

CH24 — Maintenance System

Revision: R1 (2026-01-02) • Status: Draft • Bundle ID: HZ-V1

Purpose: Define the Maintenance System that helps users retain mastered paths/gameplans over time without overwhelm, including UI, scheduling, notifications hooks, overload prevention, data model, and acceptance tests.

Required Front-Matter (CH00 §2)

- **Doc ID:** HZ-V1-CH24_Maintenance_System_R1
- **Revision:** R1 (2026-01-02)
- **Status:** Draft
- **Depends on:** CH03, CH08, CH20, CH21, CH22, CH23, CH27, CH28, CH31
- **Related:** CH06, CH15, CH26, CH29, CH33, CH36, CH37
- **Supersedes:** N/A
- **Owned Decisions:** Maintenance data model + UX, overload prevention rules, maintenance scheduling rules, maintenance session entry/exit logic (as it relates to maintenance), maintenance hub screen inventory and routing, maintenance copy pack for this chapter.
- **Open Questions / Placeholders:** See §12 (Placeholder Registry for CH24).

1. Scope and Non-Goals

This chapter specifies the **Maintenance System** for Handz V1: a set of screens, rules, and scheduling logic that helps users maintain recall and execution readiness for previously planned and practiced paths/gameplans. Maintenance is not a separate practice engine; it orchestrates sessions using the existing Practice Mode pipeline and logs with additional metadata.

In scope:

- Maintenance Hub (today's maintenance, backlog, recommendations, search/filter).
- Selecting maintenance targets (paths, flows, gameplans) and converting them into maintainable units.
- Overload prevention (capacity-based scheduling, backoff rules, user control).
- Maintenance scheduling (days/time windows, reminders, snooze, reschedule).
- Maintenance sessions (entry, progression, completion, interruption, carryover).
- Maintenance metrics that emphasize progress without shame ("On track" vs "broken").
- Dependencies and hooks into Notifications (CH27), Offline/Sync (CH28), Error States (CH31).

Non-goals (explicitly out of scope for CH24):

- Scientific claims, study citations, and wording that implies quantified performance improvement (owned by CH26).
- Exact paywall copy, pricing, and trial terms (owned by CH25, CH15).
- Full mastery scoring algorithms or combat-readiness modeling beyond what is necessary to schedule maintenance (owned by CH23).
- Communities, gym assignments, coach dashboards (future versions).

2. Definitions (CH03 References)

This chapter uses terms defined in CH03. If a definition conflicts, CH03 is authoritative. Key terms used here:

- **Flow:** a user-authored decision structure composed of moves and optional sequences (see CH12–CH14).
- **Path:** a specific traversal through a flow from a start node to an end node (owned by CH23).
- **Gameplan:** a named set of one or more paths (possibly across multiple flows) selected for mastery/maintenance (owned by CH23).
- **Maintenance Item:** a maintainable unit derived from a path or a gameplan subset, scheduled for periodic drill.
- **Due:** a maintenance item that is recommended to be drilled within a time window.
- **Backlog:** overdue maintenance items not yet drilled.
- **Capacity:** user-selected daily/weekly limit that prevents overload.
- **Snooze:** user action to push a maintenance item to later without losing progress framing.

3. Product Intent and User Outcomes

Maintenance exists to solve a V1 risk: users may create many flows/paths, drill them briefly, and then forget or feel overwhelmed by the growing set. Maintenance provides a structured but forgiving way to keep key paths "alive" over time.

- Help users prioritize what matters (most used, most important, soonest needed).
- Prevent maintenance overload via explicit capacity settings and soft warnings.
- Keep motivation high by avoiding punitive streak language; emphasize "resume" not "restart".
- Support both simplicity (one tap "Do today's maintenance") and complexity (custom scheduling per gameplan).

4. Core UX Model

Maintenance is presented as a lightweight layer above Practice Mode:

- **Users select targets** (paths/flows/gameplans) to maintain.

- **Handz generates a schedule** based on user capacity and an adjustable maintenance cadence model.
- **Each maintenance item launches a maintenance session** (Practice Mode variant) which consumes the same entitlements/credits as practice (See CH08; practice paywall lock in CH00 Decision Log).
- **Completion updates a maintenance record** and refreshes "next due" predictions.

5. Screen Inventory and Routing (CH04/CH05 Alignment)

CH05 provides the global screen list. This chapter defines the maintenance-specific screens and their routes. Route IDs are suggested and must be reconciled with CH04 when implemented.

5.1 Maintenance Hub (Route: /maintenance)

Primary entry point. Shows today's maintenance, backlog, and recommended items with fast actions.

Primary entry points into Maintenance Hub:

- Bottom tab: Practice tab includes a sub-tab or CTA "Maintenance" (as previously decided: Practice tab exists; maintenance may live here or as its own tab depending on CH04).
- From Session Complete screen (CH22): CTA "Add this path to Maintenance" or "Add to Gameplan" then "Maintain".
- From Gameplan detail (CH23): CTA "Maintenance settings".
- From Home/Dashboard: card "Today's Maintenance" (if enabled).

5.2 Maintenance Plan Builder (Route: /maintenance/setup)

Wizard-style flow for first-time setup and for creating/editing a maintenance plan for a gameplan.

5.3 Maintenance Item Detail (Route: /maintenance/item/:id)

Drill preview + controls: start session, snooze, edit cadence, swap target, view history.

5.4 Maintenance Session (Route: /maintenance/session)

A Practice Mode session launched in "Maintenance" context. Uses the same core UI as Practice Active (CH21) but includes maintenance-specific header context and logging metadata.

5.5 Maintenance Settings (Route: /maintenance/settings)

Global settings: capacity, preferred days, reminder windows, default cadence model, overload behavior.

6. Maintenance Hub — Detailed Spec

6.1 Layout (mobile portrait)

Top area:

- **Header:** "Maintenance" + subtle subtext showing today's date and status (e.g., "2 due today • 1 overdue").
- **Primary CTA:** "Start Today's Maintenance" (starts a bundled maintenance session of today's due items) OR if nothing due: "Browse Recommendations".
- **Status pill:** "On Track" / "Behind" / "Paused" (definition in §10).

Main sections (scroll):

- **Due Today** list (0–N items).
- **Overdue** list (collapsed by default if empty).
- **Recommended** list (always present; personalization-driven).
- **Browse** entry (search + filters + sort).

6.2 Maintenance item card design

Each card must be readable at a glance and actionable with one tap.

- **Title:** Gameplan name or Path name (user-renamable).
- **Subtitle:** e.g., "3 paths" or "1 path" + tags (optional).
- **Next due:** "Due today" / "Due tomorrow" / "Overdue 3d" (computed).
- **Quick action:** "Start" button (starts maintenance session for that item only).
- **Secondary action:** kebab menu: Snooze, Edit cadence, Remove from maintenance, View history.

6.3 Browse panel (search + filters)

Goal: help users avoid maintenance overload by letting them focus on what matters.

- **Search:** by flow/gameplan/path name.
- **Filters:** Most used, Most recently practiced, Most recently added, Overdue, Due soon, By style tag (if available), By folder (if flows have folders).
- **Sort:** Due date, Urgency score, Recently practiced, Alphabetical.
- **Bulk select:** enter select mode; actions: Add to maintenance, Remove, Snooze, Set cadence (bulk), Move to different gameplan (if supported).

7. Maintenance Plan Builder — Detailed Spec

7.1 When the user sees the builder

- First time a user taps Maintenance Hub and has no maintenance targets configured.

- When user taps "Create Maintenance Plan" from Gameplan detail.
- When user taps "Edit Maintenance" from Maintenance Hub settings.
- When user imports a flow/gameplan and chooses "Add to Maintenance" during import conflict resolution (CH19).

7.2 Wizard steps (minimum viable)

Step 1 — Choose what to maintain

- Pick one: **Gameplan** (recommended), **Whole Flow**, or **Specific Paths**.
- UI rule: show simplest choice first; advanced options behind "More options".
- If user chooses Gameplan: pick existing or create new (name required).
- If user chooses Flow: system will propose derived paths (owned by CH23; this chapter only consumes that output).
- If user chooses Specific Paths: show path picker with preview.

Step 2 — Choose goal type (affects cadence)

- Options (copy):
 - "Remember it" (light maintenance)
 - "Use it under pressure" (heavier maintenance)
 - "Fight camp" (time-bounded; optional future placeholder)
- Rule: the goal type changes *suggested* cadence, but users can override.

Step 3 — Set your capacity (overload prevention)

- User sets: maintenance days per week, target sessions per week OR target minutes per week (choose one primary input to reduce cognitive load).
- User sets optional daily cap: "No more than X items/day".
- Default: conservative and editable later (placeholder numbers until decided; see §12).

Step 4 — Reminders (opt-in)

- Ask for preferred reminder time window (not exact time): "Morning", "Afternoon", "Evening". Exact scheduling is owned by CH27.
- Allow "No reminders" option without guilt copy.
- If notifications permission not granted, show a soft explanation and a button "Enable reminders" (routes to system prompt if allowed).

Step 5 — Review and create

- Show summary: items selected, estimated weekly load, schedule preview (e.g., Mon/Wed/Fri: 2 items).
- Buttons: "Create Plan" and "Back".

- Include note: "You can change this anytime."

8. Scheduling and Overload Prevention

8.1 Design principles

- User-controlled capacity beats algorithmic ambition.
- Backlog must not spiral into shame: the system should gracefully degrade when the user misses days.
- Maintenance should prefer fewer items done consistently over many items done once.
- All "recommended" scheduling is editable; we never imply medical/scientific certainty (scientific claims handled in CH26).

8.2 Capacity model (V1)

Maintenance uses a simple capacity model composed of:

- **Weekly capacity:** user-set target number of maintenance items per week OR minutes per week.
- **Daily cap:** optional hard cap for the UI (do not schedule more than N items/day).
- **Carryover policy:** defines how missed items roll forward.

8.3 Carryover policy options (choose one for V1; others as future)

V1 recommended default: **Soft Carryover with Backlog Compression.**

- **Soft carryover:** missed items become overdue but are not automatically stacked onto the next day beyond the daily cap.
- **Backlog compression:** the system proposes a small "catch-up" suggestion like "Add 1 extra item today" but never auto-increases load unless user taps "Catch up".
- **Explicit skip:** user can mark an overdue item as "Skip for now" without penalty; it re-enters recommendations later.

8.4 Urgency scoring (for sorting, not for shaming)

Maintenance items get an internal urgency score used for ordering Due/Recommended lists. The score should not be exposed as a harsh number; instead show simple labels.

- Inputs may include: last practiced date, goal type, user importance rating (optional), recent usage frequency, upcoming "needs" date (future placeholder).
- Output buckets for UI: "Due", "Soon", "Later".

8.5 Overload prevention UI behaviors

- If the user tries to add too many items to a plan relative to their capacity, show a **soft warning**: "This is a lot. Want to start with fewer?" with options: "Keep" / "Reduce".
- If daily scheduled items exceed cap due to user edits, highlight the day and offer auto-balance: "Spread across week".

- If backlog > threshold (placeholder), show a supportive banner: "You're behind - pick 1 item today and you're back on track."

9. Maintenance Sessions and Entitlements

9.1 Launching a maintenance session

Starting maintenance launches Practice Mode with a special context flag:

```
Context = maintenance; source = maintenance_item_id; target_set = derived paths;
mode = user-selected drill mode (Practice vs Review).
```

9.2 Entitlement rules (must obey CH08)

Maintenance sessions must obey the same entitlement and credit model as Practice Mode. CH00 locks: practice is paywalled; Free receives 3 monthly practice credits usable only on saved flows; Free inbox items are view-only and cannot be practiced.

- If user is **Guest**: maintenance screens can be browsed as demo, but any attempt to create a plan or start a session must prompt account creation (See CH07/CH08).
- If user is **Free**: starting a maintenance session consumes a practice credit (if available) and is allowed only for targets that are saved flows (not inbox). If no credits remain, show paywall.
- If user is **Pro/Trial**: unlimited maintenance sessions; still enforce anti-abuse / safety limits from CH30.

9.3 Maintenance vs Practice UI differences

- Header label: "Maintenance" (so users know this is part of retention).
- Session goal summary: "Today: 2 items" / "This item: 1 path".
- Post-session summary includes maintenance-specific outcomes: items maintained, next due estimate, backlog delta.

10. Status Framing (No-Shame Design)

The maintenance system must encourage consistency without punishing missed days.

10.1 Status pills (Maintenance Hub)

- **On Track**: user completed at least one due item within the last X days OR backlog below threshold.
- **Behind**: backlog exists beyond threshold OR no maintenance completed within window.
- **Paused**: maintenance plan paused by user; no reminders; hub shows "Resume" CTA.
- **Not Set Up**: no maintenance plan; hub shows setup CTA.

10.2 Copy rules

- Never say: "You failed", "You broke your streak".
- Prefer: "Pick up where you left off", "Resume", "You're one session away from being back on track".

- Use neutral metrics: "items completed" and "items due"; keep streaks optional and secondary (CH22).

11. Data Model (V1)

This defines logical entities. Storage specifics live in CH29; naming can be translated to Firebase collections/documents.

11.1 Entities

MaintenancePlan

A plan groups maintenance targets and capacity settings. Typically one global plan + optional per-gameplan overrides.

```
{ id: string, user_id: string, name: string, // e.g., "Default Maintenance" or  
"Fight Camp A" scope: "global" | "gameplan", gameplan_id?: string, goal_type:  
"remember" | "pressure" | "custom", is_paused: boolean, created_at: ISODate,  
updated_at: ISODate }
```

Maintenanceltem

A maintainable unit, usually representing a path or a group of paths within a gameplan.

```
{ id: string, plan_id: string, label: string, // user-editable display name  
target_type: "path" | "path_set" | "flow" | "gameplan", target_ref: { ... }, //  
references owned by CH23/CH16 importance: "low" | "med" | "high" | null,  
cadence_model: "auto" | "fixed" | "custom", cadence_days?: number, // if fixed  
(placeholder) next_due_at: ISODateTime, last_done_at?: ISODateTime, created_at:  
ISODateTime, updated_at: ISODateTime, is_archived: boolean }
```

MaintenanceSchedule

User preferences that constrain scheduling.

```
{ id: string, user_id: string, preferred_days: ["mon","wed","fri"], // local time  
time_window: "morning"|"afternoon"|"evening"|null, weekly_capacity_type:  
"items"|"minutes", weekly_capacity_value: number, // e.g., 6 items or 45 minutes  
daily_cap_items?: number, carryover_policy: "soft_compress", reminders_enabled:  
boolean, updated_at: ISODateTime }
```

MaintenanceLog

One completion record per maintenance session or per item (choose one; V1 can do per session with item breakdown).

```
{ id: string, user_id: string, session_id: string, // links to practice log (CH22)  
completed_at: ISODateTime, items: [ { maintenance_item_id: string, status:  
"done"|"partial"|"skipped", notes?: string } ], backlog_before: number,  
backlog_after: number }
```

12. Placeholder Registry (CH24-Owned)

Anything not locked must remain a placeholder in CH24 until decided and then also logged in CH00 §6.

- **PLACEHOLDER: MAINT_CAPACITY_DEFAULT** • Owner: CH24 • Options: (A) 3 items/week, (B) 6 items/week, (C) minutes-based default • Default: A • Decide-by: before implementation
- **PLACEHOLDER: OVERDUE_THRESHOLD** • Owner: CH24 • Options: 3 items / 7 items / 14 items • Default: 7 items • Decide-by: before release QA

- **PLACEHOLDER: CADENCE_PRESETS** • Owner: CH24 • Options: (A) 3/7/14 days, (B) 2/5/10 days, (C) user-defined only • Default: A • Decide-by: before UI copy lock
- **PLACEHOLDER: GLOBAL_VS_PER_GAMEPLAN_PLANS** • Owner: CH24 • Options: (A) global only in V1, (B) allow per-gameplan overrides, (C) multiple plans • Default: B • Decide-by: implementation
- **PLACEHOLDER: PRO_GATING_MAINTENANCE_AUTOPLAN** • Owner: CH25/CH24 • Options: (A) autoplan for all, (B) autoplan Pro-only, (C) reminders Pro-only • Default: A • Decide-by: pricing review

13. Error States and Offline Behavior (CH28/CH31)

Maintenance must degrade gracefully. Detailed global behavior is owned by CH28 and CH31; this section defines maintenance-specific cases to be implemented there.

- **Offline browse:** show cached maintenance items; hide "Start" if practice cannot start offline (depends on CH28 practice offline rules).
- **Sync conflict:** if a target flow/path changed since last schedule calculation, show "Needs update" badge; tapping opens item detail explaining what changed and offers "Recompute".
- **Missing target:** if a flow/path is deleted, maintenance item becomes "Broken"; offer: Remove from maintenance or Replace target.
- **Timezone shift:** if device timezone changes, maintenance due dates should rebase to local midnight boundaries to avoid instant overdue spikes.
- **Notification permission denied:** show a non-blocking banner "Reminders are off" with action "Enable".

14. Acceptance Tests (Given / When / Then)

- **MT-01 Setup CTA:** Given a new user with no maintenance plan, when they open Maintenance Hub, then they see "Set up maintenance" CTA and cannot see a due list.
- **MT-02 Create plan:** Given user completes plan builder with capacity and selected targets, when they tap "Create Plan", then plan and items persist and hub shows due/recommended lists.
- **MT-03 Start item session:** Given a due maintenance item, when user taps "Start", then app navigates to Maintenance Session and session header shows "Maintenance" and target name.
- **MT-04 Free entitlement:** Given a Free user with 0 practice credits, when they tap "Start" on a maintenance item, then they see paywall and the session does not start.
- **MT-05 Free credit consumption:** Given a Free user with 1 practice credit and item targets a saved flow, when they start maintenance and complete the session, then credits decrement by 1 and maintenance log is saved.
- **MT-06 Snooze:** Given an overdue item, when user taps Snooze (e.g., "Tomorrow"), then next_due_at updates and the item moves out of Overdue list.
- **MT-07 Backlog compression:** Given backlog exceeds threshold, when user opens hub, then a supportive banner appears offering "Add 1 catch-up item" and selecting it increases today's session bundle by 1 without exceeding daily cap.
- **MT-08 Target changed:** Given a maintenance item references a path that is no longer valid after flow edits, when hub loads, then item is marked "Needs update" and tapping it offers recompute/replace.
- **MT-09 Pause/resume:** Given a maintenance plan is paused, when hub opens, then status pill shows "Paused" and CTA shows "Resume"; when resumed, due dates recompute based on current date.
- **MT-10 No-shame copy:** Given user misses a week, when they open hub, then copy avoids punitive wording and offers a simple next step.

14.2 Chapter checklist

- Maintenance Hub renders: Due Today, Overdue, Recommended, Browse.
- Plan Builder completes end-to-end and persists schedule preferences.
- Maintenance Item Detail supports Start, Snooze, Edit cadence, Remove.
- Maintenance Session launches using Practice Mode engine with maintenance context flag.
- Entitlement gating obeys CH08 locks and CH00 global decisions.
- Overload prevention soft warnings implemented.
- Error states and offline behaviors implemented as specified.

15. Replit Build Prompt (Implement CH24 Only)

You are implementing CH24 (Maintenance System) for the Handz V1 PRD Bundle (HZ-V1). Follow CH00 rules: stable IDs, no detail loss, cross-references, and required

header blocks in code comments.
Implement ONLY CH24 screens + logic. Treat referenced chapters as dependencies; do not invent their rules.

Build order:

- 1) Data layer: add MaintenancePlan, MaintenanceItem, MaintenanceSchedule, MaintenanceLog models (Firestore-friendly).
- 2) Screens:
 - MaintenanceHubScreen (/maintenance)
 - MaintenanceSetupWizard (/maintenance/setup)
 - MaintenanceItemDetail (/maintenance/item/:id)
 - MaintenanceSettings (/maintenance/settings)
- 3) Scheduling logic:
 - Compute due_today, overdue, recommended lists.
 - Enforce daily cap and soft carryover with backlog compression.
- 4) Session launch:
 - Start maintenance session by calling existing Practice Mode start flow with context={type:'maintenance', maintenanceItemIds:[...]}. - Apply entitlement gate: guest must sign up; free must have credits; pro unlimited.
- 5) Logging:
 - On session complete, write MaintenanceLog with item statuses and update next_due_at.
- 6) UI copy:
 - Use no-shame wording (CH24 §10).
- 7) Add unit tests / integration tests for MT-01..MT-10.

Stop and ask for confirmation ONLY if you hit a missing dependency that CH24 explicitly references (CH08, CH21, CH22, CH23).

16. Troubleshooting Notes (CH24)

- **Items show as instantly overdue after timezone change:** Rebase next_due_at to local day boundary; store UTC but compute UI buckets in local time.
- **Backlog keeps growing even after sessions:** Verify MaintenanceLog write + next_due_at update happens once per completed item, not once per session only.
- **Free users can practice inbox items via maintenance:** Ensure entitlement gate checks target is a saved flow and disallows inbox-only targets (CH00 lock).
- **Daily cap ignored when user bulk-adds items:** Scheduling function must enforce cap at computation time and when rendering "Today" bundle.
- **Plan edits wipe user custom cadence:** Ensure per-item cadence overrides persist unless user explicitly resets to auto.
- **UI feels punitive:** Audit copy strings against §10.2; remove "failed"/"missed" language.