CS-639 Building User Interfaces, Fall 2020, Professor Mutlu

React 2 β (3 Points)

Improving Usability Using Heuristic Evaluation

In this assignment, you will put the ten usability heuristics we learned in class into practice toward improving the usability of your *React 2 β* deliverable. You will focus on specific components of your design, identify potential violations of the heuristics, make design recommendations to address these violations, and implement recommendations that are feasible to create a new deliverable. Use this opportunity to make concrete design decisions about your project, to improve your design using the heuristics, and to build a keen eye for identifying usability issues as a UX developer.

Step 1—Identify A Focus. (0.2 Points) Review your *React 2 β* deliverable with a critical eye to identify 3–5 "components" that you think are most consequential for user experience.

Step 2—Review the Heuristics. Review the ten usability heuristics we discussed in class from the slides, what principle each heuristic represents, and examples of the violations of the heuristics.

Step 3—Identify Potential Violations. (1.0 Points) Focusing on your components, inspect your design, considering each usability heuristic, for any violations of the heuristics.

Step 4—Develop Design Recommendations. (0.4 Points) For each violation you identified in the previous step, provide a design recommendation for addressing it, assessing its feasibility.

Step 5—Implement Your Recommendations. (1.4 Points) Implement the design recommendations that you identified as "feasible" in the previous step in your prototype, updating your design.

Submission Details

GitHub Classroom Starter Code

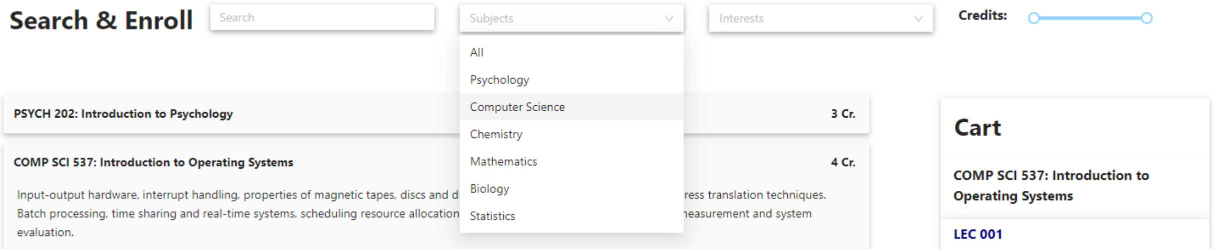
React 2 β will build on your implementation of React 2 α. You should copy your code from your React 2 α project to the React 2 β repository linked above, as that will be your starter code. When you commit and push, ensure that you are committing and pushing to the react2-beta repository, not react2-alpha.

To complete the assignment, you will need to submit a completed version of this document as PDF to Canvas. In addition, you will submit your repository name and latest commit hash from GitHub Classroom, e.g. react2-beta-ctnelson1997, 2b0ef83.

Step 1. Identify A Focus. (0.2 Points)

In this step, you will review your *React 2* ☒ deliverable with a critical eye to identify 3–5 "components" that you think are most consequential for user experience and that you will put under the microscope of heuristic evaluation in the next step. In real life, your application might have hundreds of components, screens, or pages, and you will have to focus your efforts on a limited set that will make the most difference in terms of effectiveness and user experience. Similarly, you will review your design and identify 3–5 components to focus on. Here, a "component" can be the entire page/view (e.g., recommended courses) or a reusable component (e.g., the course component, the rating component), but not something as small as a button or label. Provide screenshots of each component below and provide a brief justification (1–2 sentences) of why you think each one is a critical component.

1. Search and filter: Enable users to navigate through potential thousands of courses, which would be a common feature if there are a lot courses.



PSYCH 202: Introduction to Psychology 3 Cr.

COMP SCI 537: Introduction to Operating Systems 4 Cr.

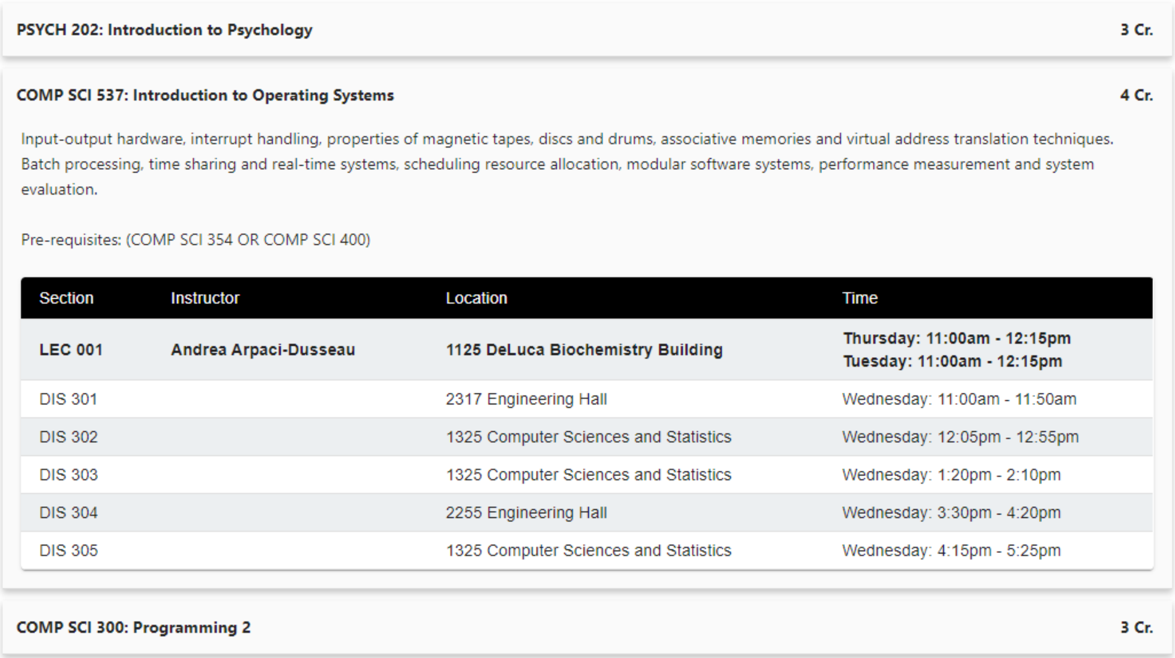
Input-output hardware, interrupt handling, properties of magnetic tapes, discs and d ress translation techniques. Batch processing, time sharing and real-time systems, scheduling resource allocation neasurement and system evaluation.

Cart

COMP SCI 537: Introduction to Operating Systems

LEC 001

2. Course accordion: Allow users to view details and add courses to cart. Hence would be the key feature for the application.



PSYCH 202: Introduction to Psychology 3 Cr.

COMP SCI 537: Introduction to Operating Systems 4 Cr.

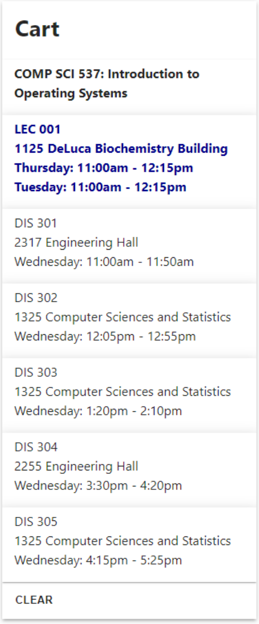
Input-output hardware, interrupt handling, properties of magnetic tapes, discs and drums, associative memories and virtual address translation techniques. Batch processing, time sharing and real-time systems, scheduling resource allocation, modular software systems, performance measurement and system evaluation.

Pre-requisites: (COMP SCI 354 OR COMP SCI 400)

Section	Instructor	Location	Time
LEC 001	**Andrea Arpaci-Dusseau**	**1125 DeLuca Biochemistry Building**	**Thursday: 11:00am - 12:15pm** **Tuesday: 11:00am - 12:15pm**
DIS 301		2317 Engineering Hall	Wednesday: 11:00am - 11:50am
DIS 302		1325 Computer Sciences and Statistics	Wednesday: 12:05pm - 12:55pm
DIS 303		1325 Computer Sciences and Statistics	Wednesday: 1:20pm - 2:10pm
DIS 304		2255 Engineering Hall	Wednesday: 3:30pm - 4:20pm
DIS 305		1325 Computer Sciences and Statistics	Wednesday: 4:15pm - 5:25pm

COMP SCI 300: Programming 2 3 Cr.

3. Cart of added courses: After the user added the courses to cart, it's natural for the user to review what they added, or if they added multiple sections, they might want to take a look at which particular section they want to take.



Cart

COMP SCI 537: Introduction to Operating Systems

LEC 001
1125 DeLuca Biochemistry Building
Thursday: 11:00am - 12:15pm
Tuesday: 11:00am - 12:15pm

DIS 301
2317 Engineering Hall
Wednesday: 11:00am - 11:50am

DIS 302
1325 Computer Sciences and Statistics
Wednesday: 12:05pm - 12:55pm

DIS 303
1325 Computer Sciences and Statistics
Wednesday: 1:20pm - 2:10pm

DIS 304
2255 Engineering Hall
Wednesday: 3:30pm - 4:20pm

DIS 305
1325 Computer Sciences and Statistics
Wednesday: 4:15pm - 5:25pm

Step 2. Review the Heuristics.

Carefully review the ten usability heuristics we discussed in class from the slides, what principle each heuristic represents, and examples of the designs that violate and support the heuristics. Below is a cheat sheet for Nielsen's ten heuristics that you can use in the next step. (This step does not have any deliverables.)

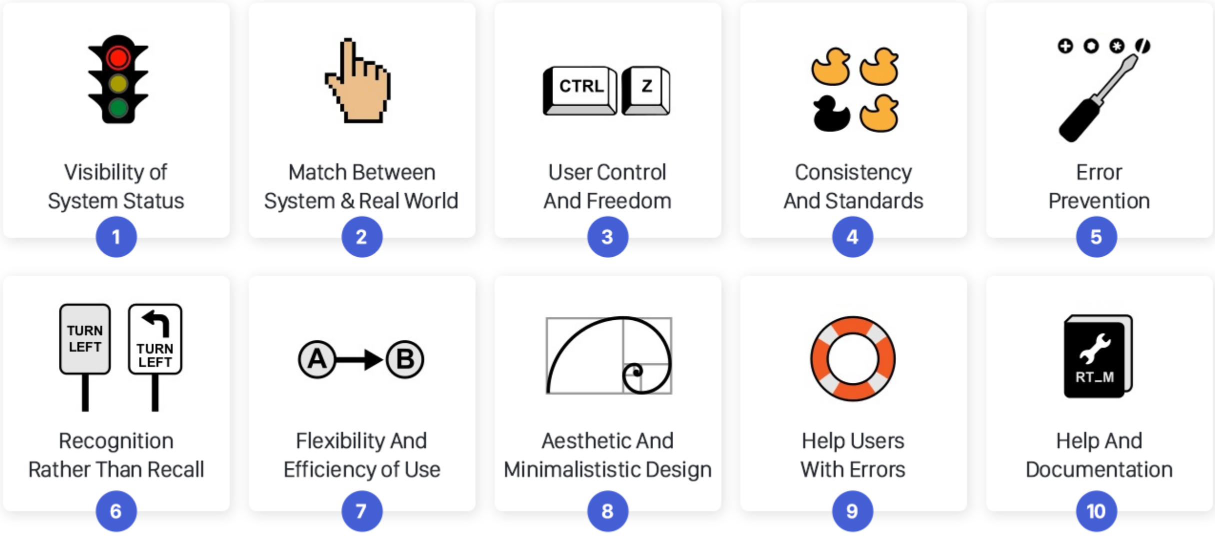

Visibility of System Status
1
Match Between System & Real World
2
CTRL
Z
User Control And Freedom
3
Consistency And Standards
4
Error Prevention
5
TURN LEFT
TURN LEFT
Recognition Rather Than Recall
6
A
B
Flexibility And Efficiency of Use
7
Aesthetic And Minimalistic Design
8
Help Users With Errors
9
RT_M
Help And Documentation
10

Image source: UX Collective

Step 3. Identify Potential Violations. (1.0 Points)

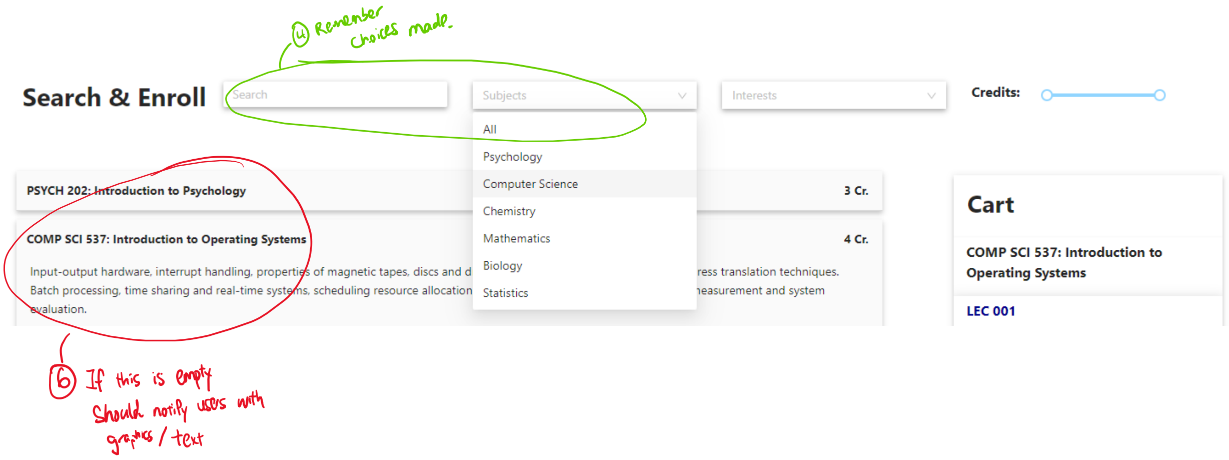
Focusing on your components, inspect your design, considering each usability heuristic, for any violations of the heuristics. For each violation, use the following table to briefly describe the violation and give it a unique number (specified in the # column). Make copies of your screenshots from Step 1, focusing on the design elements you are considering in this step, and mark them with the unique numbers so that the reader of your report can find the location of the violation in the screenshots and read your description in the table below. In addition, color-code the violations for severity, highlighting with red, orange, yellow, green, and gray for the severity-rating scale we covered in class (with red being most severe to gray being a non-issue), while cyan denotes heuristic considered from previous implementation.

Heuristic	#	Search and Filter	#	Sections Accordion	#	Cart
Visibility of system status		User can see changes immediately after applying filters	1	No hints to notify user courses are successfully added		Users can see changes immediately after removing
Match between real world & system						
User control & freedom					2	No undo button if user accidentally remove from cart.
Consistency & standards		Unified design language implemented across the application		Unified design language implemented across the application		Unified design language implemented across the application
Error prevention			3	Still allow user to add courses that did not meet pre-req to cart		
Recognition rather than recall	4	Users have to retype search word every time they use the platform.			5	Courses added to card are not saved
Flexibility & efficiency of use						
Aesthetic & minimalist design		Minimalistic design with shadows as affordance of clickable content		Highlight when mouse hover over sections as affordance of clickable		Minimalistic design with shadows as affordance of clickable content
Help users with errors	6	If search result returns empty or credits go out of bound, no error message				
Help & documentation	7	None provided, user might get confused to use platform	7	Users might get confused on their possible action	7	Users might get confused on their possible action

Step 4. Develop Design Recommendations. (0.4 Points)

For each violation you identified in the previous step, provide a design recommendation for addressing it along with an indication of whether or not it is feasible to implement the recommendation as an extension of your *React 2* ⊠ deliverable. (Only recommendations that are beyond the capabilities we learned in class or beyond the scope of the project should be marked as not being feasible.) Order the table of recommendations based on the severity of the usability problem from most severe to least severe. Use the table below to describe your recommendations, adding additional rows as needed, and follow the same color-coding from the previous step for severity ratings.

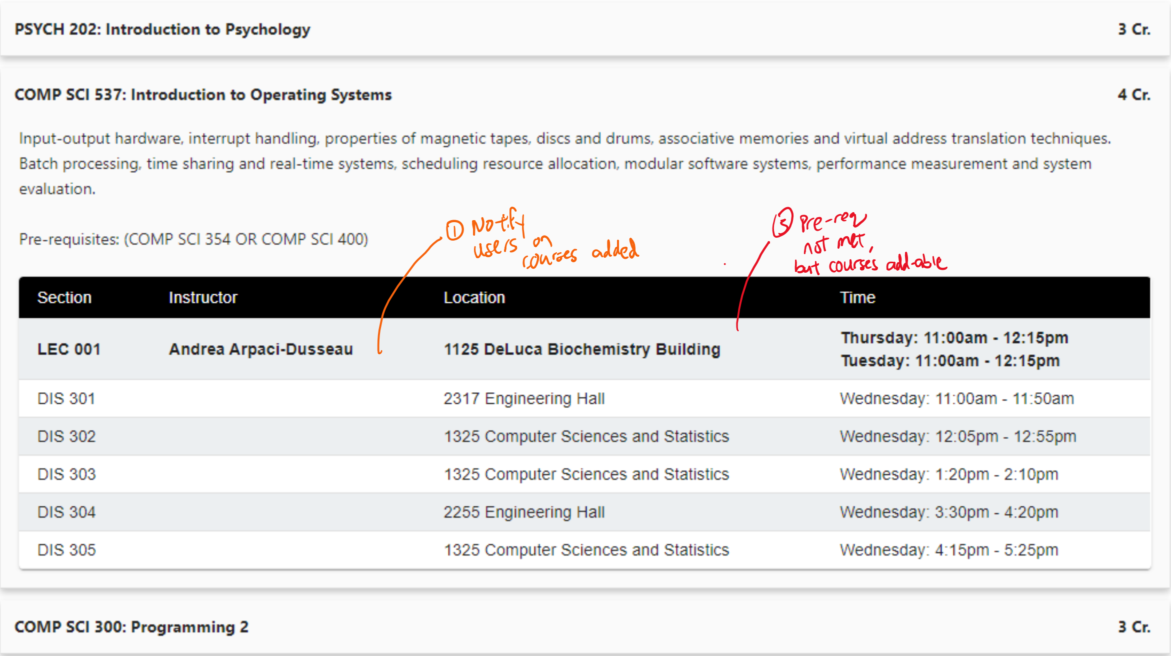
#	Recommendation	Feasibility (Yes/No)
3	Add notifications to notify users pre-req not met	Yes
6	Add icons to notify users on result returning none	Yes
1	Add notification to notify users courses are successfully added	Yes
2	Add an undo button allowing users to revert the last changes made	Yes
5	Remember choices added to cart by users	No, require change in API
4	Remember choice made by users on filters	No, require change in API
7	Add support and documentations	Yes



Input-output hardware, interrupt handling, properties of magnetic tapes, discs and d… …ress translation techniques. Batch processing, time sharing and real-time systems, scheduling resource allocation… …neasurement and system evaluation.

④ Remember choices made.

⑥ If this is empty should notify users with graphics / text



PSYCH 202: Introduction to Psychology **3 Cr.**

COMP SCI 537: Introduction to Operating Systems **4 Cr.**

Input-output hardware, interrupt handling, properties of magnetic tapes, discs and drums, associative memories and virtual address translation techniques. Batch processing, time sharing and real-time systems, scheduling resource allocation, modular software systems, performance measurement and system evaluation.

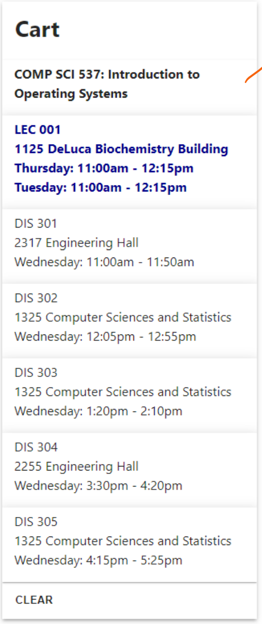
Pre-requisites: (COMP SCI 354 OR COMP SCI 400)

Section	Instructor	Location	Time
LEC 001	**Andrea Arpaci-Dusseau**	**1125 DeLuca Biochemistry Building**	**Thursday: 11:00am - 12:15pm** **Tuesday: 11:00am - 12:15pm**
DIS 301		2317 Engineering Hall	Wednesday: 11:00am - 11:50am
DIS 302		1325 Computer Sciences and Statistics	Wednesday: 12:05pm - 12:55pm
DIS 303		1325 Computer Sciences and Statistics	Wednesday: 1:20pm - 2:10pm
DIS 304		2255 Engineering Hall	Wednesday: 3:30pm - 4:20pm
DIS 305		1325 Computer Sciences and Statistics	Wednesday: 4:15pm - 5:25pm

COMP SCI 300: Programming 2 **3 Cr.**

① Notify users on courses added

③ Pre-req not met, but courses add-able



Cart

COMP SCI 537: Introduction to Operating Systems

LEC 001
1125 DeLuca Biochemistry Building
Thursday: 11:00am - 12:15pm
Tuesday: 11:00am - 12:15pm

DIS 301
2317 Engineering Hall
Wednesday: 11:00am - 11:50am

DIS 302
1325 Computer Sciences and Statistics
Wednesday: 12:05pm - 12:55pm

DIS 303
1325 Computer Sciences and Statistics
Wednesday: 1:20pm - 2:10pm

DIS 304
2255 Engineering Hall
Wednesday: 3:30pm - 4:20pm

DIS 305
1325 Computer Sciences and Statistics
Wednesday: 4:15pm - 5:25pm

⑤ Courses added are not saved

② No undo button on accidental removal of courses in cart

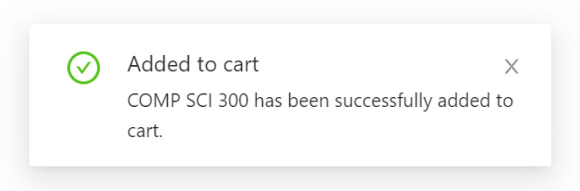
Step 5. Implement Your Recommendations. (1.4 Points)

In this step, you will implement the design recommendations that you identified as “feasible” in the previous step in your prototype, updating your design. To receive full points, you will implement at least three design recommendations that can improve one or more of the components you focused on. Submit your improved React project based on instructions below and provide a paragraph that summarizes the outcome of the heuristic evaluation. In this paragraph, reflect on how your design improved, what you learned about usability in the process of applying the heuristics, and whether you gained any unexpected insights about your design.

Your deliverable will be a completed version of this document, attached to the canvas assignment as a PDF, and the GitHub Classroom repository name and latest commit hash.

The changes I made includes:

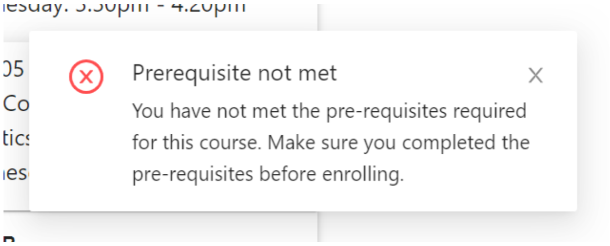
(1) Prompt notifications on course successfully added to cart



Added to cart

COMP SCI 300 has been successfully added to cart.

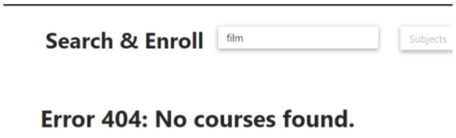
(5) Prompt notification to notify user to add courses that do not meet pre-req to cart.



Prerequisite not met

You have not met the pre-requisites required for this course. Make sure you completed the the pre-requisites before enrolling.

(6) Added headings on no courses found.



Search & Enroll

film

Subjects

Error 404: No courses found.

My design has become more user friendly. The user can now understand more on what has happened – Notifications to notify users when a course is added to the cart and when users added a course that they did not meet pre-req of to the cart. An error message was also shown on screen when users' filter yield no results. With this it might be less confusing for novice users to navigate around the user interface to perform the task they intended. Overall, it allowed the user to better understand their actions and what can and cannot be done.

I found out that the heuristic approach really guided me to finding ways to improve on my design. Sure some of the choices make sense during implementation, like sections numbers from the API, but these guides create a comprehensive outlook for system designers to make sure they are no blind spot that they might have missed.

Alerting the users was something I figured out when testing my website on previous project (i.e. improve design by thinking out loud). For the things that I did not implement but made suggestions above, things like add color to prompt users what are added and what are not are also blind spots that I missed.