# 3.5 INTERACTION BEHAVIOURS

**Interaction Behaviours:**

- **Interaction behaviours** describe how users act, react, and respond while interacting with a system, interface, or product.
- Understanding these helps designers create intuitive and user-friendly experiences.

**Key Interaction Behaviours:**

- **Goal-oriented behaviour:** Users interact with systems to achieve specific goals quickly and efficiently (e.g., booking a ticket, submitting a form).
- **Exploratory behaviour:** Users browse or explore features without a fixed goal, common in learning or discovery-based apps.
- **Habitual behaviour:** Actions performed repeatedly and automatically, such as checking notifications or refreshing feeds.
- **Error-driven behaviour:** Users react to errors by retrying, undoing, or seeking help, highlighting the need for good error handling.
- **Contextual behaviour:** User actions vary based on context like device, environment, time, or emotional state.

**Importance of Studying Interaction Behaviours:**

- Helps predict user actions and expectations.
- Reduces cognitive load and user frustration.
- Improves task completion speed and satisfaction.
- Supports inclusive and accessible design.

**Interaction Design Principles:**

**Interaction design principles** are guidelines that ensure usability, consistency, and positive user experience.

**a) Visibility:**

- Visibility means that important elements and actions should be clearly noticeable and easy to find.
- If users cannot see a function, they cannot use it.
- Users instantly understand what actions are possible without searching.

**Application:**

- Organize elements using visual hierarchy

- Keep navigation visible and consistent
- Highlight key actions (e.g., primary buttons)
- Use progressive disclosure (show information only when needed)

**Example:**

- ✓ A visible "Submit" button placed at the bottom of a form.
- ✓ "Add to Cart" and "Buy Now" buttons are **highly visible** and highlighted in Amazon App.

**b) Feedback:**

- ➢ Feedback informs users about the result of their actions.
- ➢ It eliminates confusion and builds confidence.
- ➢ Users receive immediate confirmation that their action was successful.

**Types of Feedback:**

- Visual (color change, animations)
- Auditory (notification sounds)
- System messages (success/error alerts)

**Application:**

- Provide immediate response to user actions
- Show progress indicators for loading actions
- Clearly highlight errors and how to fix them

**Example:**

- ✓ A button changing color when clicked.
- ✓ Double ticks indicate message delivery and reading in WhatsApp.
- ✓ Typing indicator shows when the other user is typing.

**c) Consistency:**

- ➢ Consistency ensures that similar actions and elements behave the same way throughout the interface.
- ➢ Reduces learning effort and cognitive load.

**Types of Consistency:**

- Visual (colors, fonts, icons)
- Functional (same action → same result)
- Content (terminology, tone)
- Platform (same experience across devices)

**Application:**

- Use standard UI patterns
- Maintain uniform layouts and controls
- Keep terminology consistent

**Example:**

- ✓ Same icon used for "Settings" across all screens.
- ✓ Like, Comment, and Share buttons appear in the same order everywhere in Facebook.
- ✓ Play, Pause, Volume icons remain consistent across devices in Youtube.

**d) Affordance:**

- ➢ Affordance indicates how an element should be used without instructions.
- ➢ Makes interfaces intuitive and self-explanatory.
- ➢ Users instinctively know how to interact with elements.

**Application:**

- Design clickable elements to look clickable
- Use real-world metaphors
- Provide visual cues and hover effects

**Example:**

- ✓ A raised button suggests it can be pressed.
- ✓ Plus (+) button suggests adding new files in Google Drive.

**e) Learnability:**

- ➢ Learnability refers to how easily new users can understand and use an interface.
- ➢ First impressions strongly influence user retention.
- ➢ New users quickly learn how to use the app with minimal effort.

**Key Aspects:**

- Intuitive navigation
- Familiar patterns
- Clear feedback

**Application:**

- Simplify initial interface
- Hide advanced features

- Provide hints and visual guidance

**Example:**

- ✓ Tooltips explaining icons for first-time users.
- ✓ Simple layout with clear labels like "Mute" and "Start Video" in Zoom.

**f) Flexibility:**

- ➢ Flexibility allows users to customize and adapt the app to their personal preferences.
- ➢ Supports diverse users, devices, and abilities.

**Application:**

- Allow theme and font size changes
- Design responsive layouts
- Support multiple input methods

**Example:**

- ✓ Dark mode and adjustable text size options.
- ✓ Users can change themes, font sizes, and layouts.

**g) User Control:**

- ➢ User control ensures users feel empowered, not trapped.
- ➢ Builds trust and reduces frustration.
- ➢ Users feel safe experimenting without fear of irreversible mistakes.

**Key Controls:**

- Undo / Redo
- Cancel actions
- Back navigation

**Application:**

- Confirm destructive actions
- Allow reversal of mistakes
- Provide clear exit paths

**Example:**

- ✓ "Undo delete" option after removing a file.
- ✓ Undo and Redo options for every action.

**h) Simplicity:**

- ➢ Simplicity focuses on minimizing complexity and cognitive load.

- Improves efficiency and usability.
- Users complete tasks quickly and efficiently.

**Application:**

- Remove unnecessary elements
- Use clear language
- Maintain white space
- Prioritize essential information

**Example:**

- ✓ Minimal login page with only essential fields.
- ✓ Simple booking process: location → destination → confirm ride.

**i) Accessibility:**

- Accessibility ensures interfaces are usable by people of all abilities.
- Promotes inclusivity and legal compliance.
- People of all abilities can interact comfortably with the system.

**Application:**

- Follow WCAG guidelines
- Provide alt text for images
- Ensure keyboard navigation
- Maintain sufficient color contrast

**Example:**

- ✓ Screen-reader friendly navigation.
- ✓ High-contrast mode and text resizing.

**j) Error Prevention and Recovery:**

- This principle focuses on preventing errors and helping users recover easily.
- Reduces frustration and task failure.
- Users are guided to avoid mistakes and recover easily.

**Application:**

- Anticipate possible user mistakes
- Display clear, friendly error messages
- Provide undo options
- Offer help resources

**Example:**

- ✓ Warning message before deleting important data.
- ✓ Confirmation prompts before money transfers.

**Relationship Between Behaviours and Principles**

- Interaction behaviours explain **how users act**.
- Interaction principles guide **how systems should respond**.
- Together, they ensure user-centered, intuitive, and effective interface design.

## 3.6 SKETCHING PRINCIPLES

**Introduction to Sketching in UI/UX:**

- Sketching is a foundational activity in UI/UX design.
- It allows designers to quickly explore ideas, experiment with layouts, and visualize solutions before moving into detailed wireframes or prototypes.
- These sketches are usually rough and hand-drawn, focusing on structure and functionality rather than visual perfection.

**Sketching helps designers:**

- Think creatively
- Identify potential usability problems early
- Communicate ideas clearly to stakeholders
- Refine concepts before investing time in high-fidelity designs

It is a low-cost, high-impact method for early-stage design exploration.

**Benefits of Sketching:**

**1. Idea Generation:**

- Sketching enables rapid generation of multiple design concepts.
- Designers can quickly draft various layouts and compare them to determine the most promising solution.

**2. Visualizing Design Concepts**

- It helps designers visually represent abstract ideas.
- By sketching, they can test variations in layout, navigation, and user flow at an early stage.

**3. Speed and Flexibility:**

- Sketching is fast and adaptable.
- Designers can easily modify or discard ideas without wasting time on detailed development.

**4. Collaboration:**

- Sketches are simple and easy to understand, making them ideal for discussions with stakeholders, developers, and clients.
- They encourage constructive feedback and shared understanding.

**5. Cost-Effective:**

- Compared to high-fidelity prototypes, sketching requires minimal resources.
- It reduces development costs by eliminating weak concepts early.

**Tips for Sketching:**

**1. Don't Worry About Being Messy:**

Sketches are not final products. Rough, imperfect drawings encourage creativity and faster idea generation.

**2. Start with Pencil and Paper:**

Using pencil and paper avoids distractions from digital tools and promotes free thinking.

**3. Explore Multiple Concepts:**

Designers should create multiple variations instead of focusing on a single idea. This helps in comparing and selecting better solutions.

**4. Keep It Simple:**

Use basic shapes, lines, and labels. Avoid unnecessary design details like color and typography at this stage.

**5. Sketch Quickly:**

Speed keeps creativity flowing and prevents emotional attachment to one concept.

**6. Get Feedback:**

Share sketches with stakeholders and teammates to gather insights and refine ideas.

**7. Iterate:**

Refine sketches continuously based on feedback and evaluation.

**8. Use Templates:**

Templates improve consistency and speed in creating structured sketches.

**Different Sketching Techniques:**

**1. Sketch Noting:**

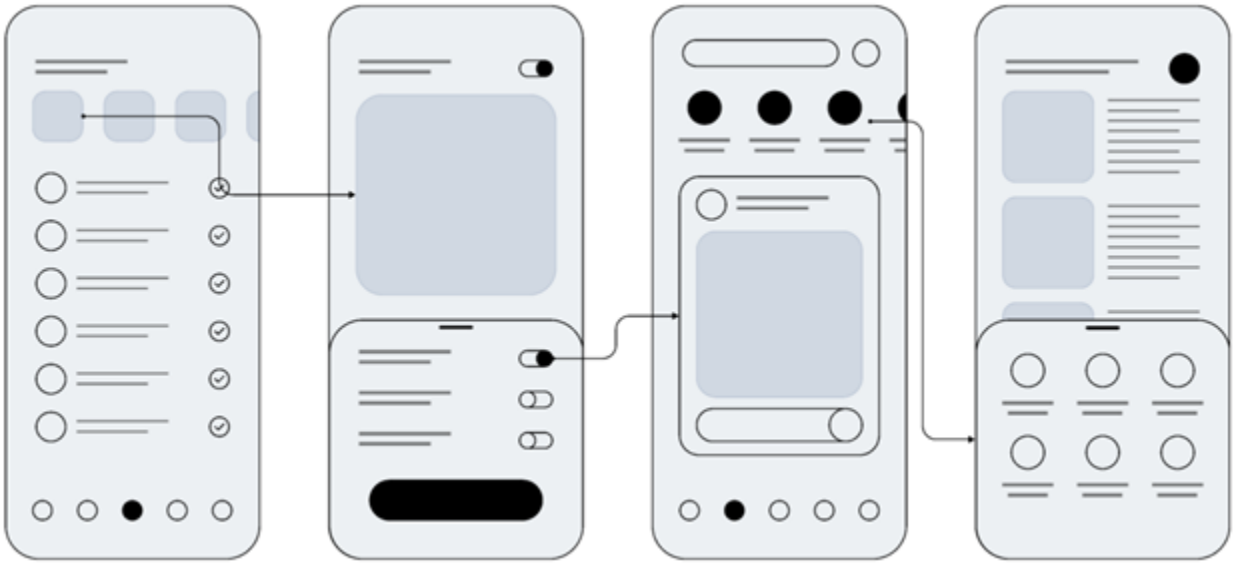
- This involves creating visual notes during meetings or interviews.
- It helps capture important ideas and identify patterns.

**2. Storyboarding:**

- Storyboards visually represent user journeys and interactions.
- They help designers understand how users interact with a product over time.

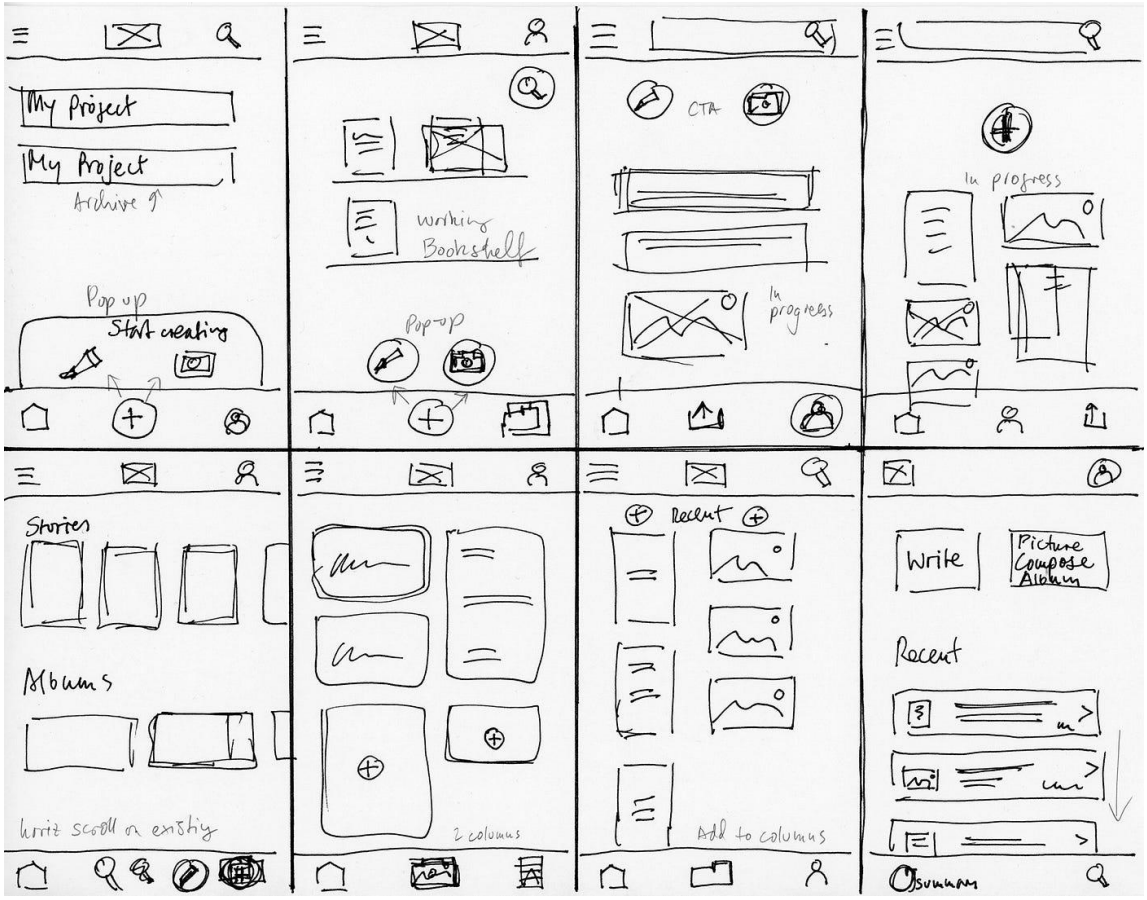
**3. Wireframing:**

Wireframes are low-fidelity layouts that focus on structure and placement of UI elements without detailed design.



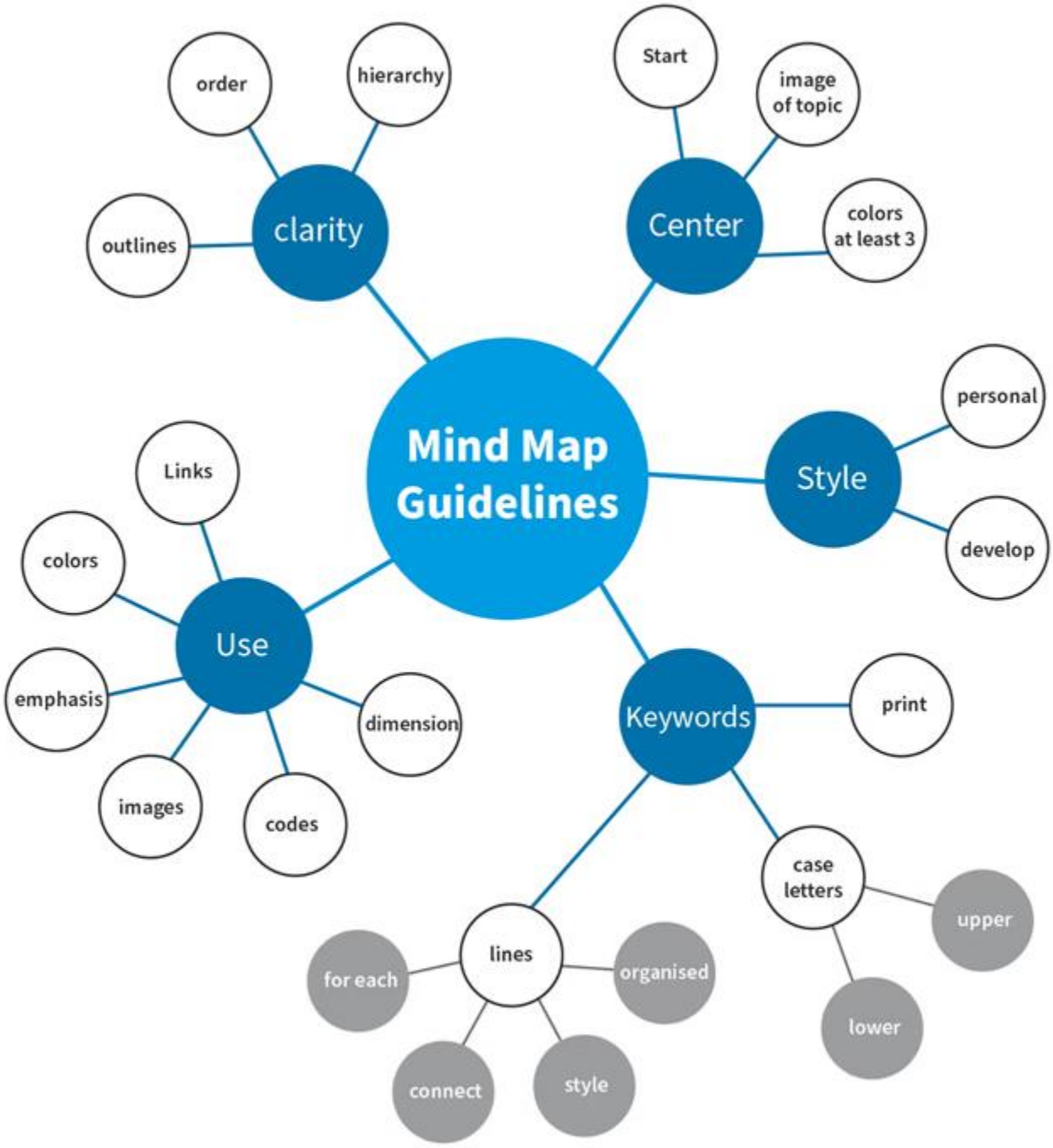
**4. Thumbnail Sketching:**

Small, quick sketches used to explore multiple layout ideas rapidly.



**5. Mind Mapping:**

A visual diagram that connects ideas and concepts, useful for brainstorming and organizing thoughts.



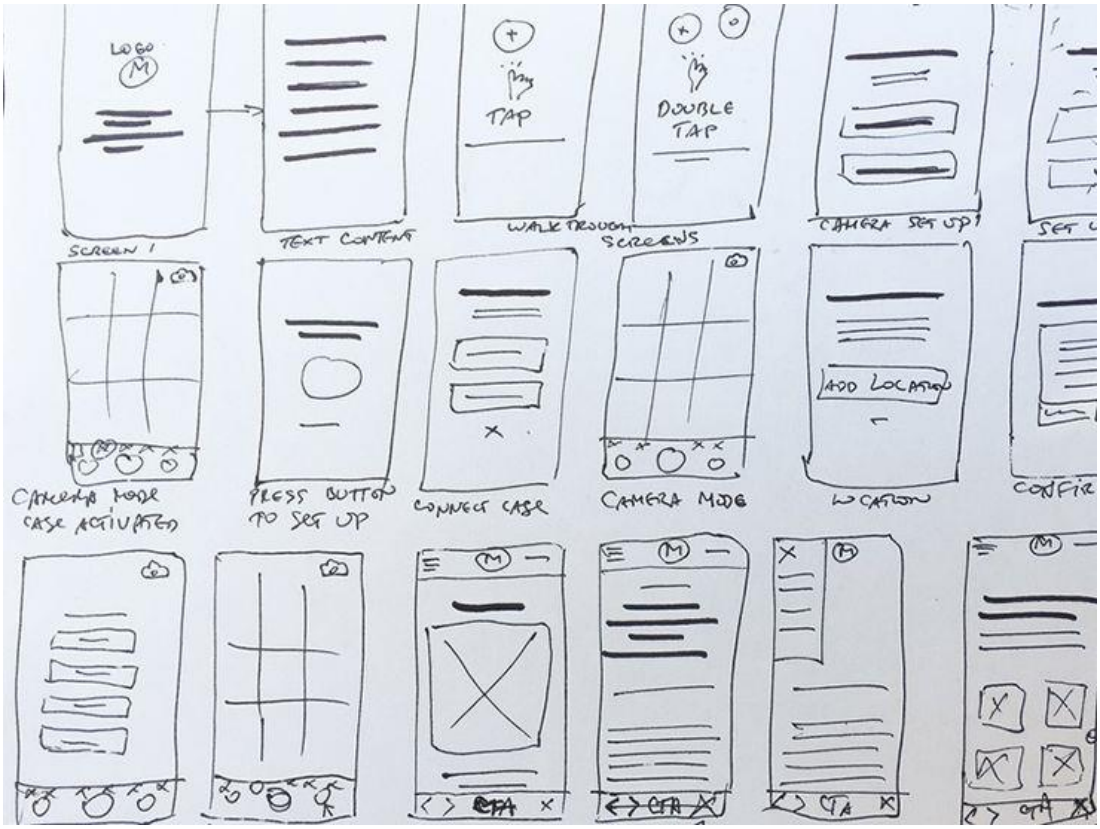

**6. Rapid Prototyping:**

Creating simple interactive models to test functionality and gather user feedback early.

**Tools for Sketching:**

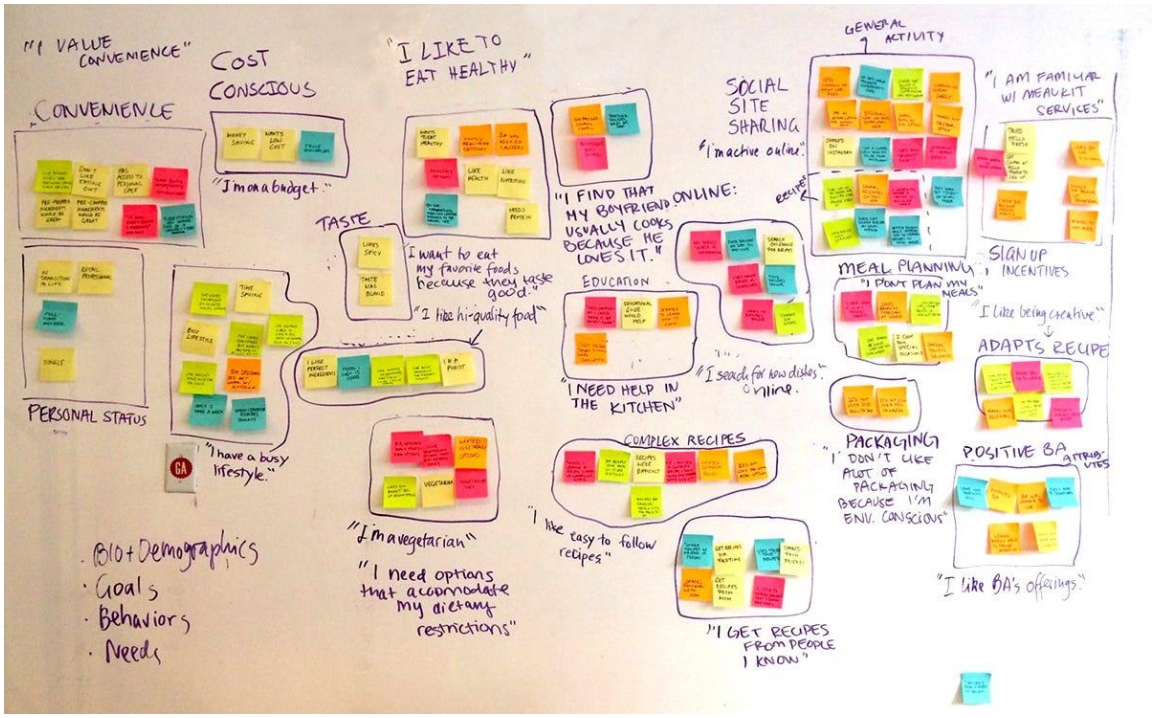
**1. Pencil and Paper:**

Traditional and cost-effective. Encourages quick and spontaneous ideation.




**2. Sticky Notes:**

Flexible and useful for collaborative brainstorming sessions.



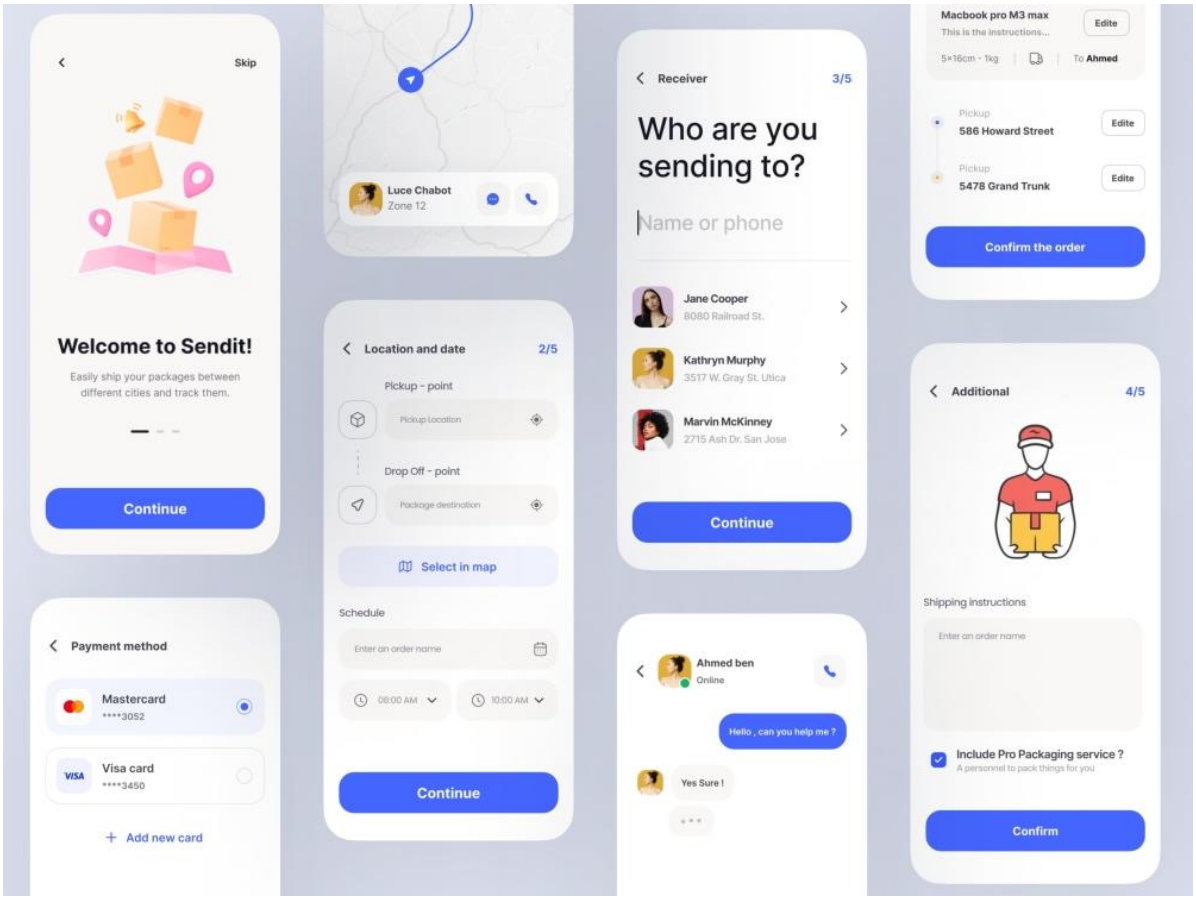
### 3. Digital Sketching Tablets:

Devices like the Wacom Intuos and iPad Pro allow precise digital sketching.



### 4. Sketching Software:

Tools like Sketch and Figma offer grids, templates, and UI components for digital design.



**5. Whiteboards:**

Ideal for team collaboration and group ideation.

