of the webpage. Since users have the ability to sign in from any screen and the only other action in the title bar is to go back to the homepage, it made sense to have the title bar at the top of every page. On the notification registration page I used panels to split the screen in half, presenting the course information right and the user actions/input on the left. Due to the minimalistic design I have employed, the webpage is dynamic to many screen sizes, allowing users to grow or shrink the web page as they please without losing access to any features or information.

Events, Input and Errors

The only events aside from user-driven is the responsiveness of the webpage. The website is responsive to window events via the struts and springs layout. When resizing windows on the website the empty space on the page will be consumed and pushed onto a new line when applicable. If the screen is too small for a single component to be rendered, a scrollbar appears to view the entire component.

Since there is no backend connected to the website, the only other events in the website are user-driven input events via their keyboard and mouse. The system minimizes room for error by using mouse-only inputs when possible, such as switches and checkboxes. The system also has an autocomplete textbox (with a built in dropdown) to limit user error when searching for courses. Errors that do occur in the system are handled elegantly. Since there is no backend to support SSO providers, when selecting an SSO provider that is not supported (any options except guest), a dismissable alert informs the user that their selection is unavailable. Standard textboxes are only used sparingly and in each instance have validators to catch and inform the user of invalid data before progressing through the app.

Evaluation

Design is a crucial aspect to a website. The perfect application, created with a horrible design becomes a horrible application that would cause more problems for users than it would solve. I have personally come across many applications with a bad design which leads me to search for competing applications, even with less functionality, that have a great design.

The first criteria to evaluate my design on is its functionality. I believe the application is a good minimum viable product for a webadvisor notifying application. There is room to expand the functionality, as with most applications, but I do believe the core functionality is sufficient to solve a big problem for a large number of students. I believe the functionality matches that of the user's expectation since a more condensed application (less steps to achieve the same results), would lead to a cluttered and less clear interface; I strived to create an interface that is easy to use for new and experienced users.

It is also important to evaluate the design on user experience, especially for new users since all users must be new at some point and without a simple interface will not continue to use the website. The core of the website is a single task, creating notifications for courses. Although the management dashboard and login are more difficult to find, a new user would search for a course and intuitively add notifications for that course, a simple aspect to learn. Only once they have become familiar with that aspect would they notice the ability to sign in and manage their notifications. I truly believe the application is easy to learn and understand for new users - my primary goal in the design of the interface. I also believe the users will be satisfied with the system since it is visually appealing and will help them focus on work without the constant worry of webadvisor information changing under their feet.

Finally, it is important to discuss the problems with the current interface to ensure they do not supersede the benefits it provides. Currently the biggest issue with the application is the lack of SSO providers. The website is not powered by a backend which makes it impossible to currently support any SSO providers, however, many users would be hesitant to constantly check their email for a sign in code with each guest login. Another potential area for confusion is the current lack of notification consolidation. One user could save the same or different notification settings to their email address for a single course. Although seemingly simple on the surface, the notification consolidation complexity left me with not enough time to implement that feature. Finally, a possible problem could be the presence of the email prompt when saving a notification even when logged in. The intention behind that was to allow a user to let their friends sign-up for notifications on their account, however, there is room for confusion by end users.

Although not a perfect website, I believe the solutions this website provides significantly outweighs the problems. It would also be very beneficial to create an evaluation checklist guided by the DECIDE framework or an expert evaluation of the system.

Experts

Users are usually the most difficult and expensive method to evaluate a design. Although important, users should not be introduced very early in the process to minimize the cost and stress of finding ample users. Cognitive walkthroughs from experts is a great way to receive feedback before users see the application. In this assessment, a set of scenarios and actions would be outlined on the webpage, it is the experts job to assess whether the actions are intuitive and to fix the unintuitive design decisions. This step is critical in minimizing user confusion and improving the intuition in the design.

It would also be valuable to receive a heuristic evaluation from a team of experts, since up to 80% of usability problems are discovered by only 5 experts. The steps in heuristic evaluation would include an introduction to their tasks, an evaluation period of the system, and a session for the experts to discuss and prioritize the problems discovered.

It is important for multiple experts to see the system in action, however, their evaluation is not infallible, and must be used as a precursor or along-side user testing to create a user-friendly system.

User Testing

User testing is very important with all applications. No matter how competent the designers, developers and experts are it is inevitable for issues to arise in the users of the application. The user testing would focus heavily on the time it takes to discover and complete tasks within the system as well as the source for their errors and confusion. Once they have completed the testing it would also be important to focus on the user's goals and questions they have about the system. A satisfaction questionnaire is another great way to obtain insight on the users experience, including issues they may not have been able to describe. The questionnaire is a great source of leading and open-ended questions to get the best information out of each user.

Questionnaire

Questionnaires are a great tool to obtain feedback from users. Similar to a website, a questionnaire must have a well-thought out design to optimize its value to all stakeholders in the application. A well-designed questionnaire has the ability to prompt users for valuable feedback they may not have otherwise provided. It is also a great tool to make conclusions from and save to revisit at a future date from a large number of people. The best questionnaires have a variety of questions covering factual information about the interviewee and the system, opinions about the system, and their attitude towards the system. The right users are usually difficult and expensive to find, so a questionnaire that can be widely distributed is more likely to contain lots of valuable data than other forms of user input, unless a pre-existing pipeline exists. User questionnaires are a great tool for retrieving information at scale, but not without their drawbacks. Users can have trouble explaining issues they have encountered with the system, limiting their effectiveness in the results. Their results are also very subjective and likely only have a small picture of the system rather than the whole picture.

With all this information I have decided to provide a sample of a questionnaire I would create to obtain feedback on Guelph Course Notifier.

Name: ___________

Major: ___________

Years completed: 1 2 3 4+

Gender: Male Female Decline to answer Other: ________

Please check the box that corresponds with your satisfaction for the following questions:

Question	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I am satisfied with Webadvisor					
I get into all my classes without worry					
I believe there are tools that can improve my course selection experience					
I check webadvisor at least once a day around the dates of my course selection window					
Guelph Course Notifier solves at least one problem I have during course selection					
Guelph Course Notifier was an easy tool to learn					
I will use Guelph Course Notifier once it is completed					
If I could change one thing about Guelph Course Notifier it would be:					
I would like to see more notification					

options in Guelph Course Notifier (please specify)	

Effort

While developing the website for this project I decided to challenge myself with a multitude of new tools. Aside from a single small component at my last co-op, I have never developed in React, nevertheless created my own webpage from scratch. It was a difficult but rewarding task to build a web page in React and I was excited to defeat the challenge slowly over the last month. I also use FontAwesome and Material-UI for icons and styling. Although FontAwesome was trivial to learn, Material-UI was a difficult framework to customize but well-worth the learning curve. I am hoping to build out a backend over the break to connect to this webpage!