

# Hybit — Hydration-Powered Virtual Run

*Product Requirements (v1, updated)*

*Updated: November 08, 2025*

---

## 1) Overview & Goals

Hybit turns everyday drinking into a 7-hour virtual run where hydration controls how fast stamina depletes—and therefore how quickly you advance toward a marathon (42.195 km). The app pairs with a Bluetooth “coaster” that detects intake and converts it into stamina/pace effects within a single-day run.

- **Starting stamina = 300 (max 300)**
- **Race duration = 7h (420 min)**
- **Max distance = 42.195 km**
- **No recovery: stamina never increases during a session; it only decreases per penalties.**

Goals

- Make hydration instantly visible and motivating via a simple daily “run.”
- Nudge users to meet intake targets with clear progress feedback.
- Provide a “start → run → finish” flow that tolerates brief connectivity gaps.

## 2) Target Users & Value

- Office workers & students who want to be more productive → hydration-driven micro-goals and a clear daily finish line.

Primary value: each drink measurably affects stamina (and thus pace), creating habit momentum toward 42.195 km.

## 3) Core User Flows

Onboarding (first run)

1. Choose avatar; confirm gender for hydration target (Men 3500 ml, Women 3000 ml).
2. Pair Bluetooth coaster; confirm time sync.
3. Land on “Start your day”; tap Start to begin the 7-hour run.

Daily run (7h session)

- Coaster streams time-stamped intake; app aligns to 10-minute scoring windows.
- Stamina only depletes per penalty rules below. Missing interval data = 0 ml (max penalty); corrupt data is skipped and logged.
- Distance advances via a linear stamina  $\rightarrow$  distance mapping; total distance capped at 42.195 km.
- At 7h, show Finish with total distance, best run, and Start new race.

Edge states

- If the user hasn't tapped Start, remain on Start with connection prompts.
- Before scoring: Bluetooth + coaster connected + time synced.

## 4) Functional Requirements

### 4.1 Hydration & Stamina Engine (No-Recovery + Penalty Table)

- Scoring cadence: every 10 minutes (first interval has a special rule).
- Targets: Men 3500 ml; Women 3000 ml per 7h session.
- Definitions: For each 10-min interval after the first, let  $R$  = required ml;  $A$  = actual ml;  $S = \max(0, R - A)$ . Stamina starts at 300 and can only decrease to 0.

Special first 10 minutes ( $t = 0-10$ )

Requirement: 500 ml; deductions at  $t=10$ :

- 500+ ml  $\rightarrow$  0 pts
- 250–499 ml  $\rightarrow$  -15 pts
- 1–249 ml  $\rightarrow$  -30 pts
- 0 ml  $\rightarrow$  -40 pts

Remaining 6h50m (41 intervals of 10 min)

Distribute the remaining target evenly across 41 intervals. For each interval:

- $S = 0 \rightarrow$  0 pts
- $0\% < S \leq 25\%$  of  $R \rightarrow$  -2 pts
- $25\% < S \leq 50\%$  of  $R \rightarrow$  -4 pts
- $50\% < S \leq 75\%$  of  $R \rightarrow$  -6 pts
- $75\% < S \leq 100\%$  of  $R$  (or more)  $\rightarrow$  -6.5 pts (interval cap)

## Clamps & Caps

- Session cap: Total deducted stamina =  $\min(300, \text{sum of interval deductions})$ .
- No recovery: stamina never increases within the session.
- Data gaps: treat missing as  $A = 0$  (max penalty); skip & log corrupt packets.

## Distance Calculation

- Distance progression rate is a linear function of current stamina; cumulative distance capped at 42.195 km.

## Penalty tuning (explicit)

- The above values yield a worst-case near 300 when the user drinks nothing:  $\approx 40 + (41 \times 6.5) \approx 307$ , then clamped to 300. If you want an easier/harder session, adjust the interval tiers (especially the 6.5-pt cap) and re-validate that the total never exceeds 300.

## 4.2 Session Lifecycle

- Start: user taps Start; 7h countdown begins after time sync.
- Backgrounding/quit: coaster keeps logging; on reconnect, the app re-syncs time and backfills before applying penalties.
- End: auto at 7h or manual early end → show current distance; mark as incomplete.

## 4.3 Device & Connectivity

- Bluetooth (required): BLE pairing, auto-reconnect, dedup duplicates, foreground & background reads.
- Internet (only for time sync):
- Used solely to verify current time against a trusted source at session start, every ~10 minutes, and on reconnect.
- If unavailable, use device clock; show “Time not verified” and continue. When internet returns, perform lightweight resync (tolerate drift up to  $\pm 2s$ ; if greater, recompute last two minutes to prevent double penalties).
- No other cloud dependencies in v1: no auth/profile/cloud backup online.

## 4.4 UI Requirements

- Onboarding: avatar picker; Bluetooth permission; coaster pairing; clear target explanation.
- Run screen:
- Stamina bar (0–300)

- Distance counter (0–42.195 km)
- Status tray: Bluetooth / Coaster; Time status (Verified / Not verified)
- Cue for the first-10-minute 500 ml window
- Finish: total km, personal best, Start new race.

## **5) Non-Functional Requirements**

- Reliability: at-least-once ingestion; idempotent merge; resilient to short disconnects.
- Performance: UI update latency <100 ms; BLE notify/poll per device spec (~1–5 s).
- Security/Privacy: hydration data stored locally; BLE encrypted in transit; clear privacy copy.
- Battery: background BLE optimized; suspend polling when screen is off and no cup present.

## **6) Analytics & Success Metrics**

- Activation: % who complete pairing and tap Start on day 1.
- Engagement: avg intake (ml), % hitting gender target, % hitting 500 ml by 10 min.
- Outcome: avg distance/session, % completing 7h run, 7-day retention.
- Quality: BLE reconnect success rate, data-gap minutes/session, clock-drift incidents/resyncs.

## **7) Edge Cases & Error States**

- No coaster detected: “Connect coaster” sheet with retry/help.
- BLE pairing denied: persist pending state; allow re-prompt.
- Out-of-order timestamps: reorder; discard outside session window; log.
- Clock drift: resync; recompute last two minutes to avoid double penalties.
- Missing/corrupt data: missing → 0 ml (max penalty); corrupt packets skipped & logged.

## **8) Dependencies & Open Questions (updated)**

Hard dependencies

- Coaster firmware:
  - Emits intake volume with timestamps and device clock.
  - Supports time-sync hooks and session markers (start/end).
  - Stable BLE characteristic(s) for volume + timestamp, with monotonic sequence or nonce to dedup.

- BLE stack & OS capabilities:
- iOS/Android background BLE modes enabled; reconnection strategy; packet size/MTU within platform limits.
- Permissions flows (Bluetooth) and power-saving exemptions where needed (Android battery optimizations).
- Time sync provider:
  - Trusted network time (e.g., OS or NTP-backed service). Internet usage is only for time verification; the run proceeds if offline with “Not verified” status.
  - Drift handling:  $\pm 2s$  tolerance; beyond that, recompute last two minutes.
- Penalty configuration (local):
  - Penalty table is bundled in-app (no remote config, since internet is not used beyond time sync).
  - Versioned locally; updates ship via app releases; unit tests must assert “max total  $\leq 300$ .”

Nice-to-have (post-v1)

- Cloud backup/sync, accounts/auth, leaderboards—out of scope for v1.
- Telemetry/analytics upload; if later added, must be strictly opt-in and not required for core operation.

Open questions

- Final copy around gender targets and ability to adjust target later.
- Whether to expose a manual “Time verify now” action when internet returns.
- Exact BLE reconnect backoff policy (e.g., exponential vs fixed).

## 9) Further Information



