Heuristic Evaluation on

Anthony's design

Evaluator Name:

Nagare Negishi

SEVERITY RATING

0 = I don't agree that this is a usability problem at all

1 = Cosmetic problem only: fix if time is available

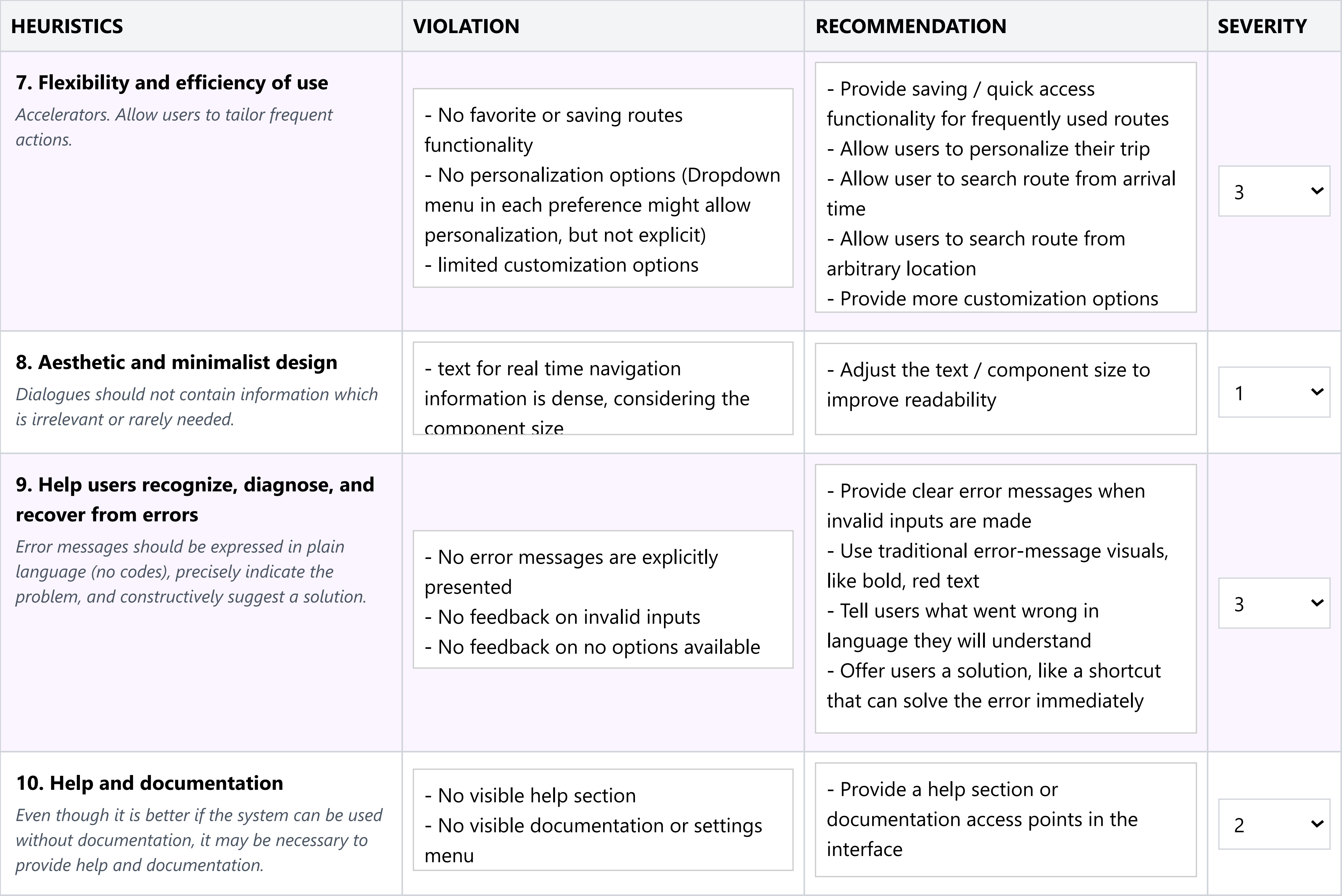
2 = Minor usability problem: fixing this should be given low priority

3 = Major usability problem: important to fix, given high priority

4 = Usability catastrophe: fix this before product can be released

HEURISTICS	VIOLATION	RECOMMENDATION	SEVERITY
1. Visibility of system status *Always keep users informed about what is going on, through appropriate feedback within reasonable time.*	- No loading indicators when searching for routes or locations - No scroll indicators to show the user where they are	- Add loading indicators when searching for routes or locations - Add scroll indicators show the user where they are	2
2. Match between system and the real world *Follow real-world conventions, making information appear in a natural and logical order.*	Describe the violation...	Provide recommendations...	0

HEURISTICS	VIOLATION	RECOMMENDATION	SEVERITY
3. User control and freedom *Users should leave the unwanted state without having to go through an extended dialogue. Undo and redo.*	- No visible "back" button - No "Undo" option (While not necessary for presented task)	- Add a "back" button to all screens for easy navigation - Add cancel/undo options	3
4. Consistency and standards *Users should not have to wonder whether different words, situations, or actions mean the same thing.*	Describe the violation...	Provide recommendations...	0
5. Error prevention *Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.*	- The search field doesn't appear to constrain invalid inputs - Assuming the search field accepts name of the location, emojis are likely to be invalid	- Improve the search field capabilities to prevent invalid inputs - Implement functionality to suggest valid options from user input in the search field - Disable or hide emoji option from the keyboard	3
6. Recognition rather than recall *Minimize the user's memory load by making objects, actions, and options visible.*	- User must know the name of the station to search for it	- Favourite / recent history options to reduce memory load - Destination suggestions based on user input	2



HEURISTICS	VIOLATION	RECOMMENDATION	SEVERITY
7. Flexibility and efficiency of use *Accelerators. Allow users to tailor frequent actions.*	- No favorite or saving routes functionality - No personalization options (Dropdown menu in each preference might allow personalization, but not explicit) - limited customization options	- Provide saving / quick access functionality for frequently used routes - Allow users to personalize their trip - Allow user to search route from arrival time - Allow users to search route from arbitrary location - Provide more customization options	3
8. Aesthetic and minimalist design *Dialogues should not contain information which is irrelevant or rarely needed.*	- text for real time navigation information is dense, considering the component size	- Adjust the text / component size to improve readability	1
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.*	- No error messages are explicitly presented - No feedback on invalid inputs - No feedback on no options available	- Provide clear error messages when invalid inputs are made - Use traditional error-message visuals, like bold, red text - Tell users what went wrong in language they will understand - Offer users a solution, like a shortcut that can solve the error immediately	3
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.*	- No visible help section - No visible documentation or settings menu	- Provide a help section or documentation access points in the interface	2