

1. **Visibility of system status:**

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Feedback indicates this is not an issue.

2. **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.

Interview indicates that it is very easy to use, and we always use real world non technical language.

3. **User control and freedom:**

Users often choose by mistake and will need a clearly marked "emergency system function exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Home button and back buttons present throughout. No lock ins.

4. **Consistency and standards:**

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Aesthetic and functionally consistent.

5. **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.

Input validation on all fields, and confirmation messages.

6. **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.

Booking confirmation etc displays all relevant information.

7. **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.

Simple system so not relevant.

8. **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.

Yes.

9. **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.

Custom HTTP error message to take you home, and input error messages

10. **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.

User manual - should make it more accesible.