

Handz V1 PRD Bundle

CH06 — Design System (R1)

Doc ID: HZ-V1-CH06_Design_System_R1

Revision: R1 (2026-01-02)

Status: Draft

Depends on: HZ-V1-CH00 (Manifest), HZ-V1-CH01 (Product Definition), HZ-V1-CH04 (IA & Nav), HZ-V1-CH05 (Screen Inventory)

Related: HZ-V1-CH12 (Flow Builder Interaction Model), HZ-V1-CH20–CH22 (Practice Mode), HZ-V1-CH25 (Monetization), HZ-V1-CH31 (Error States), HZ-V1-CH32 (Accessibility)

Supersedes: —

Owned Decisions: UI look/feel, token names, component specs, motion rules, typography rules

Open Questions / Placeholders: See §12

Purpose. This chapter defines the visual language and interaction feel for Handz V1: colors, typography, spacing, components, motion, and UI states. It is written so implementation teams (including vibe-coding agents) can build consistent UI without guessing or inventing styles screen-by-screen.

1. Design North Star

Handz should feel **minimal + modern** (Notion-like clarity), **slightly playful** (Duolingo-like friendliness), and **progress / game oriented** (SupaBetter-like momentum) without looking generic. The UI should feel “addictive” in the sense that it rewards small actions, keeps screens clean, and makes the next step obvious.

Core vibe keywords: clean, punchy, confident, fast, friendly, focused.

Striking context: use striking metaphors subtly (chips, tags, streaks, “round” language), but avoid gritty / aggressive visual tropes.

1.1 Principles

- **Progressive disclosure:** show only what’s needed now; advanced detail is always available one tap away.
- **Fast UI:** most interactions should feel instantaneous; avoid heavy transitions and long loaders.
- **One primary action per screen:** the main CTA is visually dominant; secondary actions are quiet.
- **Readable at a glance:** large headings, spacious line-height, strong contrast; no dense walls of text.
- **System consistency:** reuse the same components and states everywhere; do not hand-style one-off UIs.

2. Color System

Handz uses a light, clean base (white / off-white) with a bold, high-energy orange as the brand anchor. All colors are defined as tokens; screens and components must reference tokens only (no hard-coded hex

values in UI).

2.1 Brand anchor (locked)

Primary Orange: #FF7F11 (used in brand accents and the app icon background).

If the palette is revised later, Primary Orange remains the anchor unless CH06 is revised (R2+).

2.2 Light theme tokens (default)

Token	Meaning / Usage
color.brand.primary	#FF7F11 — Primary CTA, active states, key highlights.
color.brand.primaryHover	#E86F05 — Pressed state for primary CTA (iOS: pressed/active).
color.brand.primarySoft	#FFE6D3 — Subtle highlight backgrounds (badges, callouts).
color.text.primary	#0F172A — Main text.
color.text.secondary	#475569 — Secondary / helper text.
color.text.tertiary	#64748B — Placeholder / meta text.
color.surface.base	#FFFFFF — Main surface.
color.surface.alt	#F8FAFC — Alt surface (lists, grouped backgrounds).
color.surface.elevated	#FFFFFF — Cards / sheets; use elevation for separation.
color.border.subtle	#E2E8F0 — Hairline borders.
color.border.strong	#CBD5E1 — Stronger dividers / emphasized outlines.
color.action.disabledBg	#E5E7EB — Disabled control background.
color.action.disabledText	#9CA3AF — Disabled label.
color.semantic.success	#16A34A — Success states.
color.semantic.warning	#F59E0B — Warning states.
color.semantic.danger	#EF4444 — Destructive / error states.
color.semantic.info	#2563EB — Informational states / links.

2.3 Usage rules

- Primary Orange is used for **one thing at a time**: the primary CTA or the primary focus element on the screen.
- Never use red/orange together as two competing accents on the same screen; danger red is reserved for errors/destructive actions.
- Backgrounds must remain calm. Prefer **color.surface.*** and **color.brand.primarySoft** for subtle emphasis.
- Always meet accessibility contrast: body text on surface should be WCAG AA or better. If a token fails, adjust token values in CH06 rather than per-screen overrides (see CH32).

- Use semantic tokens for status (success/warning/danger/info) - never invent new status colors in a feature.

3. Typography

Handz uses a two-font system: a bold display face for headings and a clean sans for body/UI. This keeps it modern but distinct. Font usage is tokenized so it can be swapped later if needed.

3.1 Font families (locked for V1)

- **Display / headings:** Bebas Neue (all-caps friendly, punchy).
- **Body / UI:** Nunito Sans (high readability, friendly, modern).
- **System fallback:** iOS San Francisco (SF Pro) if custom fonts fail to load; do not block app usage if fonts fail.

3.2 Type scale (iOS points)

Token	Meaning / Usage
<code>type.display.h1</code>	34/38, tracking +1% — hero headings (rare; onboarding only).
<code>type.display.h2</code>	28/32, tracking +1% — screen titles (most top-level screens).
<code>type.display.h3</code>	22/26, tracking +0.5% — section headers inside a screen.
<code>type.ui.title</code>	17/22 — primary labels (cards, list headers).
<code>type.ui.body</code>	15/20 — standard reading text.
<code>type.ui.bodySmall</code>	13/18 — helper text, captions.
<code>type.ui.meta</code>	11/14 — timestamps, subtle counters.
<code>type.ui.mono</code>	13/18 — optional: IDs/diagnostics in debug builds only (see CH36).

3.3 Text rules

- Headings use Bebas Neue; **avoid** using it for paragraphs or long labels.
- Body text uses Nunito Sans; maintain generous line height for readability.
- Sentence case for UI labels (not ALL CAPS), except display headings that are intentionally styled.
- Numbers: use tabular numbers where supported for timers and counters (Practice mode).
- Do not truncate critical labels; if truncation is unavoidable, provide secondary line or tooltip.

4. Spacing & Layout

Handz uses a simple 4pt spacing grid. Padding and component sizes are standardized so screens feel consistent.

4.1 Spacing tokens

Token	Meaning / Usage
space.1	4pt — micro spacing.
space.2	8pt — tight spacing (chips, small gaps).
space.3	12pt — standard spacing between related elements.
space.4	16pt — default screen padding (most screens).
space.5	24pt — section breaks.
space.6	32pt — major breaks (between major sections).

4.2 Shape tokens

Token	Meaning / Usage
radius.sm	10pt — chips, small cards.
radius.md	14pt — buttons, inputs, cards.
radius.lg	18pt — modals, large sheets.
border.hairline	1pt — standard border.
shadow.elev1	Subtle card shadow; use sparingly, prefer borders.
shadow.elev2	Bottom sheets / modals.

4.3 Screen layout rules

- All screens are portrait-only (see CH02).
- Default content inset: **16pt** left/right; top inset respects Safe Area.
- Primary CTA placement: bottom sticky bar for creation flows (builder, practice setup) when appropriate.
- Lists: use grouped sections with subtle separators; avoid heavy dividers.
- Use whitespace to separate ideas; never cram multiple dense controls into one viewport without grouping.

5. Iconography & Illustration

Icons should be consistent and modern. Prefer SF Symbols on iOS for speed and consistency. Use filled icons for active states and outline icons for inactive states.

5.1 Icon rules

- Stroke weight: match SF Symbols default; avoid mixing line styles.
- Icon size: 20pt for toolbar, 24pt for primary actions; 16pt for inline.

- Color: inactive uses color.text.tertiary; active uses color.brand.primary or color.text.primary depending on context.
- Avoid illustration-heavy onboarding in V1; keep it clean. Use small friendly mascot-style touches only if they do not add layout complexity.

5.2 Brand mark usage (logo / app icon)

The app icon uses the existing black-and-white Handz mark (no color fill) on a Primary Orange background (#FF7F11). No additional text in the app icon. Maintain generous padding so the mark breathes inside iOS icon masks.

6. Motion & Haptics

Motion should feel fast and functional. Use subtle animation to communicate state changes (expand/collapse, success, errors). Avoid long transitions. Use haptics to reinforce key actions (add node, complete set, successful save).

6.1 Motion tokens

Token	Meaning / Usage
<code>motion.fast</code>	120ms — button press feedback, micro transitions.
<code>motion.base</code>	200ms — expand/collapse, modal fade/slide.
<code>motion.slow</code>	320ms — rare: onboarding transitions only.
<code>easing.standard</code>	Ease-out for entrances; ease-in-out for toggles.

6.2 Haptic rules (iOS)

- Light impact: add move / add branch / reorder confirm.
- Success notification: flow saved, session completed, upgrade success.
- Warning notification: hitting soft cap, offline warning.
- Error notification: save failed, invalid action blocked.
- Never overuse haptics; no haptic on every scroll/tap.

7. Component Library (V1)

All UI is built from a shared component set. If a new component is required, add it here (CH06 revision) before implementing one-off UI.

7.1 Buttons

- **Primary Button:** filled, background color.brand.primary, label color.surface.base; radius.md; height 48pt.
- **Secondary Button:** outline (border.subtle), background transparent or surface.alt; label color.text.primary.

- **Tertiary Button:** text-only; label color.brand.primary; used for low-priority actions.
 - **Destructive:** text or outline using color.semantic.danger; confirm dialogs required for destructive actions (see CH31).
 - **Disabled:** use color.action.disabledBg and color.action.disabledText; keep label readable.
- Button states:** default, pressed (primaryHover), disabled, loading (spinner), success (temporary check).

7.2 Inputs

- **Text Field:** height 44pt; radius.md; border.subtle; background surface.base; placeholder text.tertiary.
- **Multiline:** auto-expands up to 6 lines before scrolling; always show remaining character count only if we add explicit caps.
- **Search Bar:** integrated icon left; clear button right; uses surface.alt background.
- **Dropdown / Picker:** uses bottom sheet on iOS; selected value appears as a pill-row input.

Validation rules:

- Inline error text (text.semantic.danger) appears below field; border becomes danger.
- Do not block typing with modal alerts; errors are shown inline unless submission is attempted.
- Success state may show subtle checkmark for fields that validate asynchronously (e.g., username).

7.3 Chips, Tags, and Pills

- **Tag Pill:** rounded radius.lg; background brand.primarySoft; text primary; used for filters and quick labels.
- **Status Pill:** semantic colors for success/warning/danger/info; small meta text.
- **Branch Label Pill:** used on edges or below nodes; max 24 chars, then wraps to 2 lines.

7.4 Cards and List Rows

- Cards: surface.elevated, border.subtle, radius.md, internal padding space.4.
- List rows: full-width tap target (min 44pt). Use divider border.subtle between rows.
- Swipe actions: use native iOS patterns; destructive swipe is red.

7.5 Modals / Bottom Sheets

- Bottom sheets for pickers, add-move, add-branch, reorder selection; radius.lg; drag handle visible.
- Full-screen modals only when the user is in a focused creation flow (e.g., move editor).
- All modals must support: cancel, close (X), and swipe-to-dismiss where appropriate.
- Keyboard behavior: fields must remain visible; auto-scroll to focused input.

7.6 Toasts / Snackbars

- Use to confirm non-blocking actions: “Saved”, “Link copied”, “Added to library”.
- Position: bottom above tab bar; auto-dismiss 2.5s; include Undo only when meaningful.
- Error toast used only for transient network errors; blocking errors use inline or dialogs.

8. Handz-Specific UI Components

These components are unique to Handz and must be consistent everywhere they appear.

8.1 Move Node Card (Flow Builder)

- Shape: rounded rectangle radius.md; min width 120pt; height auto (2 lines max by default).
- Contents: Move name (title), optional micro-tags row (stance, range) if enabled later.
- Node menu: tap node opens quick actions (Replace, Add Next, Add Branch, Delete).
- Selected state: outline with brand.primary and subtle glow; do not fill the node with orange (too loud).
- Drag: long-press to drag; show haptic on drag start; keep 60fps target.

8.2 Edge / Connector UI

- Solid edge = linear continuation; dashed edge = conditional branch (see CH12).
- Edge label uses Branch Label Pill; position near midpoint; auto-avoid overlaps where possible.
- Tapping the edge opens Sequence Detail Editor entry point (see CH14) if enabled for that edge.

8.3 Canvas Controls

- Zoom in/out buttons bottom-right; show current zoom % on long-press.
- Pan: one-finger drag; select node by tap; lasso selection is NOT required for V1.
- Mini-map is optional; if implemented, it must be toggleable and not always visible.

8.4 Practice Timer Card

- Timer uses large numerals (type.display.h2 or h3 depending on screen).
- Primary controls: Pause/Resume; secondary: Skip; destructive: End (requires confirm).
- Rest screen uses calm background and highlights the upcoming sequence name.
- Completion moment: subtle confetti micro-animation is allowed (<= 1s), but must not lag older devices.

8.5 Paywall & Upsell surfaces

- Paywall is a calm sheet with clear benefits and one primary CTA. Avoid dark patterns.
- Benefits use check-list cards; 3–6 bullets max per screen; deeper scientific detail goes to CH26.
- Always include: restore purchases, manage subscription, privacy link.

9. Standard UI States

9.1 Loading

- Use skeleton loaders for lists/cards; avoid spinners for list fetches.
- Use a spinner only for blocking actions (saving, upgrading).
- Loading must never block navigation back.

9.2 Empty states

- Every empty state includes: (1) short headline, (2) one-sentence explanation, (3) one primary CTA.
- Empty states may use a small icon; no large illustrations required for V1.
- Tone: encouraging, not shamey.

9.3 Error states

- Inline for field errors; toast for transient network issues; modal only for destructive or blocking flows.
- Error copy must say what happened and what to do next. Avoid “Something went wrong” alone.
- Whenever possible, include Retry.

10. Content Style Guide

- Voice: confident, coach-like, supportive. Avoid cringe / corny fight talk.
- Be specific: “Save this flow to practice later” instead of “Proceed”.
- Use plain language: “Branch” can be explained as “If they do X, you do Y”.
- Keep labels short; explanations go in helper text or onboarding overlays.
- Never promise outcomes as guarantees; frame as tools and principles (see CH26).

11. Implementation Guardrails (non-negotiable)

- All colors, typography, spacing are referenced via tokens (central theme file).
- No custom per-screen styling unless it becomes a component and is documented in CH06.
- All new UI patterns must be added to this chapter before building.
- Accessibility is not optional; minimum tap targets and contrast must be met (see CH32).

12. Placeholders / Items to Decide Later (Owner: CH06)

Anything not explicitly locked is a placeholder. Defaults below are recommended but may change in CH06_R2.

- **PLACEHOLDER:** Dark mode support • Options: Off in V1 / Limited / Full • Default: Off in V1 • Decide-by: before App Store submission.
- **PLACEHOLDER:** Illustration/mascot style • Options: none / subtle / full mascot • Default: subtle icons only.
- **PLACEHOLDER:** Sound design • Options: none / subtle / more game-like • Default: none.
- **PLACEHOLDER:** Confetti / celebration intensity • Options: none / subtle / medium • Default: subtle.

13. Acceptance Test Checklist (CH06)

Given/When/Then tests

- **Test 1:** Given any screen, when a primary CTA is present, then only one element uses color.brand.primary prominently and all secondary CTAs are visually quieter.
- **Test 2:** Given a form field validation error, when the user submits invalid input, then the field border and helper text use semantic.danger and the error copy explains the fix.
- **Test 3:** Given a list screen, when data is loading, then skeleton rows appear and the screen remains usable (back navigation works).
- **Test 4:** Given a modal, when the user swipes down or taps Cancel, then the modal dismisses without losing already-saved data.
- **Test 5:** Given the flow builder canvas, when a node is selected, then the selected outline uses brand.primary and does not rely on color fill alone.
- **Test 6:** Given any tappable control, when measured, then it meets minimum 44pt tap target.

Checklist

- Theme tokens exist for: colors, typography, spacing, radius, borders, motion.
- Core components implemented: Button, TextField, Chip/Pill, Card, ListRow, BottomSheet, Toast.
- Handz-specific components implemented: Move Node Card, Branch Label Pill, Canvas Controls.
- Default light theme is applied across all screens (no mixed ad-hoc hex values).
- Accessibility checks: contrast, tap target, dynamic type scaling behaviors defined.

14. Replit Build Prompt (CH06 only)

You are implementing Handz V1. Read HZ-V1-CH00 and HZ-V1-CH06. Implement ONLY CH06.

Goal: Create a reusable design system and component library in the codebase so all later screens can reuse it without guessing.

Tasks:

- 1) Create a Theme/Tokens module:
 - Define tokens for colors, typography, spacing, radius, borders, and motion exactly as listed in CH06.
 - Export both raw tokens and semantic helpers (e.g., colors.text.primary).
- 2) Build core components:
 - Button (Primary/Secondary/Tertiary/Destructive/Disabled>Loading)
 - TextField (single + multiline) with inline validation
 - SearchBar
 - Chip/Pill (Tag/Status/Branch label)
 - Card and ListRow
 - BottomSheet modal pattern
 - Toast/Snackbar
- 3) Build Handz-specific components:
 - Move Node Card (selected state, quick actions hook, draggable-ready)
 - Branch Label Pill

- Canvas Controls (zoom buttons placeholder)

4) Add basic ally:

- Ensure 44pt tap targets
- Use accessible labels
- Avoid low-contrast text

Rules:

- Do not hardcode hex values in components; reference tokens.
- If you must invent anything not in CH06, write it into a PRD Assumptions comment block and STOP.

Deliverables:

- theme/tokens.ts (or .js)
- components/ directory with the components above
- a demo screen that renders every component/state for QA.

15. Troubleshooting Notes (CH06)

- If UI looks inconsistent across screens: search for hard-coded styles; replace with tokens.
- If text truncates unexpectedly: verify lineHeight and allow flex wrapping; avoid fixed heights on text containers.
- If contrast feels weak: adjust token values in CH06 (do not patch per screen).
- If performance drops: ensure shadows are not overused; prefer borders and flat surfaces.