**Human-Computer Interaction**

Jakob Nielson’s 10 general principles of usability heuristics for interactive design were applied in an original analysis of the current code journal’s user interface. Below shows the list of heuristics and the identified features and issues which are explained with brief justifications. Suggestions for improvements were also provided for the issues identified.

|  |  |  |
| --- | --- | --- |
| **List of Heuristics** | **Identified Problem or Feature** | **Brief Justification** |
| **#1: Visibility of system status** | User reminders display the access permission – e.g. ‘You may not see the Upload section if you are not an author’. | Users can know the permission they have in the journal. |
| User statistics are displayed in the sidebar on the ‘Home’ page. | The number of posts, journals, comments, and likes are reported to the user statistics on the sidebar on the ‘Home’ page so users can get detailed reports on engagement and comprehensive insights into publishing behaviour. |
| Loading icon for page transitions and time to query from the recommendation system. | Indicates that the system is in a transition state and is loading a new page (loading folders and video). Reassures the user that a longer wait is normal, and that the system is still working. |
| Button changes colour on hover – clear feedback on the user’s action. | The colour change of a button communicates that the system is working and reduces uncertainty. It prevents users from tapping the same button multiple times. |
| **#2: Match between system and the real world** | User-centric language such as ‘Your Posts – relates to the user in first-person pronoun. | The user of first-person pronouns such as ‘my’ and ‘me’ relates to the user on a personal level. This immediately connects the features with the user. |
| Recognisable terms and familiar language/instructions, e.g., ‘Title’ or ‘Description’ in input fields | The use of recognisable and familiar language allows users to intuitively understand the instruction to enter a title of a post and the input functionality. |
| Navigation bar is in a place that is familiar to the user – navigation is at the top of the page. | The navigation bar is designed with the users’ habits in mind. |
| **#3: User control and freedom** | Users have the option to ‘dismiss’ reminders that stick to the bottom of the screen. | Users have the option to dismiss reminders at the bottom of the screen if it is blocking their workflow. |
| Users can choose the number of posts that are displayed in the list of viewing history | Users can control the amount of information that is displayed on the interface. |
| Users can exit the Help Menu using the ‘×’ button to return to the main page. | Users can escape the Help Menu and resume their work with a clear exit button. |
| Users can delete posts and comments | Users have the option to delete posts and comments |
| Cannot restore a deleted post or comment – a post’s data is permanently deleted and cannot be recovered. | Users cannot undo the deletion of a post because the action is permanent and unrecoverable. |
| Cannot delete user account or edit user information | Users cannot delete their accounts or edit and update their information. |
| Authors cannot change the access and sharing permission of the posts | Authors might want to keep some of their posts private. However, all posts published are public. |
| Users cannot opt-out of notifications | Notifications of other users commenting on their posts are notified via email but there is no option to unsubscribe notifications. |
| **#4: Consistency and standards** | Common website components e.g., notifications, modals, search bar lower the learning curves of users – user-familiar and consistent components increase user learnability and ease of interaction. | Beginner users can easily understand how to interact with common website components, and they do not need to learn new interactions. Users can therefore focus entirely on browsing post content. |
| User account login, home page and features are well-design and follow industry standards | Users are familiar with the layout and instructions for signing up or logging into an account. |
| Most icons, buttons and page layouts are visually consistent | Icons and buttons used on the website closely align with how other sites represent the same concept, e.g., submit and delete buttons. |
| Common utility tools, e.g., Save, Submit, Delete are externally consistent | Visual placement of utility tools is designed in compliance with ruling web conventions. |
| Home page and field titles remain a necessity and are externally consistent | There is an explicit link called ‘Home’ to minimise home page navigation confusion. Users can go back to the home page as a common task. It gives users a starting point if they are lost when exploring the website. |
| **#5: Error prevention** | User confirmation to permanently delete a post | Prevent users from accidentally deleting a post |
| Prevents the user from uploading unsupported file formats | There is guidance on the type of file formats that the code journal supports. |
| No auto-complete search suggestions for the search bar  (#6: Recognition rather than recall) | There is no auto-complete to prevent user slips from performing queries with typographic errors. |
| The text editor does not support undo and redo to prevent mistakes  (#3: User control and freedom) | Support undo and redo of edits can prevent user mistakes. Providing the ability to undo the most recent action helps users to feel more secure and more confident to experiment with unfamiliar features, since they are aware that a mistake is low cost and can be easily fixed. |
| **#6: Recognition rather than recall** | Viewing history on the user account page | The viewing history is a record of the addresses of the posts a user has recently visited, and data associated with those websites. The saved data helps make the website load faster if users revisit it. |
| Posts tags are categorically organised in the search results. Tags help users to browse and explore specific content. | Posts with similar tags are categorically organised in the search results. Users can easily identify similar and relevant posts by clicking on the tags. |
| Placeholders of input fields serve as prompts and instructions  (#10: Help and documentation) | Help users to recognise what information is required in the input fields. For example, ‘Title’ for the video title, ‘Press ENTER to add a new tag’ for users to add a new tag when uploading a new post. |
| Buttons on the navigation bar are recognisable and follow the user’s mental model (#2: Match between system and the real world) | For example, the recognisable ‘Home’ page button allows users to easily locate featured and posts that are available. |
| User account information provides a stronger cue for user control  (#3: User control and freedom) | User account information reminds the user is signed in and provides a stronger cue for user control and video management. |
| No auto-complete search suggestions for the search bar  (#5: Error prevention) | The search bar does not provide users with relevant search results while typing in keywords. Users might need to recall more relevant keywords for better search results. This increases the user’s memory burden. |
| **#7: Flexibility and efficiency of use** | Users can post multiple files for a post | The code journal supports multiple files for a post to the group and organises the files in categories. |
| Keyboard shortcuts serve as accelerators for expert users – speed up frequent switching of pages. | Keyboard shortcuts are available as an alternative way to switch pages for expert users. For example, ‘alt + b’ switches to the ‘Home’ page. Accelerators improve the efficiency of repetitious page transitions. |
| Users can upload and download files from posts | There is flexibility in accessing the files in posts. |
| Users can upload images and videos. The website shows a preview of images and video files. | Users can preview an image or a video before downloading a file. |
| Users who are authors can edit the source code of files. | The text editor allows authors to edit and change the content in source files. Users can then save new versions of such changes. |
| Hierarchical roles allow access to different user features – allow personalisation: users have different needs at different times. | Well-designed roles allow personalisation of user features – i.e., ‘Viewer’, “Author’, ‘Reviewer’ of posts. |
| ‘Upload’ supports a broad range of file formats – allows user flexibility. | Users have multiple methods to accomplish the same task. The file to be uploaded can be in different formats, so users can select a preferred format that works for them. |
| Users cannot customise the web interface to suit their needs | Expert users might want to customise the interface to suit their needs of peer-reviewing posts or organising posts in folders. The interface should adapt to the user’s dynamically changing needs. For example, a user can set up multiple window arrangements (workspaces). |
| **#8: Aesthetic and minimalist design** | Organised interface components: the Home, Upload and Account pages all adhere to the Gestalt proximity principle and have high usability. | The white space between fields makes the user’s uploading and logging in process intuitive and convenient. Input fields are legible and easy to read. |
| Syntax highlighting is a feature in the text editor of posts that displays text, especially source code, in different colours and fonts according to the category of terms. | Syntax highlighting focuses on language-specific keywords, operators, and elements that have the same meaning in any given piece of code. The code is more legible and easier to understand. |
| Contrasting colours of buttons, icons and page layouts are consistent   * Colour scheme enhances the contrast between the text and background | The journal adheres to a consistent colour scheme: orange, black, white, grey are prominent colours in the user interface. The website’s text contrast increases the legibility and usability. It also makes the design memorable. |
| Essential information for decision making is displayed on most pages  High signal-to-noise ratio – e.g. novice users can use the ‘Show Tour’ guide in the Help Menu. | There is no unnecessary noise or information on most pages and users have a clear mental model of the actions required to achieve the desired outcome. |
| Gestalt proximity principle is applied to user input design   * Field labels are close to their corresponding input fields. | Field labels are visually closer to the user input fields which increases usability and proximity. |
| Visual hierarchy of website components is proportional   * Design elements are organised according to the order of importance. | The sizes of the website buttons, icons, title, subtitle, field labels are proportional to their importance. The scale visually controls the delivery of the user experience. Small, medium, and large components in the design are easily differentiable. |
| Scalable and responsive web interface across mobile devices | Chrome’s inspection developer tool shows that the web application is responsive on mobile devices. |
| List view of post recommendations in ‘Home is clear and there is information of posts. | Clear layout of posts in ‘Home’. It is easy for the user to navigate from folder to folder. The meta information of posts helps the user to easily browse the posts that they are interested. |
| **#9: Help users recognise, diagnose, and recover from errors** | Error message of denying the user access to a deleted post – redirects users back to the home page. | The error message of denying the user access to a deleted post is complemented with a solution to explore other posts. |
|  | Error messages are written in a consistent style and tone of voice   * Readable plain language * Polite tonality * Precise descriptions | The code journal does not use technical jargons to overwhelm users and error messages are written in precise descriptions to assist the user in troubleshooting. |
|  | Tips and guidance messages stay away from user criticism   * Constructive advice * Explicit indication of an issue | Rather than criticising the user’s error, error messages explicitly indicate the issue and offer users constructive advice. |
|  | Error messages are displayed as pop-ups on the browser | The code journal clearly informs users when an error has occurred with an error message displayed as a pop-up. A combination of visual treatments and a message indicates that an error has occurred. |
|  | Search box does not preserve the user’s original query terms to facilitate revisions | If no search hits were found, the original query terms do not allow users to revise and search a wider scope without retyping queries. |
| **#10: Help and documentation** | Users can start of skip onboarding ‘Show Tour’ tutorials from the beginning (proactive help) | Familiarise new users with the interface at the first launch. |
|  | Confirmation before permanent deletion of data (proactive help) | Confirmation to prevent user mistake of permanently deletion. |
|  | Online documentation and help are readily available for users in the Help Menu | Users will be directed to the Help and Documentation page by clicking on the ‘Browse Support’ button. |
|  | Alternative text is not displayed on hover over buttons | Push revelations to assist users is a suggestion for improvement. |
|  | ‘Show Tour’ in the Help Menu provides a limited walkthrough for beginner users | Real-time user support is not available in case of an emergency. |
|  | No live support on the website | Real-time user support is not available in case of an emergency. |