

# Neilsen Usability Heuristics for User Interface Design

<https://www.nngroup.com/topic/web-usability/>

<https://blog.prototypr.io/10-usability-heuristics-with-examples-4a81ada920c>

## 1. Visibility of system status

- Keep users informed of what is going on/status of system (appropriate feedback, reasonable time)
- E.g. indicate that page is loading
- E.g. color change, loader, time-left graphics

## 2. Match between system and real world

- Follow real-world conventions, use phrases/concepts familiar to user
- Make information appear in a natural and logical order
- Cater to the language of the target user base

## 3. User control and freedom

- Support undo and redo for user mistakes
- Clearly indicated
- Support leaving an unwanted state (straightforward approach)

## 4. Consistency and standards

- Avoid using different words, actions for the same thing
- Formats of tables, buttons, display should be the same

## 5. Error prevention

- Prevent errors from occurring in the first place
- Check for errors, conditions leading to errors
- Present users with confirmation option
- E.g. correct spelling, set rules for user password (validate as user types)

## 6. Recognition rather than recall

- Instructions on how to use system should be visible/easily retrievable
- Make objects, actions, options visible
- E.g. suggest to user set of options (search bar), let them pick from list

## 7. Flexibility and efficiency of use

- Tailor system to support frequent actions for frequent users
- Keep system easy to use start for novice/beginner users
- E.g. default and custom installation of software ("Advanced Settings")

## 8. Aesthetic and minimalist design

- Avoid irrelevant, rarely needed information
- Clear elements and content that do not support page goals and tasks
- Hide additional information ("Learn More")

## 9. Help users recognise, diagnose and recover from errors

- Express errors in plain language (no codes)

- Precisely indicate problem and solution
- Exception handling (so relevant error messages to user)
- E.g. Empty state messaging (when item's content can't be shown/"No items to show", 404, 500 (pages that suggest actions))

<https://material.io/design/communication/empty-states.html#content>

## 10. Help and documentation

- Not be too large
- Easy to search
- Focus on user's task, concrete steps to be carried out

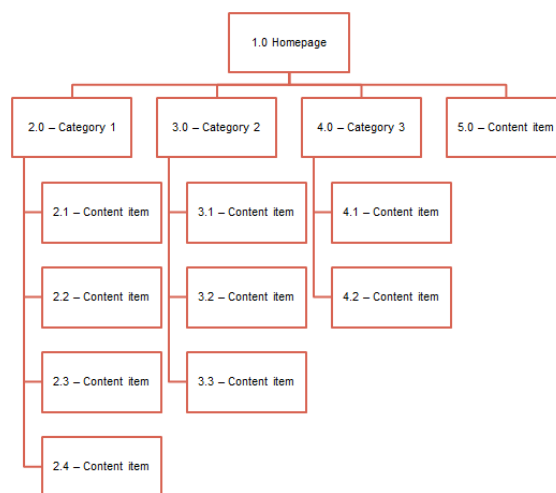
## Terminology

### - Sitemap

<https://theuxreview.co.uk/sitemaps-the-beginners-guide/>

<https://www.uxbooth.com/articles/the-rules-for-modern-navigation/>

- Hierarchical diagram showing structure of application
- Show how navigation is structured (which page leads to what page(s))
- Show relationship between different pages
- Key attributes: homepage/reference number and label for each item



- Strive for “flat navigation”, where user can access deepest nested page in site within one to two clicks

### - Card sort

<https://dynamapper.com/blog/19-ux/425-card-sorting-guide>

- Give users opportunity to categorise content
- Understand how users will navigate through pages while searching for specific content
- List of topics, list of categories (users can change, can add to these)
- Sort into piles, group contents in such a way that is intuitive and convenient to user

### - Wireframes

<https://theuxreview.co.uk/wireframes-beginners-guide/>

- Visual representation of user interface

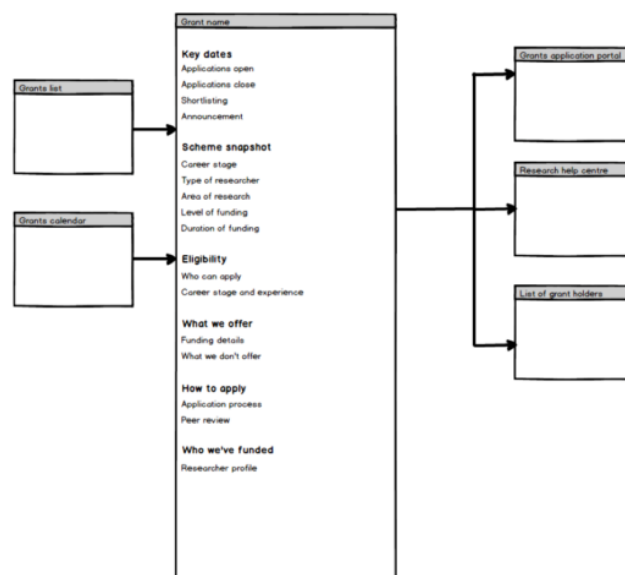
- Define hierarchy of items on screen
- Ranges from low-fidelity sketch to high-fidelity and fully annotated diagram
- Start with user's goals, motivations and tasks in mind
- Understand structure of website (through sitemap)/purpose of each page
- Avoid putting images

## - Content models

<https://uxmastery.com/content-modeling/>

<https://alistapart.com/article/content-modelling-a-master-skill>

- Documents all the different types (elements/classes) of content in a given project
  - Contains detailed definitions of each content type's elements and their relationships to each other
1. Understand types of content you have
  2. Analyse parts each type is made of
  3. Define how those parts relate to each other (e.g. ingoing and outgoing paths)
- Consistency in page structure (e.g. same highlights, same grouping of information)
  - Give components names, describe attributes (i.e. shape, colour, essential or optional) and combine in different ways to create new formats



- Questions to ask:
1. How structured does content have to be? (Need to have specific, uniquely identifiable data for sorting/filtering content?)
  2. How flexible does content have to be? (Need to be specific in elements of a page, number and order of pages? Or need to be able to add varying content elements?)
  3. How reusable does content have to be? Can make reusable parts separate and shared across different pages?
  4. Is it all right to have users/admin fill in many distinct data fields in order to enter unstructured content?

## - Scenario maps

<http://www.uxforthemasses.com/scenario-mapping/>

<https://www.nngroup.com/articles/ux-mapping-cheat-sheet/>

- Story outlining how a user will go about completing a tasks
- Can be really detailed or very high level
- Minimally: who, what, when, where, why, how
- Maps out all the steps that a user will take to complete a tasks
- Focuses on what user will do, not how he will do it

1. What the user does (what happens)

E.g. User brings up a larger image of the item which he thinks...

2. Comments or information relevant and important at that particular step

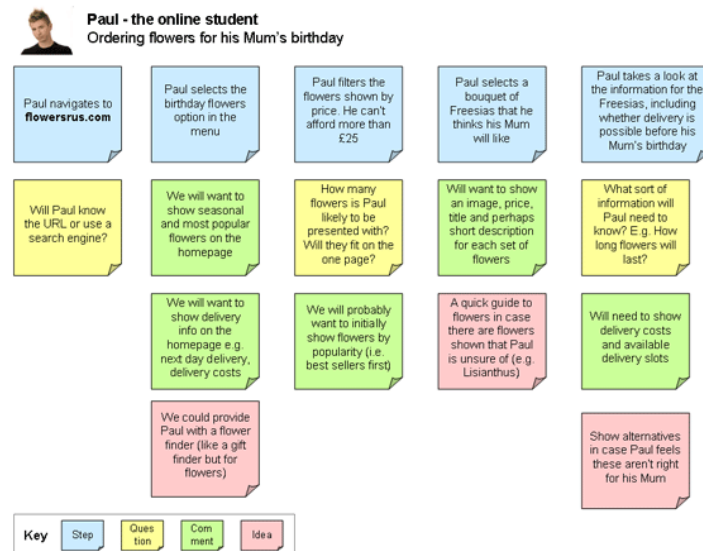
E.g. Alternative images available for item, front and side shot

3. Questions and assumptions

E.g. Will images be of the same size and aspect ratio?

4. Ideas or good suggestions

E.g. Good to be able to zoom in to see item in more detail



- Keep steps relatively high-level, focus on key tasks (scenarios where everything goes right)

- Style guides

<https://www.toptal.com/designers/ui/ui-styleguide-better-ux>

<https://www.toptal.com/designers/ux/why-startups-need-a-styleguide>

<http://www.bbc.co.uk/gel/>

<https://speckyboy.com/inspirational-examples-ui-style-guides/>

<https://dribbble.com/shots/1817828-Style-Guide/attachments/301226>

- Record all design elements and interactions in product
- List crucial IU components (e.g. buttons, typography, colour, navigational menus)
- Document important UX components (e.g hover states, dropdown fills, animations)

- Documentation

<https://www.uxpin.com/studio/blog/stop-busywork-the-guide-to-strategic-ux-documentation/>

<https://www.uxpin.com/studio/blog/3-step-guide-lightweight-ux-documentation/>

<https://medium.com/the-versatile-designer/10-principles-for-good-ux-design-documentation-dd5b812bebc1>

[https://docs.google.com/document/d/1yrU5F6Gxhkfma91wf\\_lbZfexw8\\_fahbGQLW3EvwdfQI/edit](https://docs.google.com/document/d/1yrU5F6Gxhkfma91wf_lbZfexw8_fahbGQLW3EvwdfQI/edit)

<https://react-styleguidist.js.org/docs/documenting.html>

- For UX design, research documentation
- User research, customer profile kits, experience maps, user journeys, wire-flows, wireframes, sitemaps, style guides
- Understand target user-base, how document will be used, language readable to user
- Version control, can be easily edited by anyone, always up-to-date
- Replace initial assumptions, speculative data, outdated research and biased opinions with real data and insights as project progresses
- Good to have illustrations (not decorative but informative) like animated gifs, screen record best practices

### - Usability testing

<https://www.experienceux.co.uk/faqs/what-is-usability-testing/>

- Ask actual users to complete tasks, see where they encounter problems and confusion
- Evaluate how easy a website is to use, for user to accomplish goals
- Directly collecting feedback from end user
- Different options for different users, observe differences in reaction

### - Personas

<https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>

<https://www.usability.gov/how-to-and-tools/methods/personas.html>

- Fictional characters created to represent different user types who may use the product
- They do not describe real people but composed from real data collected from multiple individuals
- Design according to how personas will experience, react and behave in relation to features or changes
- Understand and meet the underlying needs of personas

## Details

### Empty states

<https://material.io/design/communication/empty-states.html#content>

- Basic: non-interactive image (drawing, humorous, cute, consistent with other pages) and text tagline (helpful, consistent, conveys purpose, non-actionable/no call-to-action)
- Alt: starter content/recommendations

### Search bar

<https://material.io/design/navigation/search.html#persistent-search>

- Show best match if no matches yet
- Extensions: historical suggestions, auto-completion of queries, voice input

### Navigational bar

<https://material.io/design/components/app-bars-top.html#theming>

## Dividers

<https://material.io/design/components/dividers.html#types>

- Use inset dividers to group items (not separate individual items)
- Use inset dividers when there are anchoring elements (e.g. icon/avatar)

## Dashboard

<https://uxplanet.org/10-rules-for-better-dashboard-design-ef68189d734c>

- Simple view, present all information
- Show trends, updates
- Preview of crucial information important to user
- Easy to navigate to various areas of application that need attention
- Some suggestions:

### 1. Purpose of dashboard

- **Operational** (tasks are time-sensitive, present data deviations and current resources/status, support actions for speed and efficiency)
- **Analytical** (provide at-a-glance information, not focused on immediate action, help users analyse trends and decision-making)

### 2. Representation of data

- Static, dynamic changes over time
- Relationships (scatterplots, network diagrams, bubble charts)
- Compare (barcharts, line plot)
- See parts of the whole (piecharts)
- Distribution (histogram, outliers, range, median)

### 3. Clear and consistent naming convention and formatting

- Data consistency

### 4. Layout

- Decide flow of layout and prioritise
- Grids
- Position key information from left to right (top to bottom)
- Create layout such that users don't need to go back and forth (continuous flow for easy scanning)

### 5. Building blocks with consistent structure

- Cards that hold information, charts and control
- Infinitely manipulatable, good for responsive design/user customisation
- Increases overall scalability of designs in future

### 6. Double your margins

- 24px for padding

### 7. Don't hide information, or rely on interactions too much

- Avoid too much scrolling or interactions
- Avoid positioning information one under another to the point that only information visible above screen fold visible to user at the start
- Prioritise and identify core information
- Summarise key information
- Use additional interactions to fit more content (but don't fully rely on them)
- Don't keep information in different tabs, but don't overload dashboard with too many things either (max 5-7 widgets)

## 8. Personalisation rather than customisation

- Content relevant to individual needs of users
- Personalisation (by system itself, identify users by id to match content/functionality to their role)
- Customisation (by user, in order to meet specific needs – configure layout/content/system functionality)

## 9. Make sure data tables/lists are interactive and correctly aligned

### Data tables

<https://medium.com/@taras.bakusevych/data-tables-design-3c705b106a64>

- Grouping and sorting (Consolidating items in the same group under a section that is expanded when needed)
- Bulk actions (buttons for actions at the top, check list at the side to apply features to relevant rows)
- Inline edit, editing data straight in the table itself

### Maintenance messages

<https://www.atlassian.com/blog/statuspage/scheduled-maintenance-message-examples-inspiration>

- Think about interruption from users' perspective (understanding scope and impact of maintenance period)
- Big impact: more updates, more channels, longer lead time
- Lower impact: less detailed, less frequent updates
- Add additional reminder update right before maintenance period begins
- Updates so far in advance (longer than a week) will require more reminder updates to be scheduled
- Include the right details, but not too much
  1. What team is doing the work
  2. Exactly what work is being done
  3. What services will and will not be impacted
  4. How to follow up with questions or concerns
  5. Specifically when the maintenance window is scheduled for (broken out from other txt so it's easy to find and read)
  6. Name of sender
- Examples: *ToutApp Scheduled Maintenance, Friday Aug 29, 6pm PST / 9pm EST*
- Examples: *Maintenance: Heroku Postgres Backups Service Update on Wednesday April 18 at 20:00 UTC (16:00 EDT)*
- Format: *<Application Name> Scheduled Maintenance, <Day Month Date>, <Time>*

### Recommended content

<https://www.nngroup.com/articles/recommendation-guidelines/>

- Present recommendations prominently
- Segment suggestions into clear and specific categories
- If personalised suggestions are too diverse (due to user's varied interests), try separating recommended items according to genre
- Provide methods for users to give feedback or edit the data used to create the recommendation (e.g. Giving a thumbs-down to a recommended song on Spotify to feedback not to play similar songs in the future)
- Allows discounting of one-time behaviour (e.g. purchasing a gift for a friend with very different interests)
- Personalised suggestions that avoids information overload
- Understand source of recommendations, adds credibility to recommendation (e.g. Netflix: "Because you watch so-and-so show in the past, we thought you'll like...")
- Explanation of recommendation helps users gauge type of content included (whether they are interested in suggestions or should continue looking)
- Examples: "Based on your past history", "Related to Items You've Purchased"
- Don't neglect the ability to interact with suggestions to provide good/bad feedback

- Remember to have straightforward and descriptive messaging (clear state what each action entails and does), users appreciated it
- Update recommendations quickly and often (changes made to preferences should be reflected immediately)