

Prototype

One sentence description of the UI you are evaluating

Overall Recommendations

1-2 paragraphs of overall recommendations you have for improving the user interface given what you read about their user tasks and what you experienced in testing their prototype (1-2 paragraphs).

Summary of Violations

Category	# Violations
B1: Recognition rather than recall	
B2: Aesthetic & Minimalist Design	
B3: Match between system and the real world	
B4: Consistency and Standards	
F1: User control and freedom	
F2: Flexibility & efficiency of use	
P1: Visibility of system status	
P2: Error prevention	
P3: Help users recognize, diagnose, and recover from errors	
P4: Help & documentation	
Total violations	

List of Heuristic Violations

List the heuristic violations you identified, describe the problem and rationale for why you think it violates the heuristic, and provide a suggestion for how to fix it. For example:

1. **B4 Consistency & Standards**

The interface used the string “Save” on the first screen for saving the user’s information, but used the string “Store” on the second screen. Users may be confused by this different terminology for the same function.

Fix: Use “Save” on all screens.

2. **F1 User Control & Freedom**

The interface brings the user into a set of preference screens when they select “New User”, but doesn’t allow the user out of the dialog until they fill out all four screens. There is no way to cancel from any of the screens if a user came into the first screen by accident.

Fix: Add a “Cancel” function to each screen in the sequence.

Evaluation Heuristics for Reference (note these are only examples, not the only definitions!)

Minimize cognitive burden

- B1. **Recognition rather than recall**: Minimize the user’s memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- B2. **Aesthetic and minimalist design**: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- B3. **Match between system and the real world**: The system should speak the user’s language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- B4. **Consistency and standards**: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Maximize user freedom

- F1. **User control and freedom**: Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- F2. **Flexibility and efficiency of use**: Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Handle problems gracefully

- P1. **Visibility of system status**: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. Help the user know whether the “right thing” happened.
- P2. **Error prevention**: Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
- P3. **Help users recognize, diagnose, and recover from errors**: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

P4. **Help and documentation:** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.