UX Rules

Vocabulary

Affordance

The ability to access an object based on it's preceptive properties, i.e. a button gets clicked

Cognitive Load

The amount of mental processing power required to use something.

Information Scent

Clues that tell us what a webpage is about, or where a link will lead

Mental Model

Representation of facts (imaginary or real) associated with a certain concept or process

Effort

Is the sum of the required: Attention, Interaction, Reading and thinking

General

- attractive things work better
- minimize number of page loads
- Show all steps in a process, create a mental model
- Display relationships in data to help users build a mental model
- Buttons are for actions, links are for navigation
- Use clear specific verbs that describe actions on buttons
- Don't place buttons far from identifying information
- Don't style buttons as links
- Designs should focus on:
  - decreasing cognitive load
  - decreasing interaction cost (number of clicks, harder to read)
  - anticipating user needs
  - persuade (not deceive)
- Design task focused, not device or interface focused
  - Handoff to other devices when possible
- Users should have a clear line to escalate for help (clean error messages and service desk contact)
- Data should be presented in clean unbiased ways. Avoid decorating
- Data should be exportable
- Users should always know the current application mode, including quasimodes (like pressing shift key or caps lock)
- Use modes sparingly and make them very easy and consistent to get into and out of
- Modes need a high level of discoverability
- Modes need enough features to be worth it

- Limit the amount of information with progressive disclosure
    - □ Grey out options until behaviour is ready
    - □ don't show options until available or sought after (nested menue)
- Respect spatial memory, keep things in place and places organized (don't move controls)
- Icons should be labeled unless they are being used in known ways, e.g. wifi icon
- Don't use icons for more than one purpose
- Don't reuse icons
- Do not override default browser behavior like right click. Augment it instead if possible
- Text should try to be between 48-72 characters in length

Navigation

- Allow for multiple navigation types
    - Content links
    - Search
    - Site Nav
- Menus should support keyboard navigation
- In menus or lists, expose common or likely options. The items should appear in the list or menu as well.
- Avoid labels for overflow menus like 'more', or 'actions' be more descriptive
- Avoid button or navigation duplication. Users will try to learn the difference
- Use accordions on mobile to provide a table of contents for the user to the article

Interaction

- Larger targets are easier to hit, width is more important than width for click-ability
- horizontal content is cheaper for user attention
- Touch areas should be min 1cm x 1cm
- group related actions
- Data inputs should autosave
- Modals should stop content from being interactable
- Clicking outside of a modal should close it
- Clickability is primarily queued to the user through contrast, avoid ghost buttons.
- Tooltip timing:

  wait .1s, show feedback .3-.5s, show content after .5s.
- Use chevrons to indicate accordions
- Toolbars should be grouped by purpose, eg. edit, export
- If using a contextual menu, the user should always be aware of what is in focus
- Do not use sliders if precision is needed
- Do not use numbers for inputs where specificity is not required
- Colour can never be relied upon for meaning, interaction or navigation
- Buttons should avoid generic terms, yes, no, ok, confirm, cancel. They don't really say anything
- If it takes longer than 5s to come up with an idea for an icon, don't use an icon
- Test icons for information scent
- Split menu buttons should be very obvious or they will confuse the user, show where the click areas are
- Repetitive work should support keyboard workflows

- Do not use knobs as input controls
- Video should never autoplay
- Dropdowns should be limited to 7 items
- Do not use images on mobile unless they are relevant
- Use spinners for short load times and progress bars for long load times
- Don't ever reset a progress bar; use one bar for all tasks
- Dropdowns are better used for progressive disclosure than as an input

Forms

- Use as few fields as possible
- Provide good defaults
- Auto-populate what you can
- Auto format with input masks
- show required fields
- don't use placeholders
- error messages close to fields
- descriptive labels
- label above field
- Align labels left
- Limit animation
- Use good text contrast
- Don't use jargon
- Group items meaningfully
- Submit button below last input
- Autosave input

- Only use multi column layouts when the content calls for it (address) not for aesthetics
- Forms all need labels and descriptions above them
- do not use placeholder text
- mark required fields
- autocomplete when possible
- show error messages by the label
- Do not have errors related to time, no fade out
- Don't use multi select, use checkboxes
- Pickers should support most common entries (rooms, short list at the top)
- Selects should use the most popular option as default

Wizards

Use a wizard for a rarely done task where the user needs guidance

- Give the user a persistent diagram of the steps
- Give the user a list of required information at the start
- Pre fill as much as possible
- Use smart defaults
- If a field is conditional make it very obvious that it is now available, animate it is a good solution
- Allow users to review steps non-sequentially. Support jumping to a specific step in the persistent diagram of steps.
- Avoid forcing users to redo steps

Tables

Use for when you need to:

- Find a specific record with a large set of records
- Compare multiple items
- see status of multiple items
- act on multiple items at once
- add new information while seeing the context of other examples

table designs

- Group items if possible
- Default table layouts are high priority things on the left
- Allow users to easily hide/show columns
- Show the number of hidden columns
- Show a full list of available columns
- Allow for column reorder
- Preserve reorder and add a reset button for default
- Allow for user based order reset on complex tables
- Show clearly what the table is sorted by
- Don't use table modals
- Indicate clearly cell truncation
- Highlight row on hover or selection
- Align cells left
- Right align numeric cells like price that have the same number of decimals
- Do not center align
- Table actions should be buttons (sort, filter)
- Links within tables should have a strong information scent
- Have a clear all filter

- Allow for filters to be additive
- Allow for filters to be removed individually
- Separate sort from filters
- Show all filters applied individually
- table actions should be placed at top of table

Beneficial patterns for tables are

- table drawers,

  table rows can be expanded with context below them

- Drag and drop grouping

  Support grouping items by dragging a cell header into place

  Allow users to provide a temporary grouping of items

  Allow for grouping to be collapsible

  one click de-grouping

  Don't make drag and drop the only way to group

- Alternatives to modals

  Table Drawers

  Edit in place

  side panel edit

  Separate pages or windows (best for editing multiple items and providing highest level of focus)

- Table Pagination

  Show first page

  Show last page

  Indicate current page

  Show +- 1 from current

  Include a 'Jump to' input

Show total number of rows

Dashboards

Dashboards a interfaces that are designed for displaying values

- Dashboards are not for navigation
- Dashboards are not portals
- Dashboards should be focusing on minimizing cognitive load
- Show all information on one screen
- No scrolling
- Use length for comparison, not area
- Colour is for categorizing, not quantifying
- Label axis clearly
- Use bullet charts over gauges

Tokens

Also called chips

Used to signify status of things

Solid background with high level of contrast

Can be individually manipulated

Change Management Principles

1. Iterative change is easier to manage
2. warn users in advance about upcoming changes, communicate those changes
3. Let users toggle between old and new versions
4. Provide transition support and instruction

5. Offer a dedicated feedback channel
6. Tell user how you are addressing their key issues