# CH21 — Practice Mode: Active Session

Bundle ID: HZ-V1 • Generated: 2026-01-02

**Doc ID:** HZ-V1-CH21_Practice_Mode_Active_Session_R1

**Revision:** R1 (2026-01-02)

**Status:** Draft

**Depends on:** CH06, CH08, CH12, CH20

**Related:** CH03, CH04, CH05, CH22, CH23, CH24, CH25, CH29, CH30, CH31, CH33

**Supersedes:** None

**Owned Decisions:** Active-session interaction model, timers, controls, interruption handling, rest logic, and user-visible session-state messaging for Practice Mode.

**Open Questions / Placeholders:** Listed explicitly in §12.

This chapter defines the **in-session** behavior of Practice Mode: what the user sees once a practice session starts, how timers/sets advance, what each control does, and how we record “actual duration” and interruptions. Setup (selecting paths, ordering, per-path configs, and entitlements) is owned by CH20. Logging/history is owned by CH22.

## 1. Scope

CH21 covers the **Active Session runtime** for Practice Mode: the main active screen, pause/rest overlays, and all state transitions while a session is running.

- **Included:** timer engine, set progression, rest progression, early completion behavior, skip behavior, end-session behavior, interruptions, wake-lock, UI controls, and in-session messaging.
- **Excluded:** practice setup UI (CH20), subscription/paywall UI (CH25), warning ladder global policy (CH30), and practice logging data schemas (CH22).

Design tokens and UI components must comply with CH06 (Design System) and CH04 (Navigation Map). Plan-state gating must comply with CH08 (Entitlements & Plan States).

## 2. Primary user intent

When a user is practicing, they are usually wearing gloves and moving. The active session must be usable with minimal taps, large targets, and predictable state changes.

- Keep interaction count low (most actions should be 1 tap).
- Never require typing during an active set.
- Avoid ambiguous ‘did it save?’ moments: session state should always be obvious (Running / Paused / Rest / Completed / Interrupted).
- Track **actual duration** and **what was accomplished** (sets completed, sequences completed) rather than only time spent.

# 3. Key definitions (owned elsewhere, referenced here)

Definitions are owned by CH03 (Glossary). This chapter uses them as follows:

- **Sequence Item**: one drilled unit in the session queue (typically a selected path). Created in CH20.
- **Set**: one timed work block for a Sequence Item.
- **Rest**: recovery block between sets or between Sequence Items (depending on config).
- **Actual Duration**: wall-clock time spent from session start until session end, excluding time before session start; includes pauses unless explicitly excluded (rule in §7.7).
- **Interrupted**: session ended early (user ended, app backgrounded too long, crash recovery, etc.) but partial results are preserved.

See: HZ-V1-CH20 §(Queue + Config) and HZ-V1-CH22 §(Logging Model).

# 4. Screens & navigation (in-session)

CH21 introduces 4 runtime surfaces. Routing names are illustrative; CH04 owns final route IDs.

| Surface | Type | When shown | Exit conditions |
|---|---|---|---|
| Practice Active Session | Full screen | Session is running (work set) | Auto-advance to Rest when timer hits 0 or user completes set; user may pause or end |
| Practice Rest | Full screen | Between sets (or between items) depending on config | Auto-advance when rest timer hits 0; user may skip rest or end |
| Practice Paused | Full screen overlay | User pauses during Active or Rest | Resume returns to prior surface; End Practice exits to Session Summary (CH22) |
| Confirm End Practice | Modal | User taps End Practice | Confirm ends session; Cancel returns to prior surface |

**Back behavior:** During an active practice session, the OS/back gesture must not silently discard state. Back triggers Confirm End Practice.

# 5. Active Session UI specification

This section defines the Practice Active Session screen (the work-set surface). It must remain readable at arm’s length and tappable with gloves.

## 5.1 Layout regions

- **Top bar:** back/close (opens Confirm End), session status pill (Running), and optional progress indicator.
- **Primary focus block:** the current drilled Sequence Item (path name) and a compact ‘path preview’ (move list condensed).
- **Timer block:** large countdown (MM:SS) with subtle total time for the set (e.g., 03:47 / 05:00).
- **Set progress:** “Set X of Y” and “Item N of M” (or “Branch N of M” if user language uses that).
- **Controls row:** large buttons: Pause, Completed Set, Skip Set. Secondary actions behind a ‘More’ button to reduce clutter (see §5.4).

- **Safety zone:** bottom padding for iPhone home indicator; all primary controls must remain above it.

### 5.2 Required visible fields

- **Current Item Title** (user-defined name, default generated in CH20).
- **Context subtitle**: source flow name (optional, but recommended).
- **Timer** (counting down).
- **Set counter**: Set {currentSetIndex+1} of {setsForItem}.
- **Queue counter**: Item {queueIndex+1} of {queueLength}.
- **Assumed reps**: show as “Reps: {assumedReps} (assumed)” or “Target reps: {assumedReps} (assumed)” — no live rep counting.
- **Status**: Running.

### 5.3 Button set (primary vs secondary)

| Control | Placement | Tap behavior | Long-press? | Notes |
|---|---|---|---|---|
| Pause | Primary | Pause timer + open Paused overlay | No | Also accessible during Rest |
| Completed Set | Primary | Ends the set immediately and starts Rest (early completion) | No | See §7.2 |
| Skip Set | Primary | Marks set as skipped; starts Rest | No | See §7.3 |
| End Practice | Secondary (More) | Opens Confirm End modal | No | Never ends immediately without confirmation |
| Add Set | Secondary (More) | Adds 1 extra set to current item (or opens picker) | Optional | If included in v1, keep to +1 tap; else placeholder |

**Glove-friendly rule:** Primary buttons must be at least 44pt tall, ideally 52–60pt, with large touch targets and generous spacing.

### 5.4 Minimal clutter rule

To keep the active screen simple, v1 should show only 3 primary buttons. Any additional actions belong in a ‘More’ menu.

- More menu items (v1): End Practice, (optional) Add Set, (optional) View Path Details (read-only).
- More menu must be tappable with gloves and must not require scrolling if possible.

## 6. Rest screen UI specification

Rest is a separate full-screen surface so the user can quickly see: how long to rest and what’s next.

### 6.1 Required visible fields

- **Rest timer**: large countdown (SS or MM:SS depending on length).
- **Next item**: next Sequence Item title + optional small preview.

- **Progress**: show set completion feedback (“Set complete”) and queue position.
- **Status**: Resting.

### 6.2 Rest controls

- **Skip Rest** (primary): ends rest immediately and starts next work set.
- **Pause** (secondary or primary if only two buttons): opens Paused overlay.
- **End Practice** (secondary in More or top bar): opens Confirm End.

## 7. State machine (timers, progression, interruptions)

This section removes all guesswork: exactly when we advance states and how we compute actual duration.

### 7.1 Core states

- **RUNNING_SET**: active work timer counting down.
- **RESTING**: rest timer counting down.
- **PAUSED**: timer halted; an overlay blocks accidental taps.
- **ENDED_COMPLETED**: session finished all planned sets/items.
- **ENDED_INTERRUPTED**: session ended early (user ended, forced stop, crash recovery).

### 7.2 Early completion (workout-tracker style)

Early completion means the user finishes the set before the timer reaches 0. This must be 1 tap: “Completed Set.”

- On tap **Completed Set**: stop set timer, mark set as completed, record elapsed work time for that set, then transition to RESTING (or next set if rest=0).
- The remaining time is not carried forward. It is simply saved as ‘unused set time’.
- UI feedback: immediate confirmation (haptic + ‘Set complete’), then Rest screen appears.
- If user taps Completed Set accidentally: allow **Undo** toast for 3 seconds (v1 recommended). If undone, return to RUNNING_SET with the prior remaining time.

### 7.3 Skip set

Skip is used when a set is not productive or the user wants to move on.

- On tap **Skip Set**: mark current set as skipped; record elapsed work time; transition to RESTING (or next work set if rest=0).
- Skipped sets count toward “planned sets attempted” but not toward “sets completed” in summaries.
- UI feedback: show “Set skipped” message and what’s next.

### 7.4 Timer reaches 0 (auto-complete)

- When work timer reaches 0: automatically complete the set (as if Completed Set) and transition to RESTING.

- When rest timer reaches 0: automatically start the next work set.
- Timer ticks must clamp at 0 (never negative).

## 7.5 Progression rules

- Each Sequence Item has a planned number of sets (from CH20).
- For each item: run set -> rest -> run set -> rest ... until sets planned are exhausted.
- After final set of an item completes: transition to RESTING (if configured between items) or directly to next item's RUNNING_SET.
- When the last planned set of the last item completes: transition to ENDED_COMPLETED and hand off to CH22 Session Summary.

## 7.6 Confirm End Practice behavior

- End Practice is always a 2-step action: tap End -> confirm modal.
- Confirm modal copy (v1): "End practice session? Your progress will be saved as Interrupted." Buttons: "Keep Practicing" (default) and "End Session".
- If confirmed: transition to ENDED_INTERRUPTED and hand off to CH22 summary (with status=interrupted).

## 7.7 Actual duration and pause accounting

We agreed to track **actual duration**. This chapter defines the canonical computation so CH22 can store it consistently.

- **SessionStartTimestamp** recorded at the moment the user enters RUNNING_SET for the first time.
- **SessionEndTimestamp** recorded when the session transitions to ENDED_COMPLETED or ENDED_INTERRUPTED.
- **ActualDuration** = SessionEndTimestamp - SessionStartTimestamp (includes pause time).
- **ActiveWorkDuration** = sum of elapsed time in RUNNING_SET across all sets (excludes rest and pause).
- **RestDuration** = sum of elapsed time in RESTING.
- **PausedDuration** = ActualDuration - ActiveWorkDuration - RestDuration.

Rationale: including pause time in ActualDuration reflects what the user actually spent. CH22 can present both ActualDuration and ActiveWorkDuration to avoid confusing users.

## 7.8 Backgrounding & interruptions (iOS)

Because users might lock their phone or switch apps mid-session, we define a strict policy:

- If app goes to background while RUNNING_SET or RESTING: immediately pause the session (enter PAUSED) and show a "Paused due to leaving app" banner on return.
- If app remains backgrounded longer than a threshold, the session should be saved as Interrupted on next app open. **PLACEHOLDER: BackgroundTimeoutSeconds** (Owner: CH21). Default suggestion: 300s (5 minutes).
- If the app crashes, on next launch show "Resume interrupted session?" with options Resume or Discard. (Resume continues timers from remaining time at last persisted tick.)

Implementation note is owned by CH28 (Offline & Sync), but user-visible behavior is owned here.

# 8. Haptics, sound, and screen wake behavior

Practice sessions must feel ‘addictive’ and clear without requiring constant staring at the screen.

## 8.1 Haptics (v1)

- Set complete (auto or manual): light haptic.
- Rest complete / next set starts: light haptic + optional short chime.
- Error (e.g., can’t continue due to entitlement): warning haptic (rare in-session).

## 8.2 Audio cues (v1)

- Optional: last 3 seconds countdown ticks (off by default). **PLACEHOLDER: AudioCountdownDefault** (Owner: CH21).
- Optional: set-start and rest-start short sound (on by default). **PLACEHOLDER: AudioCueDefault** (Owner: CH21).
- All audio cues must respect device mute switch unless user enables ‘Always play cues’ (later).

## 8.3 Wake lock

- During RUNNING_SET and RESTING, keep screen awake to avoid interruption.
- If user manually locks screen, treat as backgrounding: session pauses.

# 9. Edge cases & error states

Error states are owned in detail by CH31, but practice-specific edge cases are defined here so behavior is unambiguous.

## 9.1 Timer drift / device lag

- Timers must be wall-clock-based (compare now() to start timestamp) rather than relying only on setInterval ticks, to avoid drift when the JS thread is busy.
- When drift detected (e.g., app resumes), recompute remaining time from wall-clock and clamp >= 0.

## 9.2 Low battery mode / performance

- If animations cause stutter, timers and controls must remain responsive.
- Do not render heavy flow diagrams in-session; show only a compact path preview.

## 9.3 Entitlement mismatch mid-session

Entitlements are checked before session start (CH20 + CH08). Mid-session mismatch should be rare but must be safe.

- If entitlement is lost mid-session (e.g., trial ended): allow session to finish; apply restrictions next time. (Avoid disruptive in-session paywalls.)
- If credit-based session starts and user has 0 credits: block start in CH20, not here.

## 9.4 Navigation accidents

- If user swipes from edge or tries to back out: intercept and show Confirm End.

- If OS call interrupts (phone call): auto-pause and show paused banner afterward.

## 10. Data emitted to logging layer (contract)

CH22 owns storage, but CH21 must define what it emits so implementations match.

- For each set: start timestamp, end timestamp, planned duration, actual elapsed duration, completion status (completed/skipped/auto-completed), and whether it ended early.
- For rest segments: start timestamp, end timestamp, planned rest duration, actual rest duration, skippedRest boolean.
- For the session: status completed vs interrupted, actual duration, active work duration, rest duration, paused duration, queue summary counts.

  If CH22 uses a single ‘session timeline’ array, these events become timeline entries. Exact schema belongs to CH22.

## 11. UX copy (in-session)

Final copy belongs to CH15 (Copy Pack). This chapter defines the required messages so CH15 can write them.

- Running state label: “Practice” or “Live Practice”.
- Completed Set feedback: “Set complete”.
- Skipped Set feedback: “Set skipped”.
- Rest label: “Rest”.
- Auto-advance message (optional): “Next: {ItemTitle}”.
- Pause banner on resume: “Paused while you were away”.
- End confirm: “End session? Progress will be saved as interrupted.”
- Completion: “Session complete”.
- Interruption: “Session saved”.

## 12. Placeholders owned by CH21

- **PLACEHOLDER: BackgroundTimeoutSeconds** • Owner: CH21 • Options: 120 / 300 / 600 • Default: 300 • Decide-by: before implementing resume flow.
- **PLACEHOLDER: AudioCountdownDefault** • Owner: CH21 • Options: on/off • Default: off • Decide-by: CH06 motion/audio review.
- **PLACEHOLDER: AudioCueDefault** • Owner: CH21 • Options: on/off • Default: on • Decide-by: CH06 motion/audio review.
- **PLACEHOLDER: UndoCompletedSetToast** • Owner: CH21 • Options: on/off • Default: on • Decide-by: usability test; glove accuracy.
- **PLACEHOLDER: AddSetInSession** • Owner: CH21 • Options: include / exclude in v1 • Default: include as +1 only • Decide-by: complexity check with build.

# 13. Acceptance tests (Given / When / Then)

- **Timer auto-complete:** Given a RUNNING_SET at 00:01 remaining, when the timer reaches 0, then the set is marked completed and the UI transitions to Rest within 1 second.
- **Completed Set early:** Given a RUNNING_SET with 03:00 remaining, when the user taps Completed Set, then the set ends immediately, elapsed duration is recorded, and Rest begins.
- **Undo early complete:** Given the user tapped Completed Set, when they tap Undo within 3 seconds, then the app returns to RUNNING_SET with the prior remaining time.
- **Skip set:** Given a RUNNING_SET, when the user taps Skip Set, then the set is recorded as skipped and Rest begins.
- **Rest auto-advance:** Given a RESTING timer at 00:01, when it reaches 0, then the next RUNNING_SET begins automatically.
- **Skip rest:** Given RESTING, when user taps Skip Rest, then the next RUNNING_SET begins immediately and rest is recorded as skipped.
- **Pause:** Given RUNNING_SET, when user taps Pause, then timer stops and Paused overlay appears; when user taps Resume, timer continues from the same remaining time.
- **Background pause:** Given RUNNING_SET, when app backgrounds, then session pauses; on return, Paused overlay appears and timer is unchanged.
- **End practice confirm:** Given an active session, when user taps End Practice and confirms, then session ends as Interrupted and navigates to CH22 summary.
- **Completion:** Given last set of last item completes, when it completes, then session ends as Completed and navigates to CH22 summary.
- **Actual duration:** Given a session that includes 30 seconds paused, when it ends, then ActualDuration includes pause time and PausedDuration reports approx 30 seconds.

### Checklist (quick)

- Timers clamp at 0 and do not drift noticeably.
- Primary controls are glove-friendly and always visible.
- No typing required in-session.
- Back gesture always triggers Confirm End (never silent exit).
- Resume after background behaves predictably (paused).
- Session emits per-set and per-rest events with timestamps.
- ActualDuration / ActiveWorkDuration / RestDuration / PausedDuration computed correctly.

# 14. Replit build prompt (chapter-only)

**Copy/paste prompt:**

```
Implement Handz V1 — CH21 Practice Mode: Active Session (Bundle ID HZ-V1).
Follow CH00 rules: do not guess; if any assumption is needed, write it in a 'PRD
Assumptions' comment block and stop that feature.
```

```
Dependencies: CH06 (Design tokens), CH08 (Entitlements), CH20 (Practice Setup produces a
session queue/config).

Build the in-session runtime UI and state machine:
1) Screens: PracticeActiveSession, PracticeRest, PracticePaused (overlay),
ConfirmEndPractice (modal).
2) State machine: RUNNING_SET, RESTING, PAUSED, ENDED_COMPLETED, ENDED_INTERRUPTED.
3) Timer engine must be wall-clock based (avoid drift). Clamp at 0.
4) Controls:
- Primary on Active: Pause, Completed Set, Skip Set.
- Rest: Skip Rest (+ Pause available).
- End Practice always requires confirmation; end saves as Interrupted.
5) Early completion: Completed Set ends the set immediately; record elapsed time. Show an
Undo toast for 3 seconds (placeholder-controlled) that restores prior remaining time.
6) Background behavior (iOS): on background, auto-pause. If background >
BackgroundTimeoutSeconds (placeholder default 300s), mark session as interrupted on next
open and offer Resume/Discard.
7) Emit session timeline events to the logging layer interface (CH22 will persist). At
minimum emit per-set and per-rest events with timestamps, completion status, and elapsed
times.
8) Keep screen awake during RUNNING_SET and RESTING.
9) Add haptics for set completion and rest completion (light haptic). Audio cues are
placeholders.

Deliverables:
- React Native components/screens with navigation wiring.
- A session controller (hook or state machine) with pure functions for transitions.
- Unit tests for transition logic (Given/When/Then cases above).
- Manual QA notes matching acceptance checklist.
```

# 15. Troubleshooting notes (chapter-specific)

- **Timer goes negative:** ensure remaining time clamps at max(0, computed). Avoid decrementing state below 0.
- **Timer drifts when phone is busy:** switch from interval-based decrement to wall-clock delta calculation.
- **Buttons mis-tap with gloves:** increase hitSlop/touch target; reduce number of primary controls.
- **App resumes and jumps states:** persist state snapshots (current item index, set index, state, remaining seconds) on each transition; on resume validate and repair.
- **Back gesture exits session:** intercept navigation events; always show Confirm End.
- **Session logs look wrong:** verify that each set/rest emits start/end timestamps and status; ensure early complete sets record elapsed time correctly.