CS-571 Building User Interfaces, Spring 2022, Professor Yuhang Zhao

React *β* (7 Points)

Reimagining Layout & Navigation, Improving Usability Using Heuristic Evaluation

In this assignment, you will explore the concepts we learned in the lecture, titled "*Interaction Design: Structure, Layout, & Navigation*". Using the principles and components covered in class, you will redesign your implementation of React ***α***. Then, you will put the ten usability heuristics we learned in class into practice toward improving the usability of your redesign. Use this opportunity to make concrete design decisions about your project, to improve your design using usability heuristics, and to build a keen eye for identifying usability issues as a UX developer.

Part 1—Redesign: (2.5 Points) In this part, you will analyze your current solution for the *React* ***α*** Assignment in terms of its layout and navigational elements. Then, you will use the principles and components covered in class to redesign your solution and describe your design choices.

Part 2—Implementation: (2.5 Points) In this part, you will implement your new design by extending your implementation for the *React* ***α*** Assignment using additional React and/or Bootstrap components.

Step 3—Heuristic Evaluation. (2.0 Points) Review your implementation from Part 2 with a critical eye to identify 2-3 "components" that you think are most consequential for user experience. Focusing on your components, inspect your design, considering each usability heuristic, for any violations of the heuristics.

Submission Details

GitHub Classroom Starter Code

This assignment assumes that your React β will be built on your implementation of React α. If you choose to do so, remove everything in the starter code and copy the files from your React α project to the React β repository above. However, we are providing a starter code for students who could not complete implementing React α. So, feel free to work on this assignment based on the starter code if you'd like. When you commit and push, ensure that you are committing and pushing to the *react-beta-s22* repository, not *react-alpha-s22*.

To complete the assignment, you will need to submit the following on Canvas:

1. A completed version of this document as PDF as an attachment.
2. Your repository name and latest commit hash from GitHub Classroom, e.g. `react-beta-s22-osori, 7a0bc38`, as a comment.

Part 1. Redesign (2.5 Points)

(0.2 Points) **Step 1. Analyze Layout.** Describe the current layout of your *React α* implementation, identifying at least *two* elements of layout design (e.g., golden proportion, visual hierarchy, visual scan patterns) it currently follows. Take a screenshot of your implementation and annotate the principles you identify, briefly (2-3 sentences) briefly justifying why parts or all of your implementation follow these principles.

<insert-your-annotated-screenshot-here>

(0.2 Points) **Step 2. Analyze Navigation.** Consider your *React α* implementation, what navigation model(s) does it use? Below, draw the navigation model that your implementation follows the same way navigation models were described in class.

<insert-your-navigation-model-here>

(0.6 Points) **Step 3. Conceptual Redesign.** In this step, you will reimagine your *React α* implementation, such that it uses a different set of navigation models, and/or principles of layout design. Your goal should not be to change your implementation for the sake of changing it, but consider ways in which the structures and layout and navigation principles might improve your implementation. Your conceptual redesign should involve the use of at least one layout principle, make at least one change in the navigation model, and introduce at least one element/aid to improve navigation. Provide a hand-drawn or digitally created (e.g., in Adobe XD) mock-up of your design below. Annotate your design to describe your design choices, highlighting the specific principles you employed.

<insert-your-annotated-design-here>

(1.5 Points) **Step 4. Detailed Redesign.** In this step, you will build on your mock-up to create a detailed design, determining image, color (for background and elements), type, size, icons, and so on (as we also did, to some extent, in the JavaScript β Assignment). Provide a digitally created mock-up (e.g., in Adobe XD) that shows your design choices. Annotate your mock-up to describe your design choices.

<insert-annotated-mock-up>

Part 2. Implementation (2.5 Points)

(0.5 Points) **Step 1. Inspect Library Elements.** In this step, you will inspect the standard React component library, the react-bootstrap component library, and/or an alternative that you are comfortable working with to see how you can realize the detailed design you created in the previous part using these components. You are not expected to change the library components to exactly match your design choices, but to identify which component elements might best meet your design goals. Below, copy the design and the choices you generated in Part 1 and annotate them to describe which components from the library you will use to accomplish your design goals.

<include-your-annotated-design/choices-here>

(2.0 Points) **Step 2. Implement Redesign.** The last step of this part will involve implementing the design improvements you described in Part 1, using the layout and components you described in the previous step. You can use standard React components, Bootstrap components, and/or an alternative library in your implementation. You do not have to implement new *functionality*; focus on implementing your *design*.

<include-screenshot(s)-of-final-implementation>

Part 3. Heuristic Evaluation (2.0 Points)

(0.2 Points) **Step 1. Identify a Focus.** In this step, you will review the implementation of your redesign from Part 2 with a critical eye to identify 2-3 "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 2-3 components to focus on. Here, a "component" can be the entire page/view (e.g., completed 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.

<component-screenshots-and-justifications>

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.)

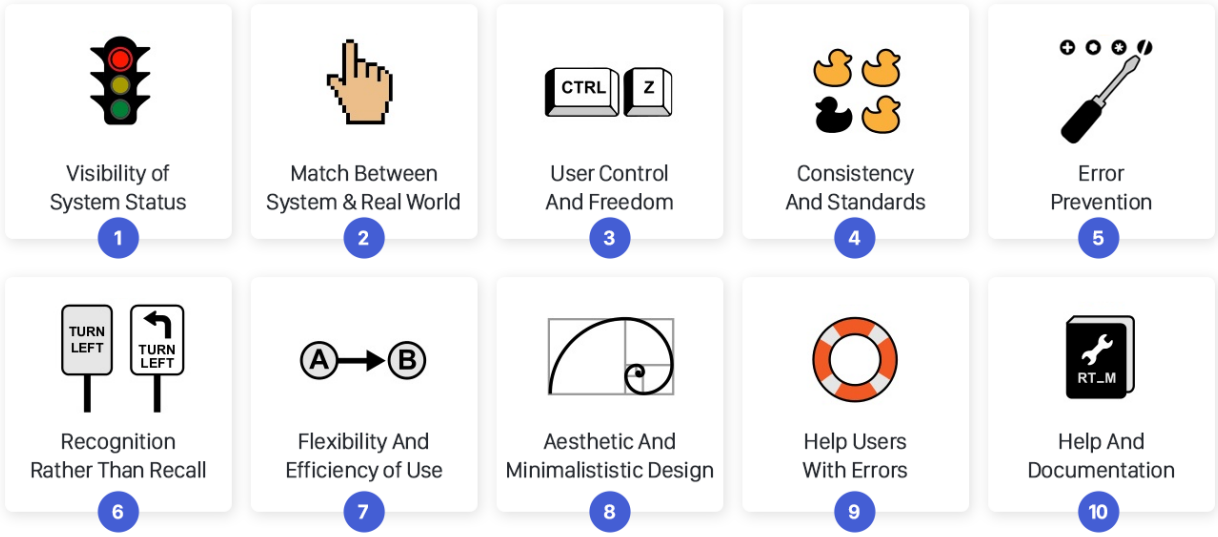

Image source: UX Collective

(1.8 Points) **Step 3. Identify Potential Violations.**

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 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 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). For each component, you will likely note violations of some of the heuristics but not others. Only highlight violations in the table below and in the screenshots, and heuristics that are not violated can be left blank.

Heuristic	#	Component 1	#	Component 2	#	Component 3
Visibility of system status						
Match between real world & system						
User control & freedom						

Consistency & standards	
Error prevention	
Recognition rather than recall	
Flexibility & efficiency of use	
Aesthetic & minimalist design	
Help users with errors	
Help & documentation	

<screenshots-marked-with-numbers>

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.