M2DG v1 — Product Requirements Document (PRD)

Version: v1.0 • Scope: Courts → Check-in → Profile → Leaderboards

Defaults locked: 100m radius • 30-minute cooldown • Daily streak (1/day)

1) Product Summary

M2DG is a discipline-first basketball platform that connects hoopers to real-world courts through verified check-ins, local motivation/tease notifications, and competitive leaderboards. The MVP focuses on Court discovery, verified presence, daily streaks, and Court Champ recognition.

2) Goals

Product goals

- Help athletes show up consistently at real courts.
- Make court presence verifiable with anti-cheat controls.
- Create competition that feels local-first: Court → City → Global.
- Encourage healthy session behavior: hydration/stretching prompts during cooldown.

MVP success criteria (measurable)

- User can find a court, get proximity prompts, check in with QR + GPS verification, view profile stats, and view leaderboards.
- Verification rejects obvious spoof attempts and enforces 30-minute cooldown server-side and client-side.
- App remains responsive on mid-range devices; core screens load quickly on normal mobile networks.

3) Target Users

- **Athletes/Hoopers** — want competition, recognition, consistency loops.
- **Casual players** — want low-friction check-in + fun prompts.
- **Youth/Parents** (later) — not MVP, but data model should remain extensible.

4) MVP Scope (Build Order)

Phase 1 (MVP)

- Courts directory (map + list + court detail + set home court).
- Check-in (QR scan + GPS verification within 100m + cooldown enforcement).
- Profile (username/display, home court, verified check-ins, daily streak).
- Leaderboards (Court/City/Global snapshots + Court Champ).
- Called Next (court queue): show who called next, with respectful turn-taking.

Non-goals for MVP

- Live streaming, full social feed/DMs, payments/subscriptions, referee disputes, admin dashboard.

5) Core UX Requirements

5.1 Court proximity notifications (100m)

Trigger: When a user enters a 100m geo-zone around a registered court, the device fires a local notification. Tone can be Motivate/Tease/Mixed. The server is not required to receive continuous location updates.

Example notifications

- “Oh, you tryna put some work in? ■”
- “I know you not too good to play ball in yah fit ■”
- “You this close… might as well clock in ■”

5.2 Leaving the court area → clock-out warning + cooldown timer UI

When the user exits beyond 100m from the checked-in court, the app shows a friendly warning and offers a voluntary “Clock Out” UX. Regardless of whether they tap it, cooldown rules still apply to prevent rapid re-check-ins.

- “You clocking out for the day?”
- “You done putting work in? Good work — way to put that work in ■”

Cooldown timer: After a verified check-in, show a visible countdown (30:00 → 00:00). If a user attempts to check in during cooldown, block the action and show: “Cooldown active — get some water, stretch.”

6) Functional Requirements

6.1 Authentication

- Email/password or magic link (MVP selection).
- Unique username required.
- Profile record created on first login.

6.2 Courts Directory

- Map + list view; search by city/name.
- Court detail: name, city/state, distance, activity prompt, “Check in” CTA.
- Set home court (MVP: allow change; later restrict to 2 changes).

6.3 Check-in (QR + GPS) with anti-cheat

- User must scan a court QR code to initiate check-in.
- Server verifies user is within 100m of court location using distance calculation (PostGIS recommended).
- Cooldown: 30 minutes enforced server-side; UI countdown enforced client-side.
- Attempt during cooldown returns user-friendly message + remaining time.
- Server stores rejection reason codes for debugging/telemetry.

6.4 Daily streak (once per day)

Definition: A day counts if at least one verified check-in occurs that calendar day (user timezone). Multiple check-ins in the same day do not increase the streak count beyond 1 for that day.

6.5 Called Next (court queue)

- Only users with an active verified session at a court can “Call Next”.
- Queue is per court; shows top 1–5 visible positions.
- A user can only occupy one queue slot per court at a time.
- Queue entries expire automatically (default: 60 minutes) or when user leaves the court area (optional v1 behavior).
- Court screen shows: “Next up: ” and “Queue: #1, #2, #3…”
- Anti-cheat: server validates eligibility, rate limits, and prevents spam.

7) Leaderboards

- Court leaderboard: top users at that court.
- City leaderboard: top users within same city.
- Global leaderboard: top users overall.
- Court Champ label: #1 at the court (based on leaderboard ranking).
- Use snapshot tables for fast reads; refreshed by scheduled job.

8) Non-Functional Requirements

- Performance: leaderboards should feel instant using snapshots; avoid expensive joins on every request.
- Security: strict RLS; client cannot mark a check-in as verified.
- Privacy: proximity prompts computed on-device; avoid constant server tracking.
- Observability: logs for check-in verification + rejection reasons.

9) Version Control & Debugging SOP

Git rules

- main is always stable; feature branches: feat/; fixes: fix/.
- Small commits at stable checkpoints with clear messages: feat:, fix:, chore:, docs:.
- Tag stable milestones: v0.1.0, v0.2.0, etc. Maintain CHANGELOG.md.
- Before merging to main: build passes, happy path verified, no noisy logs.

Debugging SOP (always follow)

- Freeze: capture exact error + what you clicked + expected vs actual.
- Reproduce: reduce to 1–3 steps.

- Isolate: UI vs network vs Edge Function vs RLS vs DB.
- Observe: logs in Flutter + Edge Functions + DB queries.
- Fix minimally: smallest change that resolves.
- Lock it: commit the fix; add guard/test if possible.

10) Next Documents

- Technical Spec: API contracts, RLS policies, Edge Functions, and data flow.
- Screen map + UI flows for Courts, Check-in, Profile, Leaderboards, Called Next.
- Anti-cheat rules doc v1: heuristics and rate limits.