Improving Usability Using Heuristic Evaluation

This handout is provided for *informational purposes only*. **There is no deliverable for this.**

By following this document, you will focus on specific components of your design, identify potential violations of the heuristics, and consider design recommendations to address these violations. 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. Review your work with a critical eye to identify 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. Focusing on your components, inspect your design, considering each usability heuristic, for any violations of the heuristics.

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

Step 1. Identify A Focus.

In this step, review your work with a critical eye to identify 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 and you will have to focus your efforts on a limited set that will make biggest difference.

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.
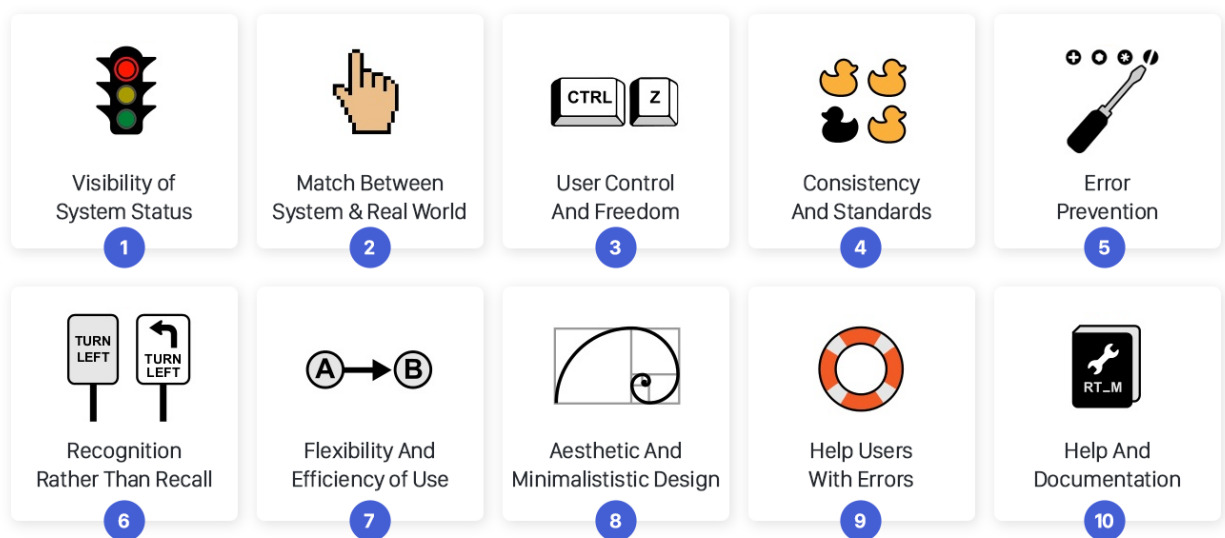

| Visibility of System Status **1** | Match Between System & Real World **2** | User Control And Freedom **3** | Consistency And Standards **4** | Error Prevention **5** |
| Recognition Rather Than Recall **6** | Flexibility And Efficiency of Use **7** | Aesthetic And Minimalististic Design **8** | Help Users With Errors **9** | Help And Documentation **10** |

Image source: UX Collective

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 briefly describe the violation and give it a unique identifier (specified in the # column). You may annotate your components using these unique identifiers, visually showing the violation. 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).

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						

Heuristic	#	Component 4	#	Component 5	#	Component 6
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						

Step 4. Develop Design Recommendations.

For the unique identifier of each violation that you identified in the previous step, you may provide a design recommendation for addressing it. The table of recommendations should be ordered 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