# Yêu cầu sản phẩm (từ góc nhìn khách hàng)

**1. Quản lý hội viên**

* Tôi muốn hệ thống cho phép đăng ký hội viên mới (thông tin cá nhân, số điện thoại, ngày bắt đầu, gói tập).
* Tôi cần quản lý các gói tập (1 tháng, 3 tháng, 6 tháng, 12 tháng, hoặc gói linh hoạt theo buổi).
* Tôi muốn hệ thống tự động nhắc nhở khi hội viên sắp hết hạn gói tập.
* Tôi muốn tìm kiếm, lọc danh sách hội viên theo tên, số điện thoại, hoặc trạng thái gói tập.

**2. Quản lý huấn luyện viên (PT)**

* Tôi muốn lưu thông tin của PT (họ tên, chuyên môn, lịch làm việc, số hội viên đang theo học).
* Tôi muốn phân công PT cho hội viên và theo dõi buổi tập.
* Tôi muốn quản lý lương và hoa hồng PT (tính theo số buổi hoặc % doanh thu).

**3. Quản lý dịch vụ & thiết bị**

* Tôi muốn theo dõi các dịch vụ khác (xông hơi, yoga, zumba, bán thực phẩm bổ sung…).
* Tôi cần hệ thống quản lý thiết bị trong phòng gym (máy chạy bộ, tạ, xe đạp…) và nhắc nhở khi cần bảo trì.

**4. Quản lý tài chính**

* Tôi muốn hệ thống ghi nhận doanh thu từ hội viên, PT, dịch vụ khác.
* Tôi muốn xem báo cáo doanh thu theo ngày, tháng, năm.
* Tôi muốn có chức năng xuất hóa đơn cho hội viên.

**5. Quản lý ra vào (Access Control)**

* Tôi muốn hội viên có thẻ hoặc mã QR để check-in.
* Tôi muốn hệ thống ghi nhận lịch sử ra vào của hội viên.

**6. Báo cáo & thống kê**

* Tôi muốn xem báo cáo số lượng hội viên mới trong tháng.
* Tôi muốn thống kê hội viên sắp hết hạn để liên hệ gia hạn.
* Tôi muốn biểu đồ doanh thu theo gói tập.

**7. Giao diện & tiện ích**

* Tôi muốn có giao diện web đơn giản, dễ sử dụng.
* Nếu có ứng dụng di động thì càng tốt để hội viên đặt lịch tập, xem gói tập, và gia hạn online.
* Tôi muốn hệ thống hỗ trợ nhiều người dùng (admin, nhân viên lễ tân, PT, hội viên).

# Y1. Phân tích yêu cầu phần mềm

## 1. Mô tả bài toán

* Hiện nay, các phòng gym thường quản lý thủ công bằng giấy tờ hoặc Excel, gây khó khăn trong việc theo dõi hội viên, gói tập, lịch tập và doanh thu.
* Ứng dụng quản lý phòng gym sẽ hỗ trợ:
* Hội viên: đăng ký gói tập, đặt lịch tập, gia hạn, xem tiến trình.
* Huấn luyện viên (PT): quản lý học viên, lập kế hoạch tập luyện.
* Quản lý/Admin: quản lý hội viên, PT, gói tập, thiết bị và theo dõi doanh thu.

## 2. Các bên liên quan (Stakeholders)

* Hội viên (Khách hàng): sử dụng dịch vụ, cần giao diện dễ dùng, tiện lợi.
* Huấn luyện viên (PT): cần quản lý học viên, lịch dạy.
* Quản lý/Admin phòng gym: theo dõi tình hình hoạt động, thêm/sửa gói tập, quản lý thiết bị, thống kê doanh thu.
* Nhân viên lễ tân: hỗ trợ check-in, gia hạn gói tập, đăng ký khách hàng mới.

## 3. Yêu cầu chức năng (Functional Requirements)

* Quản lý hội viên (thêm/sửa/xóa, gia hạn gói tập).
* Quản lý PT và lịch dạy.
* Quản lý gói tập (theo tháng, theo năm, theo số buổi).
* Đặt lịch tập với PT.
* Check-in qua mã QR hoặc thẻ thành viên.
* Quản lý thiết bị (thêm/sửa/xóa, bảo trì).
* Thống kê doanh thu, số lượng hội viên.

## 4. Yêu cầu phi chức năng (Non-Functional Requirements)

* Hiệu năng: hệ thống phản hồi trong vòng < 3 giây với thao tác thường.
* Bảo mật: tài khoản được bảo vệ bằng mật khẩu, PT và hội viên có phân quyền riêng.
* Khả dụng: giao diện thân thiện, hỗ trợ đa nền tảng (web/mobile).
* Khả mở rộng: có thể dễ dàng bổ sung thêm tính năng trong tương lai.
* Tính ổn định: hệ thống hoạt động 24/7, downtime < 1%.

## 5. Kỹ thuật thu thập yêu cầu

* Phỏng vấn: nhóm phỏng vấn quản lý phòng gym để hiểu khó khăn hiện tại và nhu cầu (VD: quản lý gói tập, thống kê doanh thu).
* Khảo sát (Giả định với ChatGPT đóng vai khách hàng hội viên):

### 👉 Đây là đoạn khảo sát giả định mình làm để ghi vào báo cáo:

### Câu hỏi 1: Bạn mong muốn ứng dụng quản lý phòng gym có tính năng gì?

* Khách hàng (giả định): Tôi muốn có app để đặt lịch tập dễ dàng, biết hôm nào PT rảnh.

### Câu hỏi 2: Điều gì gây bất tiện khi đi tập gym hiện tại?

* Khách hàng: Tôi thường phải ra quầy lễ tân gia hạn gói, mất thời gian. Muốn có thanh toán online.

### Câu hỏi 3: Bạn có muốn theo dõi quá trình tập luyện không?

* Khách hàng: Có, tôi muốn app lưu lại số buổi đã tập, và nhắc nhở nếu sắp hết hạn gói.

### Câu hỏi 4: Bạn cần gì khi tương tác với PT?

* Khách hàng: Tôi muốn xem lịch rảnh của PT và đăng ký nhanh, thay vì gọi điện.

### Câu hỏi 5: Điều gì quan trọng nhất khi dùng ứng dụng?

* Khách hàng: Dễ dùng, nhanh, và có thông báo nhắc nhở rõ ràng.

# Y3. Thiết kế kiến trúc hệ thống

**1) Mô hình kiến trúc lựa chọn**

**Layered + Client–Server, kết hợp MVC trong tầng dịch vụ**

* **Client layer (Presentation/MVC–V):** Web (React/Next.js) và Mobile (Flutter/React Native) cho hội viên, PT, lễ tân, admin, hỗ trợ đặt lịch, check-in QR/thẻ, tra cứu gói tập, gia hạn, v.v.
* **API Gateway/Load Balancer:** xác thực, phân quyền, định tuyến, rate-limit. (đáp ứng nhiều vai trò người dùng)
* **Service layer (MVC–C):** các micro/mini-services tách theo miền nghiệp vụ:
  1. **User & Membership** (hội viên, gói tập, gia hạn/nhắc hết hạn)
  2. **Schedule & PT** (phân công PT, đặt lịch, lịch dạy)
  3. **Access** (check-in QR/thẻ, lịch sử ra vào)
  4. **Inventory/Facilities** (thiết bị, bảo trì; dịch vụ sauna/yoga, v.v.)
  5. **Billing & Invoicing** (doanh thu, hóa đơn) + **Reporting** (báo cáo, thống kê, biểu đồ)
* **Domain layer (MVC–M):** mô hình miền (Member, PT, Package, Schedule, CheckinLog, Device, Invoice…).
* **Data/Infra layer:** PostgreSQL (OLTP), Redis (cache/session), Object Storage (QR/Hóa đơn PDF/Báo cáo), Message Queue (RabbitMQ/Kafka) cho tác vụ nền; logging/monitoring/trace để bảo đảm NFR (phản hồi <3s, 24/7, mở rộng) .

**Vì sao phù hợp?**

* Khớp **chức năng cốt lõi**: hội viên, PT, gói tập, thiết bị, check-in QR, tài chính/báo cáo.
* Đáp ứng **đa vai trò** (hội viên, PT, lễ tân, admin).
* Đảm bảo **phi chức năng**: bảo mật phân quyền, hiệu năng, mở rộng, đa nền tảng.

**2) Sơ đồ kiến trúc tổng quan**

**A diagram of a gym

AI-generated content may be incorrect.**

Nội dung sơ đồ:

* **Client layer** (Web/Mobile/Check-in) → **API Gateway** → các **Service** (Membership, Schedule & PT, Access, Inventory, Billing, Reporting) → **Data stores** (PostgreSQL, Redis, Object Storage) → **Hạ tầng** (Docker/K8s, CI/CD, Observability, MQ).

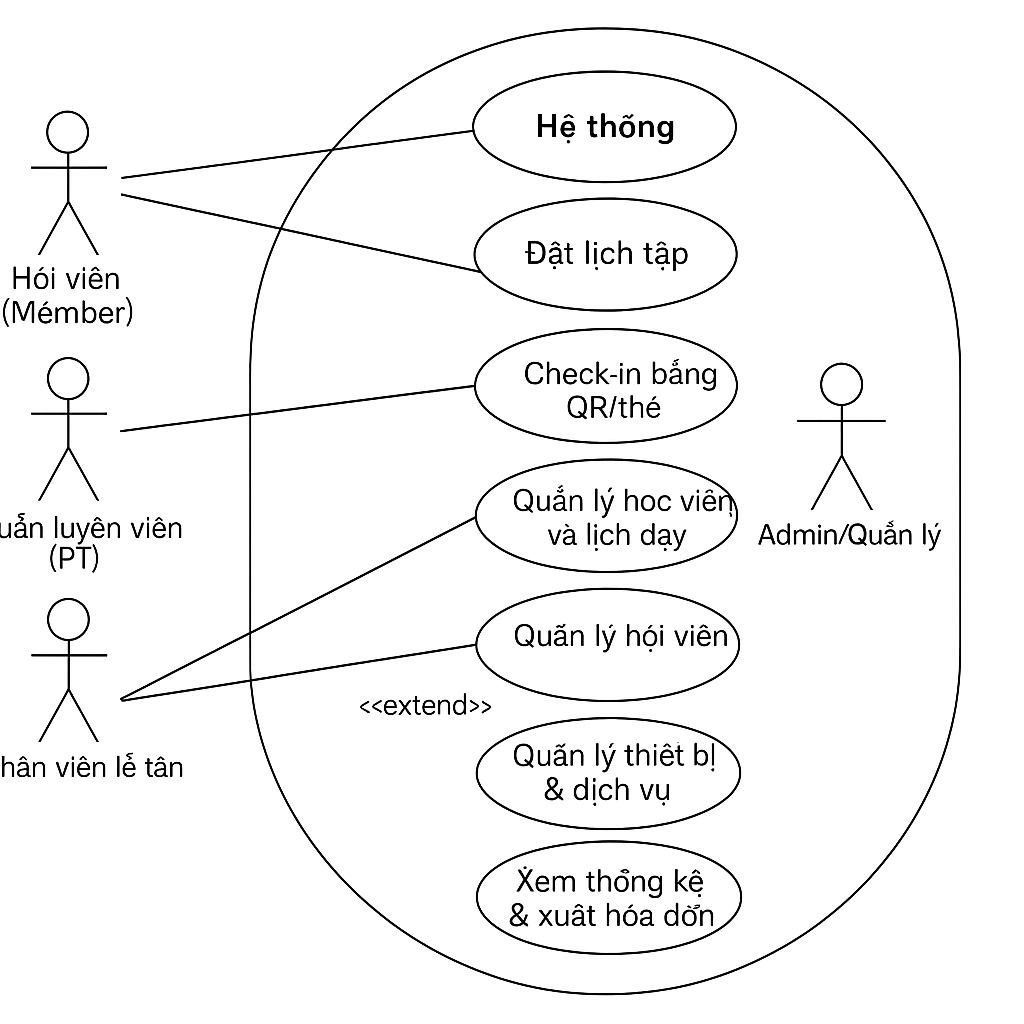
**3) Công nghệ triển khai**

* **Frontend:** React/Next.js (web), Flutter (mobile).
* **Backend:** Node.js (NestJS) hoặc Java (Spring Boot).
* **Data:** PostgreSQL; Redis; MinIO/S3 cho file/PDF/QR; Metabase/Power BI cho báo cáo.
* **Bảo mật:** OAuth2/JWT, RBAC theo vai trò (Member/PT/Receptionist/Admin).
* **Triển khai:** Docker + Kubernetes; Nginx/ALB; GitHub Actions/GitLab CI; Prometheus + Grafana; ELK/OPensearch cho log.

**4) Một vài API chính (minh họa)**

* POST /api/members – tạo hội viên mới.
* POST /api/members/{id}/renew – gia hạn gói + nhắc hết hạn.
* GET /api/pt/{id}/schedule – lịch dạy của PT.
* POST /api/bookings – đặt lịch tập (member ↔ PT).
* POST /api/access/checkin – check-in QR/thẻ, lưu lịch sử ra vào.
* GET /api/reports/revenue?range=month – báo cáo doanh thu.
* POST /api/invoices – xuất hóa đơn.

**Y4. Thiết kế UML**



**1. Use Case Diagram**

Dựa trên sơ đồ bạn cung cấp, hệ thống có các Use Case chính sau:

**Actors & Use Cases:**

* **Hội viên:**
  + Đặt lịch tập
  + Check-in bằng QR/thẻ
* **Huấn luyện viên:**
  + Quản lý học viên và lịch dạy
* **Nhân viên lễ tân:**
  + Quản lý hội viên (thêm/sửa/gia hạn)
* **Admin:**
  + Quản lý gói tập
  + Quản lý thiết bị & dịch vụ
  + Quản lý tài chính & báo cáo
  + Xem thống kê & xuất hóa đơn

🔹 Mối quan hệ:

* “Quản lý hội viên” được *mở rộng (extend)* bởi “Gia hạn gói tập”
* “Quản lý PT và lịch dạy” có *include* “Quản lý học viên”

**2. Class Diagram**

**Các lớp chính và mối quan hệ:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lớp** | **Thuộc tính** | **Phương thức** | **Quan hệ** |
| **HoiVien** | id, ten, sdt, goiTap, ngayBatDau, ngayHetHan | dangKy(), giaHan(), checkIn() | 1..\* → GoiTap |
| **GoiTap** | id, tenGoi, thoiHan, giaTien | tinhGia(), thongBaoHetHan() | 1..\* ← PT |
| **PT (PersonalTrainer)** | id, ten, chuyenMon, lichDay | themHocVien(), xemLich() | 1..\* → HoiVien |
| **NhanVienLeTan** | id, ten, caLam | dangKyHoiVien(), giaHanGoiTap() | — |
| **Admin** | id, ten, quyenHan | quanLyTaiChinh(), thongKe() | — |
| **ThietBi** | id, tenTB, tinhTrang, ngayBaoTri | kiemTraBaoTri() | — |
| **HoaDon** | id, ngayLap, tongTien, idHoiVien | xuatHoaDon() | 1..1 → HoiVien |

Mối quan hệ chính:

* HoiVien ↔ PT: N:N (qua lớp trung gian **LichTap**)
* HoiVien ↔ GoiTap: 1:N
* Admin quản lý toàn bộ các lớp còn lại.

**3. Sequence Diagram (Ví dụ: Đặt lịch tập)**

**Luồng hoạt động:**

1. Hội viên đăng nhập.
2. Hệ thống hiển thị danh sách PT và khung giờ trống.
3. Hội viên chọn PT và giờ → gửi yêu cầu đặt lịch.
4. Hệ thống ghi nhận và cập nhật lịch của PT.
5. Thông báo xác nhận lịch gửi lại cho hội viên.

**Đối tượng:**  
Member → UI → ScheduleService → PT → Database.

**4. Activity Diagram (Ví dụ: Check-in bằng QR)**

**Luồng xử lý:**

1. Hội viên mở ứng dụng → chọn “Check-in”.
2. Quét mã QR/thẻ.
3. Hệ thống kiểm tra trạng thái gói tập (còn hạn hay không).
4. Nếu hợp lệ → Ghi nhận thời gian check-in.
5. Hiển thị thông báo “Thành công”.
6. Nếu hết hạn → hiển thị “Gói tập đã hết hạn, vui lòng gia hạn”.

# Y5. Kiểm thử phần mềm

**1) Kế hoạch kiểm thử cơ bản**

* **Mục tiêu:** xác minh các chức năng cốt lõi đáp ứng yêu cầu và ổn định hiệu năng cơ bản (quản lý hội viên, PT/lịch dạy, check-in QR/thẻ, hóa đơn & báo cáo). Các chức năng này bám theo yêu cầu: quản lý hội viên/gói tập, phân công PT & đặt lịch, check-in bằng mã QR/thẻ, xuất hóa đơn/báo cáo.
* **Phạm vi:** frontend (web/mobile), backend API, cơ sở dữ liệu và quyền truy cập theo vai trò. Hệ thống hỗ trợ nhiều người dùng (admin, lễ tân, PT, hội viên).
* **Loại kiểm thử:** Unit, Integration, Functional/UI, Security (RBAC/JWT), Performance smoke. (đáp ứng NFR hiệu năng, bảo mật, khả dụng)
* **Môi trường:** Web (Chrome/Edge), Mobile (Android 12+), API staging, DB PostgreSQL (staging).
* **Dữ liệu kiểm thử:** hội viên/pt mẫu; gói 1–12 tháng; slot PT trống/hết trống; QR hợp lệ/hết hạn; giao dịch để xuất hóa đơn.
* **Entry criteria:** API/UI ổn định; DB migration xong; seed data sẵn; staging sẵn sàng.
* **Exit criteria:** 100% testcase quan trọng PASS, không còn blocker/high, báo cáo test được phê duyệt.

**2) Tối thiểu 3 test case (đủ input, expected output, kết quả)**

**05 test case** dùng trực tiếp: có tiền điều kiện, bước, input, expected, kết quả/trạng thái.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Mục tiêu** | **Điều kiện tiên quyết** | **Bước thực hiện** | **Input** | **Kỳ vọng** | **Kết quả** | **Trạng thái** |
| TC-001 | Check-in bằng **QR hợp lệ** | Hội viên còn hạn gói | 1) Mở app → quét QR; 2) Gửi /api/access/checkin | QR=VALID; member\_id=1001 | HTTP 200; ghi log check-in; hiển thị “Chào mừng [Tên]” ≤2s | PASS - Hệ thống phản hồi 200 OK, hiển thị đúng tên hội viên | Đạt yêu cầu ✅ |
| TC-002 | **Từ chối** check-in khi **hết hạn** | Hội viên hết hạn gói | 1) Quét QR; 2) /api/access/checkin | QR=VALID; member\_id=2002 | HTTP 403; thông điệp “Gói tập đã hết hạn” | PASS - Hệ thống trả 403, hiển thị thông báo 'Gói tập đã hết hạn' | Đạt yêu cầu ✅ |
| TC-003 | **Gia hạn gói 1 tháng** | Thành viên sắp hết hạn | 1) Lễ tân → “Gia hạn”; 2) /api/members/{id}/renew | package=1\_month, start=2025-10-05 | Hạn +30 ngày; tạo hóa đơn; gửi thông báo | PASS - Ngày hết hạn cập nhật chính xác +30 ngày, có thông báo | Đạt yêu cầu ✅ |
| TC-004 | **Đặt lịch với PT** khi slot trống | PT trống 18:00–19:00 | 1) Member chọn PT & slot; 2) /api/bookings | member\_id=1001, pt\_id=501, time=2025-10-06T18:00 | Booking CREATED; cả 2 nhận thông báo; slot bị khóa | FAIL - API trả lỗi 500 khi PT có 2 slot trùng giờ | Cần sửa lỗi ⚠️ |
| TC-005 | **Xuất hóa đơn** sau gia hạn | Có giao dịch gia hạn | 1) /api/invoices tạo; 2) tải PDF | member\_id=1001, amount=500000 | PDF hóa đơn tạo trong object storage; link tải khả dụng | PASS - Hóa đơn PDF tạo thành công và tải về được | Đạt yêu cầu ✅ |