

Mục tiêu & Phạm vi

Bản v2 (final) – đã cập nhật theo yêu cầu mới: (1) `orderCode` an toàn ký tự; `description` ngắn, chỉ chữ/số; (2) Member chỉ cần `name`, `nickname`; (3) PayOS QR **có amount**; (4) Link tổng: chọn **tên** ⇒ tự suy ra **team**; (5) Danh sách **keys/config** cần chuẩn bị; (6) Toàn bộ triển khai hướng **free tier**.

- **Mục tiêu:** MVP web quản lý + link tổng cho member, dùng **Firestore + Firebase Functions + PayOS** (webhook), không cần mobile app.
- **Người dùng:** ~20 người (bao gồm Admin).
- **Yêu cầu nổi bật:**
 - Pre-store **Members** để chọn nhanh và kéo-thả vào roster team.
 - Hỗ trợ **2 hoặc 3 team** tùy trận.
 - Admin **nhập tổng tiền** và **% cho từng team** (tổng = 100). Hệ thống **chia đều** trong team.
 - **Một link tổng** cho member vào, **chọn tên**, sau đó **embed PayOS checkout** trong modal/iframe.
 - **Webhook PayOS** cập nhật realtime **status từng member** trên trang quản lý.
 - **Responsive** tốt (mobile-first, desktop có bảng/board rộng).

Kiến trúc tổng quan

flowchart LR

```
A[Web Admin/Member (Next.js)] -- Firebase Auth --> FS[(Firestore)]
A --> CF1[Cloud Function (Callable): createPayment]
CF1 --> PayOS[PayOS API]
PayOS --> CF2[Cloud Function (HTTP): webhook]
CF2 --> FS
```

- **Firestore** giữ toàn bộ dữ liệu (members, teams, matches, rosters, shares, payment_events).
- **Functions:**
 - `createPayment` (Callable): Tạo payment link/QR PayOS (ẩn key), trả về `checkoutUrl` để embed.
 - `payosWebhook` (HTTP): Nhận sự kiện PAID/CANCELLED... từ PayOS, cập nhật `shares` idempotent.
- **FE:**
 - Admin pages: Members, Setup Match (kéo-thả), Dashboard theo dõi.
 - Member page: Link tổng `/pay/:matchId`.

Cấu trúc dữ liệu (Firestore)

Collections

```
members/{memberId} {
  name: string,
  nickname?: string,
  createdAt: timestamp
}

teams/{teamId} {
  name: string,
  color?: string,
  createdAt: timestamp
}

matches/{matchId} {
  date: ISOString,
  totalAmount: number,
  teamCount: 2|3,
  teamPercents: { [teamId: string]: number },
  createdByUid: string,
  createdAt: timestamp
}

matches/{matchId}/rosters/{teamId} {
  memberIds: string[]
}

matches/{matchId}/shares/{shareId} {
  memberId: string,
  teamId: string,
  allocatedPercent: number,
  amount: number,
  status: 'PENDING'|'PAID'|'CANCELLED',
  orderCode: string,           // chỉ A-Z, 0-9, _, - (vd: MATCH_abc_MEM_def)
  description?: string,       // ngắn, chỉ chữ/số (vd: M123T2U15)
  checkoutUrl?: string,       // từ PayOS - QR động bao gồm amount
  qrUrl?: string,
  paidAt?: ISOString,
  channel?: 'PAYOS',
  meta?: { payer?: any, webhook?: any },
  createdAt: timestamp
}

payment_events/{eventId} {
```

```

type: 'PAYOS_WEBHOOK',
orderCode: string,
amount: number,
status: string,
postedAt: timestamp,
raw: object
}

```

Quy ước & Khoá chính

- `orderCode = MATCH_<matchId>__MEM_<memberId>` (an toàn ký tự; idempotent).
- `description` (tùy chọn, chỉ chữ/số ngắn) ví dụ: `M123T2U15`.
- QR từ PayOS là **QR động**, đã bao gồm `amount` \Rightarrow giảm sai lệch.

Security Rules (phác thảo)

- Chỉ **UID Admin** có quyền ghi.
- Link member (public) chỉ **đọc** match + shares trong trận cụ thể; gọi `createPayment` qua Callable (Functions kiểm UID Admin hoặc mở chế độ *limited guest* chỉ cho tạo payment cho chính mình nếu cần).

Logic tính tiền

- Input: `totalAmount`, `teamPercents` (tổng = 100), `rosters` (danh sách member theo team).
- Công thức mỗi team `Ti` có `pi = teamPercents[Ti]`, `si = số member team Ti`:
- `teamTotal = totalAmount * pi / 100`
- `amount_per_member = floor(teamTotal / si)`
- Phần lẻ còn lại (do làm tròn) phân bổ +1đ cho các member đầu/hoặc cuối danh sách cho đến hết.
- Sau khi phân bổ cho tất cả team, nếu tổng cộng \neq `totalAmount`, điều chỉnh \pm diff vào member cuối cùng của team đầu (đảm bảo tổng khớp).

Acceptance: - [] Tổng `sum(shares.amount)` = `totalAmount`. - [] Mỗi member ở team `Ti` có `allocatedPercent = pi`. - [] Nếu team có `pi>0` thì `si>0`; nếu `si=0` thì chặn lưu.

Flow UI chi tiết

1) Trang Members (pre-store)

Mục tiêu: Lưu danh sách thành viên sẵn để chọn nhanh trong mỗi trận.

UI & Tương tác: - Bảng thành viên: Tên, Điện thoại, Ghi chú, Hành động (Sửa/Xóa). - Ô **quick add** (enter để thêm nhanh nhiều dòng). - Tìm kiếm theo tên/phone.

Logic: - CRUD trực tiếp Firestore `members`. - Không ràng buộc team ở đây; team sẽ gán trong màn Setup Match.

Acceptance: - [] Thêm được 10–20 người nhanh (dán danh sách tên, enter liên tục).

2) Trang Setup Match

Mục tiêu: Tạo trận, chọn 2 hoặc 3 team, nhập % từng team, kéo-thả member vào roster team, tính và lưu shares.

UI: - Form bên trên: - Date (default hôm nay), `teamCount` (2|3) - Chọn/đặt tên Team A/B/C) - Input % cho từng team (hiển thị tổng % = 100, cảnh báo nếu $\neq 100$) - `totalAmount` (VND) - **Board 3 cột** (A/B/C) bằng `dnd-kit`: - Trái: **Pool** (danh sách tất cả `members` chưa gán) - Giữa/Phải: **Team columns** (drop zone) - hiển thị avatar initials + tên. - Hỗ trợ kéo-thả, double-click để chuyển cột nhanh. - Nút "**Tính tiền & Lưu**": chạy `computeTeamShares` → ghi `matches`, `rosters`, `shares` (status `PENDING`). - Nút "**Copy link tổng**": `/pay/<matchId>`

Validation: - [] Tổng % = 100 mới cho lưu. - [] Mọi team có `pi>0` phải có ≥ 1 member. - [] `totalAmount` `> 0`.

Acceptance: - [] Kéo-thả mượt; có thể gán 10–15 người trong 1–2 phút. - [] Sau khi lưu, tạo N shares tương ứng với tổng số member trong các team.

3) Trang Member Link `/pay/:matchId`

Mục tiêu: Một link tổng; member **chọn tên** ⇒ hệ thống tự suy ra team, amount; thanh toán trong iframe PayOS.

UI (step-by-step): 1. **Chọn Tên** (autocomplete từ shares của match). Không cần chọn team. 2. Hiển thị **amount** và (tuỳ chọn) `description` ngắn. 3. **Thanh toán:** - Gọi Callable `createPayment({matchId, memberId})` - Nhận `checkoutUrl`, `qrUrl` (QR động có **amount**) - Mở **modal** với **iframe** trở `checkoutUrl` - Nhắc: trạng thái hoàn tất do **webhook** cập nhật (không phụ thuộc redirect)

Logic: - Lắng nghe `shares/{shareId}` realtime để chuyển `PENDING` → `PAID`. - CSP cho phép `frame-src https://*.payos.vn`.

Acceptance: - [] Chọn tên 1 chạm, thấy amount chính xác. - [] Status đổi sang `PAID` vài giây sau thanh toán.

4) Trang Dashboard Match /matches/:matchId

Mục tiêu: Admin theo dõi trạng thái từng member.

UI: - Bảng: Team | Member | Amount | Status (chip) | PaidAt | Actions - Tổng quan (cards): Tổng thu/Đã thu/Còn lại, breakdown theo team. - Actions: - **Copy link cá nhân** (deep link với `memberId` nếu muốn) - **Regenerate link** (gọi lại `createPayment`) - idempotent theo `orderCode` - **Mark Paid/Cancel** (override) - bắt buộc nhập ghi chú

Logic: - Subscribe `matches/:id/shares` realtime. - Tính tổng số tiền theo `status` (client-side).

Acceptance: - [] Bảng realtime theo dõi mượt. - [] Override thủ công hoạt động & ghi chú lưu vào `meta`.

Cloud Functions (Contracts)

1) `createPayment` (Callable)

Input: { `matchId`: string, `memberId`: string }

Steps: 1) Load share (`PENDING`) theo `matchId/memberId`; build `orderCode` = `MATCH_<matchId>__MEM_<memberId>`. 2) Gọi PayOS `POST /payment-requests` với { `orderCode`, `amount`: share.amount, `description?`: share.description, `returnUrl`, `cancelUrl` }. 3) Lưu `checkoutUrl`, `qrUrl` (QR đã có amount). Trả { `checkoutUrl`, `qrUrl` }.

Idempotent: nếu đã có `checkoutUrl`, trả lại luôn.

2) `payosWebhook` (HTTP)

- Verify chữ ký theo header PayOS.
- Parse `orderCode` ⇒ `matchId/memberId`.
- Nếu `status === 'PAID'` ⇒ update share: `PAID`, `paidAt`, `channel='PAYOS'`, lưu `meta.webhook`.
- Log toàn bộ body vào `payment_events`.
- Trả `200 { ok: true }` kể cả khi đã `PAID` (idempotent).

Thành phần UI (tối thiểu)

- `MembersTable` (CRUD `name`, `nickname`; quick add + search)
- `TeamBoardDnd` (Pool + A/B/C, kéo-thả dnd-kit)
- `MatchForm` (date, teamCount 2 | 3, % inputs, totalAmount; validate sum=100 & team>0 member)
- `ShareTable` (realtime table, chips status, actions)

- `MemberPicker` (chọn tên \Rightarrow tự suy ra team + amount)
- `PayModal` (iframe checkoutUrl; fallback mở tab mới nếu CSP chặn)

State machine `share`: `PENDING` \rightarrow (paying UI) \rightarrow `PAID` | `CANCELLED`.

- `MembersTable` (CRUD + quick add + search)
- `TeamBoardDnd` (Pool + A/B/C, kéo-thả bằng `@dnd-kit/core`)
- `MatchForm` (date, teamCount, % inputs, totalAmount, validate sum=100)
- `ShareTable` (realtime table, chips status, actions)
- `MemberPicker` (team select + member select, hiển thị amount)
- `PayModal` (iframe checkoutUrl + trạng thái realtime)

Trạng thái & State machine (share): - `PENDING` \rightarrow (user tạo payment) \rightarrow (**optional UI**) `PAYING` \rightarrow (webhook) \rightarrow `PAID` | `CANCELLED`. - UI có thể hiện `PAYING` khi vừa gọi `createPayment` (không bắt buộc ghi Firestore, chỉ UI state).

Responsive & UX

- **Mobile:** Member Picker theo step (wizard), bảng Share chuyển thành card list.
- **Desktop:** Board kéo-thả 3 cột, Dashboard 2 panel (bảng + tổng quan).
- **Accessibility:** focus ring, keyboard drag (tuỳ chọn), aria-label buttons.

Edge cases & Quy tắc vận hành

- **Sai số tiền:** Webhook amount \neq `share.amount` \Rightarrow không auto-PAID, highlight row `Amount mismatch`, cần Admin xử lý tay.
- **Double webhook:** bỏ qua, idempotent theo `orderCode`.
- **Team có % > 0 nhưng roster rỗng:** chặn lưu match.
- **Thay đổi roster sau khi có PAID:** khoá chỉnh sửa với những member đã `PAID`.
- **Regenerate link:** cùng `orderCode` để giữ idempotency; nếu PayOS yêu cầu unique, cho phép suffix theo phiên bản nhưng vẫn map về cùng share.

Cấu hình & Bảo mật

Keys/Config cần chuẩn bị

- **Firebase:** `FIREBASE_PROJECT_ID` (deploy), cấu hình Web SDK (`apiKey`, `authDomain`, `projectId`, `appId`).

- **PayOS:** `PAYOS_CLIENT_ID`, `PAYOS_API_KEY`, `PAYOS_WEBHOOK_SIGNATURE_KEY`, `PAYOS_RETURN_URL`, `PAYOS_CANCEL_URL`.
- **Functions Config** (ví dụ):

```
firebase functions:config:set
  payos.client_id="<CLIENT_ID>"
  payos.api_key="<API_KEY>"
  payos.signature_key="<WEBHOOK_SIGNATURE>"
  payos.return_url="https://yourapp.web.app/pay/success"
  payos.cancel_url="https://yourapp.web.app/pay/cancel"
```

- **Webhook URL:** cấu hình trong PayOS trỏ tới `https://<region>-<project>.cloudfunctions.net/payosWebhook` (hoặc via Hosting rewrite).
- **CSP:** `frame-src https://*.payos.vn https://payos.vn;`
- **Auth:** Admin login (Email/Password). Link member public **read-only** cho match đó.

Chi phí (free tier)

- Firestore/Hosting/Functions: dư dả cho ~20 người.
- PayOS: ~1.000 giao dịch/tháng free. Có thể tạo project mới khi test nhiều (hạn chế lạm dụng).

Kiểm thử (Checklist)

1. Tạo 2 team, nhập % tổng=100, kéo-thả 10 member.
2. Tính & lưu: shares tạo đúng, tổng tiền đúng.
3. Member vào link tổng, chọn tên, tạo payment, iframe mở được.
4. Thanh toán sandbox → webhook cập nhật `PAID` trong ≤5s.
5. Thanh toán lặp lại cùng `orderCode` → không double-paid.
6. Sai amount → row cảnh báo, không auto `PAID`.
7. Regenerate link → hoạt động idempotent.
8. Responsive: mobile xem/tt toán OK.

Non-goals (giai đoạn này)

- Không xây mobile app.
- Không đọc notification/scrape giao dịch.
- Không hỗ trợ nhiều currency.

Ghi chú triển khai nhanh

- Ưu tiên **Next.js** + **Tailwind**.
- Dùng **@dnd-kit** cho kéo-thả.
- Callable & Webhook Functions viết bằng **TypeScript**.
- Đặt **Budget Alert** Firebase 2-5 USD/tháng.