

# BÀI KIỂM TRA 02 (NÂNG CAO - MOBILE)

**Môn học:** Lập trình Mobile với Flutter

**Công nghệ:** Flutter (Frontend – Mobile App), ASP.NET Core Web API (Backend), Entity Framework Core, Identity, SQL Server, **SignalR (Real-time)**, **Background Services**, **Redis (Optional)**.

**TÊN ĐỀ TÀI: HỆ THỐNG QUẢN LÝ CLB PICKLEBALL "VỢT THỦ PHỐ NÚI" (PCM) - MOBILE EDITION**

## PHẦN 1: MÔ TẢ BÀI TOÁN & NGHIỆP VỤ (FULL SCENARIO)

CLB "Vợt Thủ Phố Núi" hoạt động dựa trên tinh thần "Vui - Khỏe - Có Thưởng". Phiên bản **Mobile** yêu cầu ứng dụng di động (Flutter) kết nối với Backend API, hỗ trợ **Ví điện tử**, hoạt động **thời gian thực (Real-time)** và giải đấu chuyên nghiệp. Người dùng sử dụng app trên điện thoại (Android/iOS) để đặt sân, tham gia giải, xem thông báo và quản lý ví.

### 1. Quản trị Nội bộ & Tài chính Thông minh (Smart Operations)

- **Quản lý Hội viên & Hạng mức (Membership Tier):**

- Mỗi thành viên có một Hồ sơ số (Digital Profile) và **Ví điện tử (Digital Wallet)**.
- Mỗi hội viên phải có tài khoản Identity để đăng nhập trong app.
- **Hạng thành viên (Tier):** Hệ thống tự động xét hạng (Đồng, Bạc, Vàng, Kim Cương) dựa trên tổng tiền đã chi tiêu hoặc điểm tích lũy. Hạng cao sẽ có quyền lợi đặc biệt (VD: Đặt lịch định kỳ).

- **Quản lý Tài chính & Ví (Wallet System):**

- **Nạp tiền (Deposit):** Thành viên tạo yêu cầu nạp tiền vào Ví (trong app) → Admin/Thủ quỹ duyệt → Cộng tiền vào số dư.
- **Thanh toán tự động:** Mọi hoạt động (Đặt sân, Tham gia giải đấu) đều **trừ tiền trực tiếp** từ Ví điện tử. Hệ thống từ chối nếu số dư không đủ.
- Quản lý dòng tiền Thu/Chi minh bạch. Cảnh báo quỹ âm (hiển thị trong app cho Admin/Treasurer).

- **Tin tức & Thông báo (Real-time):**

- Đăng tải thông báo, vinh danh. Tin có thể được ghim (IsPinned).
- **Thông báo đẩy (Push Notifications / SignalR):** Người dùng nhận thông báo tức thời khi: Nạp tiền thành công, Lịch đặt sân được xác nhận, Có kết quả trận đấu mới. App có thể dùng **SignalR** (real-time) hoặc **FCM** (Firebase Cloud Messaging) cho push khi app nền.

### 2. Hoạt động Thường nhật (Advanced Booking)

- **Đặt sân thông minh (Smart Booking):**

- Thành viên xem lịch trong app, chọn khung giờ và đặt sân.
- **Chặn trùng lịch:** Tuyệt đối không cho phép 2 booking giao nhau.

- **Đặt lịch định kỳ (Recurring Booking):** Hội viên VIP có thể đặt lịch cố định (VD: "Mỗi thứ 3 và thứ 5 hàng tuần trong tháng 10"). Hệ thống tự động kiểm tra và sinh các booking con.
  - **Giữ chỗ (Hold Slot):** Khi user chọn slot trên lịch, slot chuyển trạng thái "Đang giữ" (Booking Holding) đối với người khác trong 5 phút.
- **Trận đấu Giao hữu (Daily Matches):**
    - Hoạt động diễn ra thường xuyên. Kết quả làm biến động Rank DUPR.

### 3. Sàn đấu & Giải đấu Chuyên nghiệp (Pro Tournaments)

- **Kèo Thách đấu (Duel):** Thách đấu 1vs1 hoặc 2vs2 với phần thưởng nhỏ.
- **Giải đấu Chuyên nghiệp (Tournaments):**
  - **Cấu hình giải:** Entry Fee, Prize Pool, Số lượng đội, Thể thức.
  - **Entry Fee:** Trừ trực tiếp từ Ví khi đăng ký tham gia (trong app).
  - **Prize Pool:** Kết thúc giải, hệ thống **tự động chia thưởng** cộng tiền vào Ví người thắng.
  - **Cây thi đấu (Bracket):**
    - **Vòng bảng (Group Stage):** Chia bảng đánh vòng tròn.
    - **Vòng loại trực tiếp (Knockout):** Tứ kết → Bán kết → Chung kết.
  - **Tự động xếp lịch (Auto-Scheduler):** Admin bấm "Generate Schedule", hệ thống tự động chia bảng/cặp đấu ngẫu nhiên hoặc theo hạt giống (Seed).

---

## PHẦN 2: THIẾT KẾ CƠ SỞ DỮ LIỆU (DATABASE SCHEMA)

Sinh viên sử dụng Code First và Migration. Tên bảng nghiệp vụ bắt đầu bằng 3 số cuối MSSV.

### 1. Nhóm Quản trị (Operations) & Wallet

- **[xxx]\_Members:**
  - **Cơ bản:** Id, FullName, JoinDate, RankLevel (Double), Status (hoặc IsActive - bool).
  - **Liên kết:** UserId (FK → AspNetUsers) - Liên kết Identity.
  - **Advanced (Mới):**
    - **WalletBalance (decimal):** Số dư ví hiện tại.
    - **Tier (Enum):** Hạng thành viên (Standard, Silver, Gold, Diamond).
    - **TotalSpent (decimal):** Tổng tiền đã chi tiêu (để xét hạng).
    - **AvatarUrl (string):** Đường dẫn ảnh đại diện.
- **[xxx]\_WalletTransactions (Mới - Core Financial):**
  - Id, MemberId (FK → Members).
  - **Amount (decimal):** Số tiền (+ cho nạp/thưởng, - cho thanh toán/rút).
  - **Type (Enum):** Deposit (Nạp), Withdraw (Rút), Payment (Thanh toán phí), Refund (Hoàn tiền), Reward (Thưởng giải).
  - **Status (Enum):** Pending, Completed, Rejected, Failed.
  - **RelatedId (string/int?):** ID của Booking hoặc Tournament liên quan (để truy vết).
  - Description, CreatedDate.

- [xxx]\_News:
  - Id, Title, Content, IsPinned (bool).
  - CreatedDate, ImageUrl (Optional).
- [xxx]\_TransactionCategories: (Dùng cho thu chi nội bộ khác)
  - Id, Name, Type (Thu/Chi).

## 2. Nhóm Sân & Booking

- [xxx]\_Courts:
  - Id, Name (Sân 1, Sân 2...), IsActive (bool), Description.
  - **PricePerHour (decimal)**: Giá thuê sân mỗi giờ (để tính tiền tự động).
- [xxx]\_Bookings:
  - Id, CourtId (FK), MemberId (FK), StartTime, EndTime.
  - **TotalPrice (decimal)**: Tổng tiền thanh toán.
  - **TransactionId (FK)**: Liên kết với giao dịch trừ tiền trong ví.
  - **Advanced (Mới)**:
    - **IsRecurring (bool)**: Đánh dấu là đặt lịch lặp.
    - **RecurrenceRule (string)**: Quy tắc lặp (VD: "Weekly;Tue,Thu").
    - **ParentBookingId (int?)**: Nếu đây là booking con sinh ra từ lịch lặp.
    - **Status (Enum)**: PendingPayment (Chờ thanh toán), Confirmed (Đã đặt), Cancelled, Completed.

## 3. Nhóm Thi đấu (Sports Core - Advanced)

- [xxx]\_Tournaments (Nâng cấp từ Challenges):
  - Id, Name, StartDate, EndDate.
  - **Format (Enum)**: RoundRobin (Vòng tròn), Knockout (Loại trực tiếp), Hybrid.
  - **EntryFee (decimal)**: Phí tham gia.
  - **PrizePool (decimal)**: Tổng giải thưởng.
  - **Status (Enum)**: Open, Registering, DrawCompleted (Đã bốc thăm), Ongoing, Finished.
  - **Settings (JSON)**: Cấu hình nâng cao (Số bảng, số đội vào vòng trong...).
- [xxx]\_TournamentParticipants:
  - Id, TournamentId (FK), MemberId (FK).
  - **TeamName (string)**: Tên đội (nếu đánh đôi).
  - **PaymentStatus**: Đã trừ EntryFee chưa.
- [xxx]\_Matches (Siêu chi tiết):
  - Id, TournamentId (FK, nullable), RoundName (string - VD: "Group A", "Quarter Final").
  - Date, StartTime.
  - **Participants**:
    - Team1\_Player1Id, Team1\_Player2Id (nullable).

- Team2\_Player1Id, Team2\_Player2Id (nullable).
  - **Result:**
    - Score1 (int), Score2 (int) - Tỉ số tổng (VD: 2-1).
    - **Details (JSON/String):** Chi tiết các set (VD: "11-9, 5-11, 11-8").
    - WinningSide (Enum: Team1, Team2).
  - **IsRanked (bool):** Có tính điểm DUPR không.
  - **Status:** Scheduled, InProgress, Finished.
  - [xxx]\_Notifications (Mới):
    - Id, ReceiverId (FK), Message, Type (Info/Success/Warning), LinkUrl, IsRead, CreatedDate.
- 

## PHẦN 3: YÊU CẦU API & MOBILE APP (FLUTTER)

### 1. Backend API (ASP.NET Core Web API)

**Yêu cầu chung:** Authorization (JWT), Error Handling, SignalR Hub.

#### API Endpoints chi tiết:

- **Auth & Members:**
  - [POST /api/auth/login](#): Trả về JWT Token + User Info.
  - [GET /api/auth/me](#): Lấy thông tin current user + Wallet Balance.
  - [GET /api/members](#): Danh sách members (Search, Filter, Pagination).
  - [GET /api/members/{id}/profile](#): Lấy thông tin chi tiết + Lịch sử đấu + Rank History.
- **Wallet (Advanced):**
  - [POST /api/wallet/deposit](#): Member gửi yêu cầu nạp tiền (kèm ảnh bằng chứng CK).
  - [GET /api/wallet/transactions](#): Lịch sử biến động số dư của tôi.
  - [PUT /api/admin/wallet/approve/{transactionId}](#): Admin duyệt nạp tiền → **Cộng tiền vào ví** (dùng Transaction đảm bảo an toàn).
- **Courts & Bookings:**
  - [GET /api/courts](#): Danh sách sân.
  - [GET /api/bookings/calendar?from=...&to=...](#): Lấy lịch đặt sân để hiển thị.
  - [POST /api/bookings](#): Đặt sân đơn lẻ. **Logic:** Check số dư → Tạo Booking → Trừ tiền Ví → Create Transaction.
  - [POST /api/bookings/recurring](#): Đặt sân định kỳ (VIP Only). **Logic:** Loop tạo nhiều booking, trừ tổng tiền hoặc giữ chỗ.
  - [POST /api/bookings/cancel/{id}](#): Hủy sân → Hoàn tiền (theo chính sách: trước 24h hoàn 100%...).
- **Tournaments & Matches:**
  - [POST /api/tournaments](#): Tạo giải đấu.
  - [POST /api/tournaments/{id}/join](#): Tham gia giải → Trừ Entry Fee.
  - [POST /api/tournaments/{id}/generate-schedule](#): **Auto-Scheduler** chia bảng/cặp đấu.

- [POST /api/matches/{id}/result](#): Cập nhật kết quả. **Logic:** Tính lại Rank DUPR, Cập nhật nhánh đấu tiếp theo (nếu Knockout).

- **SignalR Real-time:**

- Hub [PcmHub](#) với các method: [ReceiveNotification](#), [UpdateCalendar](#), [UpdateMatchScore](#).

## 2. Mobile App (Flutter)

**Công nghệ:** Flutter 3.x, Dart 3.x. **Khuyến nghị:** Dio (HTTP), Provider / Riverpod / Bloc (state management), go\_router hoặc AutoRoute (navigation), flutter\_secure\_storage (lưu JWT), signalr\_netcore hoặc package tương tự (SignalR).

### Cấu trúc project:

- **API Client:** Service gọi Backend API (Dio + Interceptor thêm [Authorization: Bearer {token}](#)).
- **Auth:** Lưu JWT (flutter\_secure\_storage hoặc encrypted shared\_preferences). Khi 401 → clear token, navigate về màn Login.
- **State Management:** Lưu user hiện tại, số dư ví, danh sách notifications chưa đọc. **Khuyến nghị** dùng Provider/Riverpod/Bloc.

### Modules UI chi tiết:

- **Layout chính:**

- **AppBar / SliverAppBar:** Hiển thị Avatar, Tên, **Số dư ví (Live update khi có giao dịch)**.
- **BottomNavigationBar / Drawer:** Điều hướng: Trang chủ, Lịch đặt sân, Giải đấu, Ví, Cá nhân, (Admin: Quản lý).
- **Chuông thông báo:** Icon + Badge số noti chưa đọc. Tap mở màn Notifications.

- **Dashboards:**

- **User Dashboard (Home):** Biểu đồ rank (fl\_chart hoặc charts\_flutter), Lịch thi đấu sắp tới (ListView/Card), Số dư ví (nổi bật).
- **Admin Dashboard:** Biểu đồ doanh thu (Nạp/Chi), Số lượng booking trong tháng. Chỉ hiển thị khi role Admin.

- **Booking (Lịch đặt sân):**

- Màn hình **Calendar** (table\_calendar hoặc syncfusion\_flutter\_calendar): Hiển thị theo tuần/tháng.
- Slot đã đặt (màu đỏ), Slot của mình (màu xanh), Slot trống (trắng/xám). Tap vào slot trống → BottomSheet/Page đặt sân (chọn Court, giờ, xác nhận → gọi API).
- **Realtime:** Kết nối SignalR; khi có người khác đặt thành công, ô lịch cập nhật màu ngay (Stream/Consumer).
- Form **Đặt lịch định kỳ** (VIP): Chọn ngày lặp, quy tắc (VD: T3, T5) → [POST /api/bookings/recurring](#).

- **Tournament (Giải đấu):**

- Danh sách giải (Open, Ongoing, Finished). Tap vào giải → Chi tiết giải, Bảng xếp hạng (Standings), **Bracket (Cây thi đấu)**.

- **Tournament Visualizer:** Hiển thị Bracket trực quan (CustomPainter / package bracket đấu loại trực tiếp). Cập nhật kết quả real-time qua SignalR nếu có.
- Nút "Tham gia" → Check ví → [POST /api/tournaments/{id}/join](#) → Trừ Entry Fee.

- **Wallet (Ví):**

- Hiển thị **số dư ví** lớn, nút **Nạp tiền**.
- Form nạp tiền: Số tiền, upload ảnh chuyển khoản (image\_picker) → [POST /api/wallet/deposit](#).
- **Lịch sử giao dịch:** ListView (pull-to-refresh) gọi [GET /api/wallet/transactions](#). Filter: Nạp, Trừ tiền, Hoàn tiền.

- **Profile & Members:**

- Màn **Cá nhân:** Xem/sửa thông tin, Avatar, Rank, Tier. Đổi mật khẩu (nếu API hỗ trợ).
- Danh sách Members (tìm kiếm, lọc), xem profile member khác.

- **Notifications:**

- Màn hình danh sách thông báo. Đánh dấu đã đọc. Có thể nhận real-time qua SignalR và cập nhật badge.

### **Yêu cầu kỹ thuật Flutter:**

- **Responsive:** Layout hoạt động tốt trên nhiều kích thước màn hình (phone, tablet).
- **Offline cơ bản (khuyến nghị):** Cache một số dữ liệu (VD: danh sách Courts, user profile) bằng cho thư viện cache (flutter\_cache\_manager, hive, v.v.) để xem khi mất mạng.
- **Error handling:** Hiển thị thông báo lỗi (SnackBar, dialog) khi API fail; retry khi cần.
- **Loading:** Hiển thị indicator (CircularProgressIndicator, skeleton) khi đang gọi API.

## **PHẦN 4: YÊU CẦU KỸ THUẬT SYSTEM (NÂNG CAO)**

### 1. Background Services (HostedService)

- **Auto-Cancel Unpaid Bookings:** Service chạy ngầm (1 phút/lần) tìm các booking "Hold" quá 5 phút chưa thanh toán → Hủy & Release slot.
- **Auto-Remind:** Gửi notification nhắc lịch đấu trước 1 ngày.

### 2. SignalR Implementation

- **Group Management:** Mỗi trận đấu là 1 Group SignalR. Khi cập nhật tỉ số, chỉ broadcast cho user đang xem trận đó.
- **Global Notify:** Thông báo nạp tiền thành công broadcast tới User Id cụ thể.
- **Flutter client:** App kết nối SignalR Hub, lắng nghe [ReceiveNotification](#), [UpdateCalendar](#), [UpdateMatchScore](#) để cập nhật UI real-time.

### 3. Data Seeding (Quan trọng)

- **1 Admin, 1 Treasurer, 1 Referee.**
- **Members:** 20 thành viên, có Rank DUPR và Tier khác nhau.

- **Wallet:** Các thành viên mẫu có sẵn 2.000.000đ – 10.000.000đ trong ví.
  - **Data mẫu:** 1 Giải đấu "Summer Open 2026" đã kết thúc, 1 giải "Winter Cup" đang mở đăng ký.
- 

## PHẦN 5: ĐIỂM CỘNG (BONUS)

1. **Payment Gateway Integration:** Tích hợp Sandbox VNPay hoặc VietQR để nạp tiền tự động (quét mã → Webhook cập nhật ví).
  2. **Export Reports:** Xuất báo cáo doanh thu, danh sách VĐV ra Excel/PDF (Backend API + mở file/share trong app).
  3. **Optimistic Concurrency:** Xử lý tranh chấp khi 2 người cùng bấm "Đặt sân" tại 1 slot.
  4. **Chat System:** Phòng chat cho mỗi giải đấu (SignalR + UI trong app).
  5. **Push Notifications (FCM):** Firebase Cloud Messaging cho thông báo khi app đóng hoặc nền.
  6. **Biometric Login:** Đăng nhập bằng vân tay/Face ID (local\_authentication).
- 

## PHẦN 6: QUY ĐỊNH NỘP BÀI

1. **Source Code:** GitHub. Cấu trúc gợi ý:
  - **Backend:** Folder hoặc repo riêng (ASP.NET Core Web API).
  - **Mobile:** Folder hoặc repo riêng (Flutter project).
2. **Video Demo:** Quay lại luồng: Mở app → Đăng nhập → Nạp tiền → Ví tăng → Đặt sân → Ví giảm → Admin kiểm tra doanh thu (trên app hoặc API).
3. **Cú pháp nộp:** MOBILE\_FLUTTER\_[MSSV]\_[TEN]  
Ví dụ: MOBILE\_FLUTTER\_123456\_NguyenVanA

### Lưu ý:

- README mô tả cách chạy **Backend** (dotnet run, migration) và **Flutter** (flutter pub get, flutter run).
- Cung cấp **Base URL** API (localhost / deploy) để chấm bài. Nếu dùng emulator, cần **10.0.2.2** (Android) hoặc **localhost** (iOS) cho API chạy trên máy.
- **Khuyến nghị:** Đính kèm file **APK** (Android) hoặc hướng dẫn build để giảng viên cài đặt kiểm tra.

Chúc các em làm bài tốt!