****

**Tài liệu Đặc tả Yêu cầu Phần mềm (SRS)**

**Dự án: Hệ thống Quản lý Phòng khám "Nụ Cười Việt" (NCV-CMS)**

Sinh viên thực hiện: Nguyễn Văn Ánh – CNTT4

Giáo viên hướng dẫn: Nguyễn Quảng An – Nguyễn Xuân Thức

Mục Lục

[CHƯƠNG 1. GIỚI THIỆU (INTRODUCTION) 3](#_Toc214164745)

[1.1. Mục đích 3](#_Toc214164746)

[1.2. Bối cảnh và Vấn đề cần giải quyết 3](#_Toc214164747)

[1.3. Phạm vi của Hệ thống Chức năng trong phạm vi (In-scope): 3](#_Toc214164748)

[1.4. Định nghĩa, Từ viết tắt 4](#_Toc214164749)

[CHƯƠNG 2. MÔ TẢ TỔNG QUAN (OVERALL DESCRIPTION) 4](#_Toc214164750)

[2.1. Bối cảnh Sản phẩm 4](#_Toc214164751)

[2.2. Kiến trúc hệ thống 5](#_Toc214164752)

[2.3. Các chức năng chính của Hệ thống 6](#_Toc214164753)

[2.4. Đặc điểm Người dùng (Actors) 7](#_Toc214164754)

[2.5. Các ràng buộc (Constraints) 8](#_Toc214164755)

[CHƯƠNG 3. YÊU CẦU CỤ THỂ (SPECIFIC REQUIREMENTS) 8](#_Toc214164756)

[3.1. Yêu cầu về Giao diện Người dùng (User Interface Requirements) 8](#_Toc214164757)

[3.1.1. Wireframe 1: Màn hình Quản lý Lịch (Lễ tân) 8](#_Toc214164758)

[3.1.2. Wireframe 2: Màn hình Chi tiết Bệnh nhân 10](#_Toc214164759)

[3.1.3. Wireframe 3: Màn hình Tạo Phiếu khám (Nha sĩ) 12](#_Toc214164760)

[3.2. Yêu cầu Chức năng (Functional Requirements) 13](#_Toc214164761)

[3.2.1 UC-01: Đặt Lịch Hẹn Mới 13](#_Toc214164762)

[3.2.2 UC-02: Hủy Lịch Hẹn 15](#_Toc214164763)

[3.2.3 UC-03: Check-in Bệnh nhân (Khi đến khám) 17](#_Toc214164764)

[3.2.4 UC-04: Ghi nhận Phiếu Khám (Medical Record) 19](#_Toc214164765)

[3.2.5 UC-05: Thanh toán & Tạo Hóa đơn 21](#_Toc214164766)

[3.2.6 UC-06: Quản lý Danh mục Dịch vụ 22](#_Toc214164767)

[3.2.7 UC-07: Quản lý Nhân viên 22](#_Toc214164768)

[3.3. Yêu cầu Phi chức năng (Non-functional Requirements) 23](#_Toc214164769)

[3.4. Yêu cầu về Cơ sở dữ liệu Logic 24](#_Toc214164770)

[3.4.1 Thực thể & Thuộc tính (Entities & Attributes) 24](#_Toc214164771)

[3.4.2. Mối quan hệ & Khóa ngoại (Relationships & Foreign Keys) 26](#_Toc214164772)

[3.4.3 Cardinality & Optionality (Cách hiểu bằng phát biểu và ký hiệu) 27](#_Toc214164773)

[PHỤ LỤC (APPENDIX) 29](#_Toc214164774)

[*A* Biểu đồ Lớp (Class Diagram - Domain Model) 29](#_Toc214164775)

[B Biểu đồ Class Diagram 30](#_Toc214164776)

# CHƯƠNG 1. GIỚI THIỆU (INTRODUCTION)

## Mục đích

Tài liệu **Đặc tả yêu cầu phần mềm (Software Requirement Specification - SRS)** này được xây dựng nhằm mô tả chi tiết các yêu cầu nghiệp vụ, chức năng và phi chức năng cho hệ thống **Quản lý Phòng khám Nụ Cười Việt (NCV-CMS)**.  
Tài liệu được biên soạn cho các đối tượng gồm:

* **Khách hàng (Chủ phòng khám)** – để hiểu rõ hệ thống sẽ đáp ứng những gì.
* **Đội ngũ phát triển** – để triển khai thiết kế, lập trình và kiểm thử hệ thống theo đúng yêu cầu.
* **Giảng viên và người đánh giá** – để thẩm định tính đầy đủ, hợp lý của yêu cầu phần mềm.

Mục đích chính của hệ thống NCV-CMS là **số hóa toàn bộ quy trình hoạt động của phòng khám nha khoa**, giúp **giảm sai sót trong đặt lịch, tăng hiệu quả quản lý bệnh nhân, tự động hóa ghi nhận điều trị và thanh toán**, từ đó **nâng cao tính chuyên nghiệp và trải nghiệm khách hàng.**

## Bối cảnh và Vấn đề cần giải quyết

Phòng khám nha khoa tư nhân "Nụ Cười Việt" hiện có 2 nha sĩ chính và 1 nhân viên lễ tân. Mọi hoạt động từ đặt lịch hẹn, quản lý hồ sơ bệnh nhân, đến ghi chép liệu trình điều trị đều đang được thực hiện thủ công qua sổ sách và các file Excel rời rạc. Điều này dẫn đến nhiều vấn đề:

* Trùng lặp lịch hẹn: Nhân viên lễ tân thường xuyên ghi nhầm hoặc xếp chồng hai cuộc hẹn vào cùng một khung giờ của một nha sĩ, gây phiền toái cho cả bệnh nhân và nha sĩ.
* Khó tra cứu hồ sơ: Việc tìm lại lịch sử khám bệnh, các ghi chú, hoặc phim X-quang của một bệnh nhân cũ rất mất thời gian.
* Quên lịch tái khám: Không có cơ chế tự động nhắc nhở bệnh nhân về lịch tái khám, dẫn đến tỷ lệ bệnh nhân quay lại thấp.
* Thiếu chuyên nghiệp: Quy trình làm việc thủ công tạo cảm giác thiếu tin cậy và không hiện đại trong mắt khách hàng.

Do đó, việc xây dựng một **hệ thống web nội bộ quản lý toàn diện** là cần thiết để tối ưu quy trình làm việc, giảm sai sót, và cải thiện hình ảnh thương hiệu.

## 1.3. Phạm vi của Hệ thống **Chức năng trong phạm vi (In-scope):**

* **Quản lý Bệnh nhân:** Tạo, cập nhật, tìm kiếm hồ sơ bệnh nhân.
* **Quản lý Lịch hẹn:** Đặt lịch, hủy lịch, check-in, nhắc lịch tự động.
* **Quản lý Điều trị:** Ghi nhận phiếu khám, chuẩn đoán, thêm dịch vụ điều trị.
* **Thanh toán:** Tính chi phí và tạo hóa đơn cho bệnh nhân.
* **Quản lý Danh mục:** Quản lý dịch vụ, đơn giá, nhân viên và phân quyền.
* **Tích hợp hệ thống SMS/Zalo Gateway:** Tự động gửi tin nhắn nhắc lịch.

**Ngoài phạm vi (Out-of-scope):**

* Hệ thống **không quản lý kho thuốc** hoặc vật tư y tế.
* Hệ thống **không tích hợp thanh toán trực tuyến** (chỉ ghi nhận thủ công).
* Hệ thống **không lưu trữ hình ảnh X-quang** hoặc dữ liệu hình ảnh y khoa.
* Hệ thống **không cung cấp cổng truy cập cho bệnh nhân** (chỉ dùng nội bộ).

## 1.4. Định nghĩa, Từ viết tắt

|  |  |
| --- | --- |
| Thuật ngữ | Giải thích |
| **NCV-CMS** | Nụ Cười Việt - Clinic Management System |
| **SRS** | Software Requirement Specification – Tài liệu Đặc tả Yêu cầu Phần mềm |
| **Lễ tân** | Nhân viên tiếp nhận, đặt lịch và xử lý thanh toán |
| **Nha sĩ** | Người khám và điều trị cho bệnh nhân |
| **Quản lý** | Người có quyền cao nhất, quản lý danh mục và nhân sự |
| **Hồ sơ bệnh nhân** | Tập hợp thông tin cá nhân, lịch sử khám và điều trị của bệnh nhân |
| **Phiếu khám** | Ghi nhận chi tiết quá trình khám và các dịch vụ đã thực hiện |
| **SMS Gateway** | Hệ thống bên ngoài dùng để gửi tin nhắn nhắc lịch tự động |

# CHƯƠNG 2. MÔ TẢ TỔNG QUAN (OVERALL DESCRIPTION)

## 2.1. Bối cảnh Sản phẩm

Hệ thống NCV-CMS là **ứng dụng web nội bộ** dành riêng cho phòng khám Nụ Cười Việt, được triển khai trên mạng nội bộ hoặc máy chủ đám mây (cloud).  
Người dùng (lễ tân, nha sĩ, quản lý) có thể truy cập hệ thống thông qua trình duyệt web trên máy tính hoặc máy tính bảng tại quầy tiếp tân và trong phòng khám.

## 2.2. Kiến trúc hệ thống

Hệ thống được thiết kế theo **mô hình 3 tầng (Three-tier Architecture)** gồm:

1. **Tầng Giao diện (Presentation Layer)**

* Cung cấp giao diện web cho người dùng (HTML, CSS, JavaScript, ReactJS...).
* Cho phép nhập và hiển thị dữ liệu, xử lý các thao tác như đặt lịch, xem hồ sơ,...

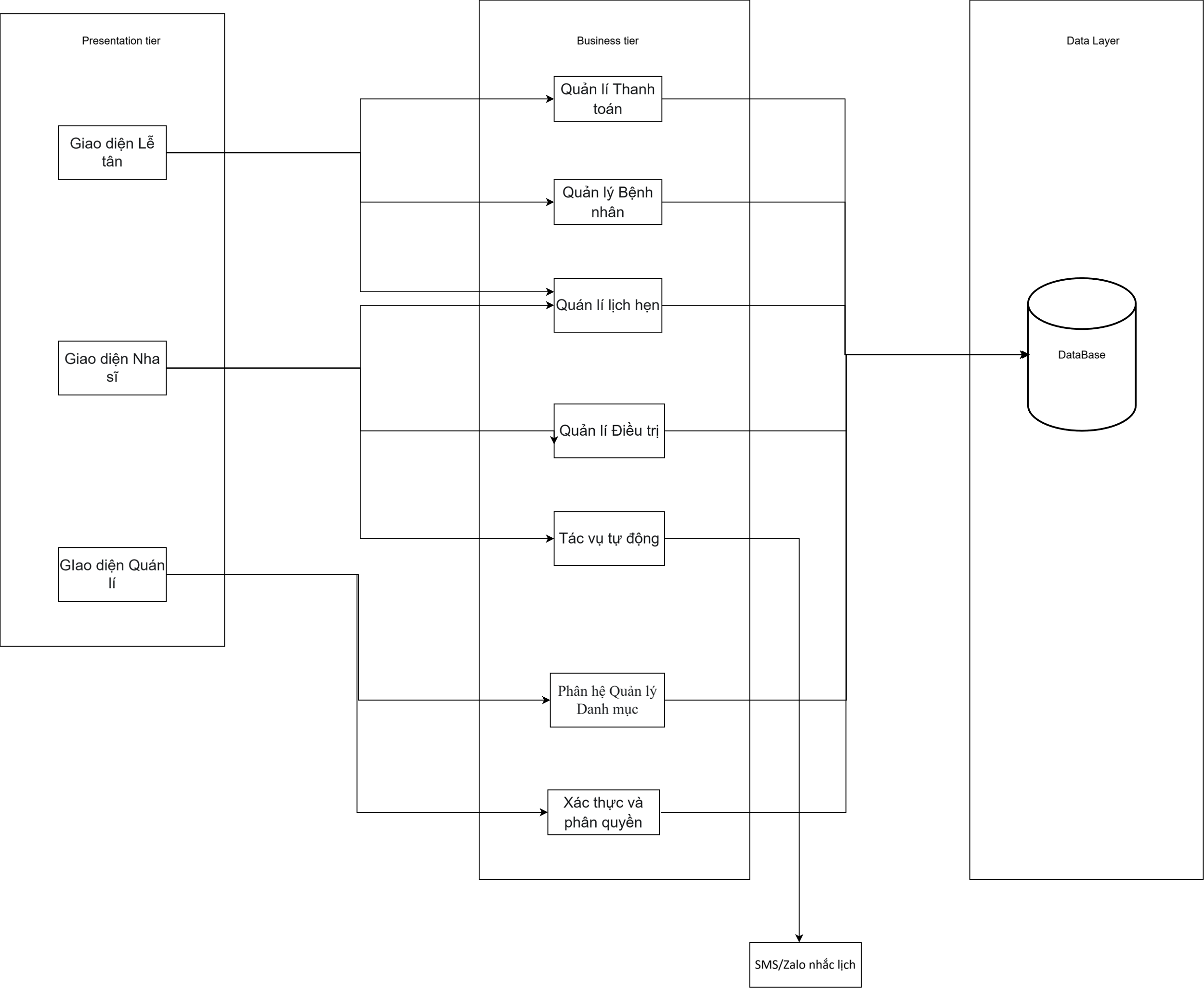
1. **Tầng Ứng dụng (Application/Business Logic Layer)**

* Xử lý các quy trình nghiệp vụ: xác nhận lịch, tính chi phí, phân quyền truy cập,...
* Được phát triển bằng Node.js/Java Spring hoặc tương tự.

1. **Tầng Dữ liệu (Data Layer)**

* Quản lý toàn bộ dữ liệu trong CSDL (MySQL, PostgreSQL,...).
* Đảm bảo tính toàn vẹn, bảo mật và đồng nhất dữ liệu.

#### **Sơ đồ minh họa kiến trúc 3 tầng:**

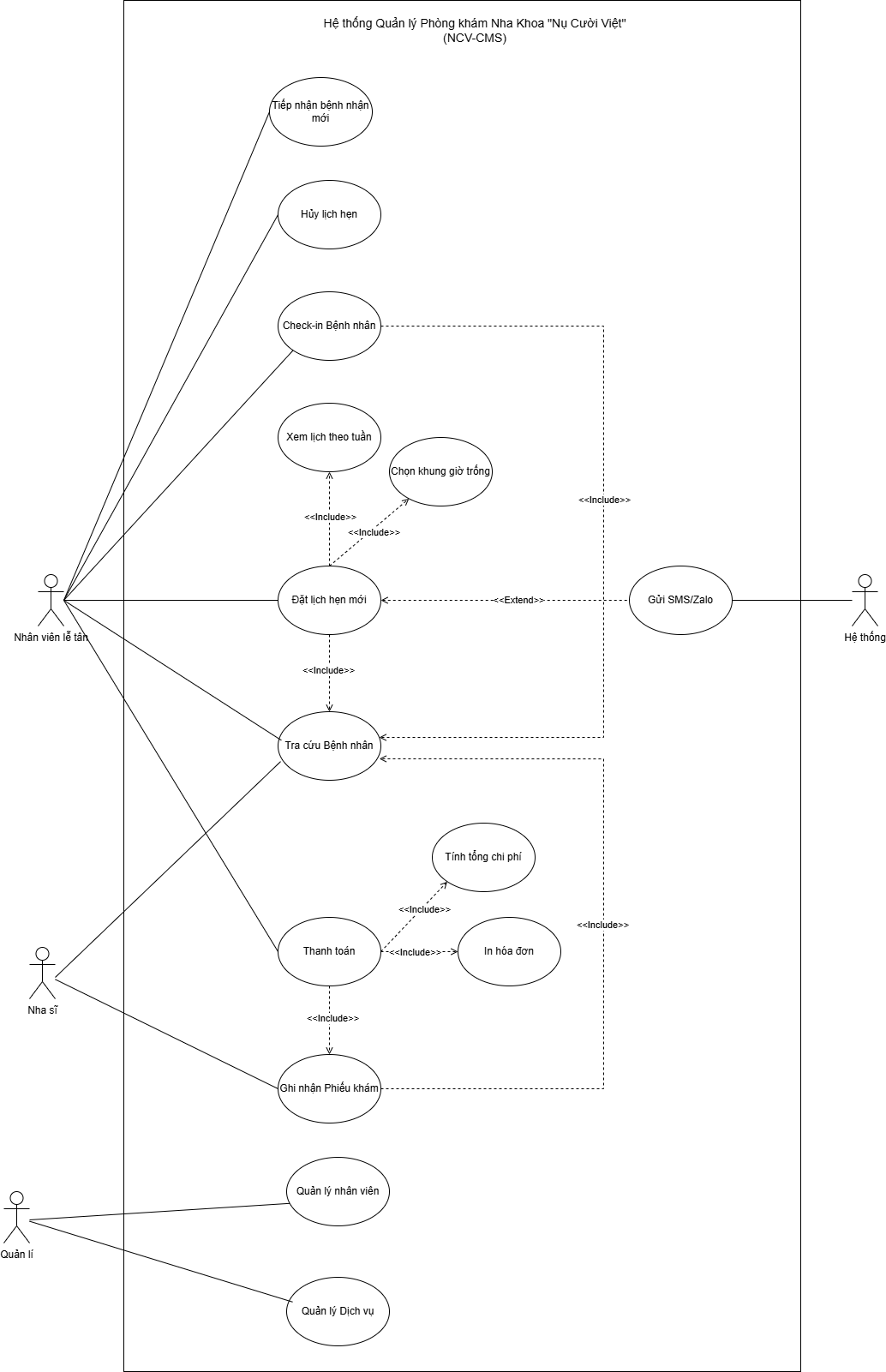


## 2.3. Các chức năng chính của Hệ thống

Hệ thống bao gồm các nhóm chức năng chính:

1. **Quản lý Bệnh nhân**
2. **Quản lý Lịch hẹn**
3. **Quản lý Điều trị & Phiếu khám**
4. **Quản lý Thanh toán & Hóa đơn**
5. **Quản lý Dịch vụ & Nhân viên**
6. **Tác vụ tự động (SMS/Zalo nhắc lịch)**
7. **Phân quyền người dùng và bảo mật**

**Biểu đồ Use Case tổng thể:**

****

## 2.4. Đặc điểm Người dùng (Actors)

|  |  |
| --- | --- |
| Actor | Mô tả vai trò |
| **Lễ tân** | Tiếp nhận bệnh nhân, đặt lịch hẹn, hủy lịch, check-in và thu tiền sau khám. |
| **Nha sĩ** | Thực hiện khám, ghi phiếu điều trị, chuẩn đoán, cập nhật dịch vụ đã làm. |
| **Quản lý** | Quản lý nhân viên, danh mục dịch vụ, xem báo cáo tổng hợp. |
| **SMS Gateway** | Hệ thống bên ngoài, tự động gửi tin nhắn nhắc lịch cho bệnh nhân. |

**Use Cases (chính miêu tả) :**

1. Quản lý hồ sơ bệnh nhân
2. Tra cứu bệnh nhân
3. Đặt lịch hẹn
4. Hủy lịch hẹn
5. Check-in bệnh nhân
6. Ghi nhận phiếu khám
7. Thanh toán và lập hóa đơn
8. Quản lý dịch vụ và nhân viên
9. Gửi nhắc lịch tự động (extend từ “Đặt lịch hẹn”)

**Quan hệ:**

## 2.5. Các ràng buộc (Constraints)

* **Kỹ thuật:**
  + Hệ thống phải chạy ổn định trên trình duyệt Google Chrome, Edge.
  + Dữ liệu được lưu trong CSDL quan hệ MySQL hoặc tương đương.
  + Thời gian tải lịch hẹn Mỗi lịch hẹn chỉ có **1 nha sĩ và 1 bệnh nhân**.
  + Một khung giờ của nha sĩ chỉ có **1 lịch ở trạng thái “Đã xác nhận”.**
  + Lễ tân không được chỉnh sửa thông tin điều trị.
  + Tạo hồ sơ và đặt lịch mới phải hoàn tất **< 3 phút.**
  + Dữ liệu bệnh nhân phải được bảo mật tuyệt đối.

# CHƯƠNG 3. YÊU CẦU CỤ THỂ (SPECIFIC REQUIREMENTS)

## 3.1. Yêu cầu về Giao diện Người dùng (User Interface Requirements)

Giao diện sẽ tuân thủ các nguyên tắc thiết kế hiện đại, tập trung vào sự **rõ ràng, nhất quán và đơn giản**. Màu sắc nhẹ nhàng, bố cục phân vùng rõ rệt, các hành động chính (Đặt lịch, Check-in, Lưu phiếu khám, Xác nhận thanh toán) có nút lớn, dễ thấy.

### ****3.1.1. Wireframe 1: Màn hình Quản lý Lịch (Lễ tân)****

#### **3.1.1.1. Mục đích màn hình**

Cho phép lễ tân theo dõi toàn bộ lịch làm việc theo tuần của nha sĩ, xem lịch hẹn hôm nay, tìm bệnh nhân và thao tác nhanh như: đặt lịch, check-in bệnh nhân, hủy lịch.

#### **3.1.1.2. Bố cục tổng thể**

Màn hình được chia thành **2 phần chính**:

##### **3.1.1.2.1. Sidebar trái – Bộ lọc & Danh sách bệnh nhân hôm nay**

**Kích thước:** 25–30% chiều ngang màn hình.

**Thành phần chi tiết:**

**Khung tìm kiếm bệnh nhân**

Input tìm theo:

* Mã bệnh nhân (Patient ID)
* Họ tên
* Số điện thoại
* Nút "Tìm kiếm" hoặc auto-search khi nhập.

**Danh sách bệnh nhân đến trong ngày**  
Mỗi item hiển thị:

* Tên bệnh nhân
* Giờ hẹn
* Nha sĩ phụ trách
* Trạng thái: Chờ khám / Đã xác nhận / Đã check-in / Đã hủy
* Badge màu thể hiện trạng thái
* Xác nhận -> Xanh biển
* Đang chờ khám -> Cam
* Đã hủy -> Đỏ

**Tương tác**:

Click vào một bệnh nhân -> mở trang Chi tiết bệnh nhân.

Hover hiển thị tooltip nhanh: lịch sử 3 lần khám gần nhất.

##### **3.1.1.2.2. Phần bên phải - Lịch theo tuần của Nha sĩ**

**Kích thước:** 70–75% màn hình.

**Thành phần:**

**Header**

* Dropdown **"Chọn nha sĩ"**
* Thông tin nhanh dạng số liệu:
* **Số bệnh nhân hôm nay**
* **Số bệnh nhân đã check-in**
* Bộ nút thao tác nhanh (button group ở góc phải):
* **Đặt lịch mới** (nút lớn, màu chủ đạo -> dễ nhìn)
* **Check-in** (bật khi chọn bệnh nhân trong danh sách)
* **Hủy lịch**

##### **3.1.1.2.3. Lịch tuần của nha sĩ**

Bố cục dạng **lưới 7 cột** (Thứ 2 - Chủ nhật)

Hàng thời gian: **mỗi ô 30 phút** (VD: 8:00, 8:30, 9:00...)

Slot có lịch sẽ hiển thị **block màu**, gồm:

* Tên bệnh nhân
* Dịch vụ chính
* Ghi chú trạng thái
* Màu trạng thái:
* Xanh lá - **Đã xác nhận**
* Cam - **Đang chờ khám**
* Đỏ nhạt - **Đã hủy**

Click vào block -> mở popup chi tiết cuộc hẹn:

* Thông tin bệnh nhân
* Ghi chú
* Các nút: Check-in, Chỉnh sửa, Hủy

#### **3.1.1.3 Kéo giữa các slot -> đổi giờ hẹn (drag & drop)**

Double-click vào slot trống -> yêu cầu “Tạo lịch mới”

Lịch tự động refresh mỗi 30s

### ****3.1.2. Wireframe 2: Màn hình Chi tiết Bệnh nhân****

#### **3.1.2.1. Mục đích màn hình**

Hiển thị thông tin bệnh nhân + toàn bộ lịch sử khám, hóa đơn, nhắc lịch.  
Là nơi để lễ tân và nha sĩ thao tác trên hồ sơ.

##### **3.1.2.*2. Bố cục màn hình***

##### **3.1.2.*2.*1. Header thông tin bệnh nhân**

Hiển thị lớn và rõ ràng:

Họ và tên (cỡ chữ lớn)

* Ngày sinh
* Giới tính
* Số điện thoại
* Địa chỉ

Trạng thái tài khoản:

* Đang hoạt động
* Ngưng hoạt động
* Lần khám gần nhất

Cạnh phải header có 2 nút:

* **In thẻ bệnh nhân**
* **In biên nhận**

##### **3.1.2.*2*.2. Phân vùng theo Tab**

Có 4 tab chính:

**Tab 1 – Lịch hẹn**

* Hiển thị danh sách lịch hẹn của bệnh nhân:
* Ngày – giờ
* Nha sĩ phụ trách
* Dịch vụ
* Trạng thái (Xác nhận / Đang chờ khám / Đã hủy / Đã khám)
* Nút nổi bật: **“Đặt lịch”** (cho phép lễ tân tạo lịch mới cho bệnh nhân).

**Tab 2 – Phiếu khám (Medical Record / Clinical Notes)**

* Hiển thị dạng bảng:
* Mã phiếu khám
* Ngày khám
* Nha sĩ khám
* Chuẩn đoán
* Ghi chú
* Nút: **Tạo phiếu khám**  
  -> chỉ nha sĩ mới được phép bấm (role permission).
* Click vào 1 phiếu -> xem chi tiết phiếu khám.

**Tab 3 – Hóa đơn**

Danh sách hóa đơn kèm trạng thái:

* Mã hóa đơn
* Ngày tạo
* Tổng tiền
* Trạng thái: Đã thanh toán / Chưa thanh toán
* Click vào -> mở chi tiết hóa đơn.

**Tab 4 – Nhắc lịch**

* Danh sách các nhắc lịch tự động
* Thời gian gửi
* Nội dung
* Kết quả gửi (Thành công / Lỗi / Khách không phản hồi)

##### **3.1.2.*2*.*3. Hành vi***

* Click vào avatar bệnh nhân -> sửa thông tin
* Tab tự động load dữ liệu khi chuyển tab
* Trạng thái đổi màu theo từng mục

### ****3.1.3. Wireframe 3: Màn hình Tạo Phiếu khám (Nha sĩ)****

#### **3.1.3.1. Mục đích**

Cho nha sĩ tạo hồ sơ khám, nhập chuẩn đoán, ghi chú và chọn dịch vụ thực hiện.

#### **3.1.3.2. Bố cục chi tiết**

#### **3.1.3.2.1. Header**

Hiển thị:

* Tên bệnh nhân
* Giờ hẹn
* Nha sĩ phụ trách
* Trạng thái: “Đang khám”

Có nút phụ:

* **Quay lại chi tiết bệnh nhân**

##### **3.1.3.2.2. Phần nhập Phiếu khám (bên trái)**

***(1) Chuẩn đoán***

* Textbox lớn dạng multiline (5–7 dòng)

***(2) Ghi chú thêm***

* Textbox multiline cho ghi chú nội bộ hoặc mô tả chi tiết

***(3) Chọn dịch vụ***

* Dropdown chọn dịch vụ
* Nội dung dropdown hiển thị: Tên dịch vụ + Đơn giá
* Nút **Add** để thêm dịch vụ vào đơn khám
* Validation: Không chọn dịch vụ mà bấm Add -> cảnh báo.

##### **3.1.3.2.3. Danh sách dịch vụ đã chọn (bên phải)**

Bảng có các cột:

* Dịch vụ
* Đơn giá
* Số lượng (default = 1, cho phép chỉnh)
* Tổng tiền từng dòng
* Nút “Xóa”

Phần dưới cùng:

* **Tạm tính:** Tổng tiền các dịch vụ
* **Giảm giá:** (nếu có)
* **Tổng cuối:** Bold – dễ thấy

##### **3.1.3.2.4. Footer – Nút hệ thống**

Hai nút lớn, nằm trên cùng một hàng:

* **Lưu phiếu khám**
* Lưu tạm
* Không chuyển trạng thái bệnh nhân
* **Kết thúc khám -> Chuyển sang Chờ thanh toán**
* Đổi trạng thái lịch hẹn thành “Chờ thanh toán”
* Tự động tạo Hóa đơn nháp (draft)
* Điều hướng lễ tân đến màn hình Thanh toán

##### **3.1.3.2.5. Hành vi**

Dịch vụ thêm vào -> cập nhật tổng ngay

Nếu rời trang khi chưa lưu -> popup cảnh báo

Nhập chuẩn đoán để trống -> không cho kết thúc khám

## 3.2. Yêu cầu Chức năng (Functional Requirements)

### ****3.2.1 UC-01: Đặt Lịch Hẹn Mới****

**Mô tả ngắn:**  
Nhân viên lễ tân tạo một lịch hẹn mới cho bệnh nhân (có thể là bệnh nhân đã có hồ sơ hoặc tạo hồ sơ mới tại chỗ).

**Actor chính:**  
Nhân viên lễ tân

**Điều kiện tiên quyết (Pre-conditions):**

* Nhân viên vào hệ thống
* Nha sĩ có lịch trống vào ngày/giờ được chọn.

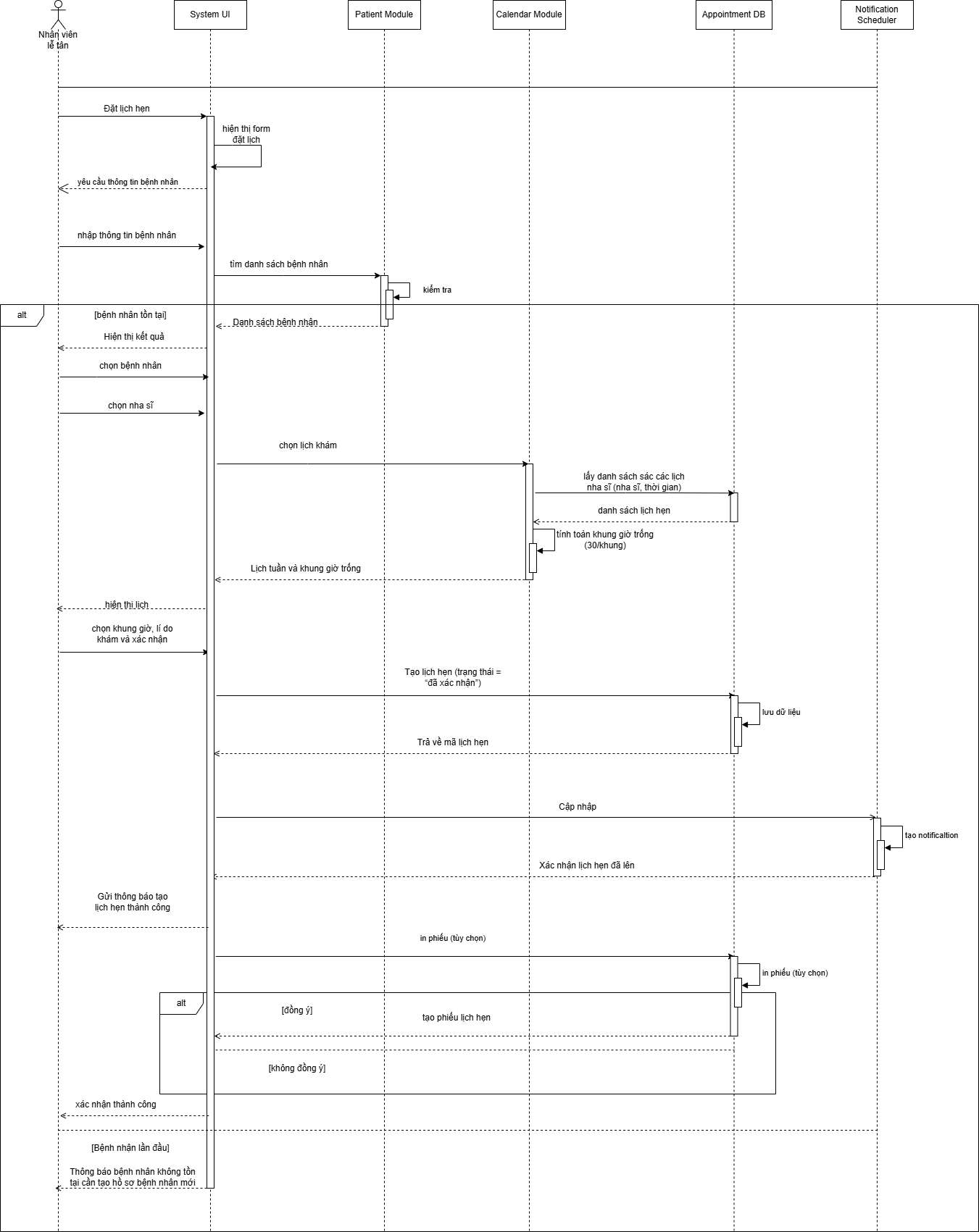
**Luồng sự kiện chính (Main Flow):**

1. Nhân viên chọn chức năng **“Đặt lịch hẹn”**.
2. Nhân viên tìm kiếm và chọn bệnh nhân (bằng mã, tên hoặc số điện thoại).
3. Nhân viên chọn nha sĩ và xem **lịch làm việc theo tuần** (mỗi khung 30 phút).
4. Hệ thống hiển thị các khung giờ đã được đặt (đánh dấu màu).
5. Nhân viên chọn khung giờ trống phù hợp.
6. Nhập **Lý do khám**.
7. Nhân viên xác nhận; hệ thống tạo bản ghi **Appointment** với status = "Đã xác nhận".
8. Hệ thống tạo **NotificationLog** để gửi nhắc lịch 24 giờ trước khi khám.
9. Hệ thống hiển thị thông báo xác nhận và (tùy chọn) in **phiếu hẹn**.

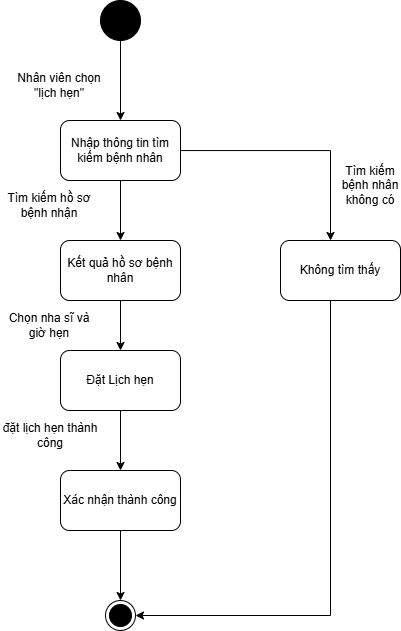
**Luồng thay thế (Alternative Flows):**

* Bệnh nhân không tồn tại -> thông báo lễ tân cần tạo hồ sơ bệnh nhân mới

**Sequence Diagram**



**State Diagram (Appointment):**



### ****3.2.2 UC-02: Hủy Lịch Hẹn****

**Mô tả ngắn:** Nhân viên lễ tân hủy một lịch hẹn theo yêu cầu bệnh nhân hoặc do thay đổi nội bộ.  
**Actor chính:** Nhân viên lễ tân

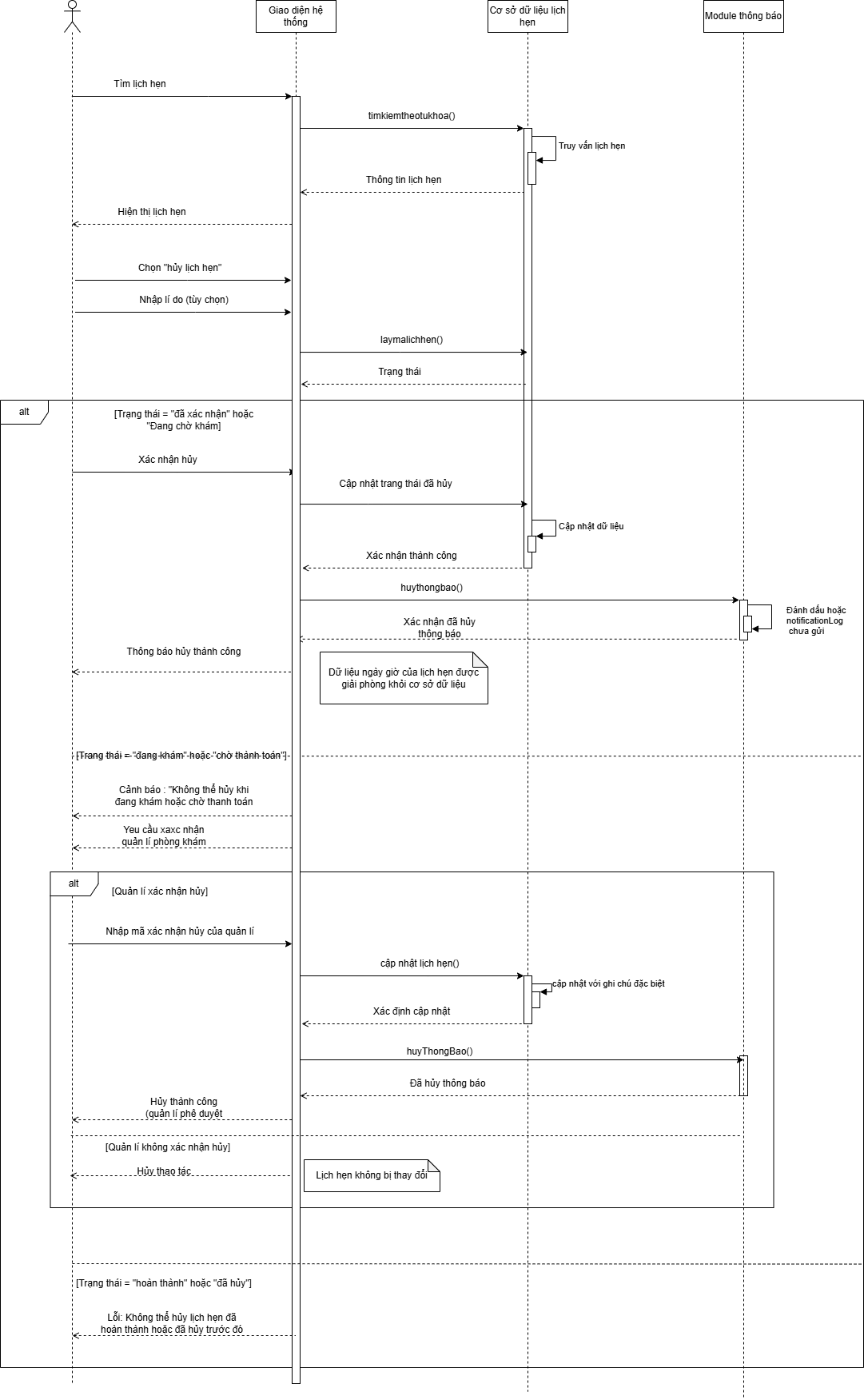
**Điều kiện tiên quyết:**  
Appointment cần hủy đã được tìm thấy.

**Luồng sự kiện chính:**

1. Nhân viên tìm appointment (bằng mã, tên hoặc xem trên lịch).
2. Nhân viên chọn chức năng **“Hủy”** và nhập **lý do hủy (tùy chọn)**.
3. Hệ thống cập nhật status = "Đã hủy", ghi cancelled\_at = now.
4. Hệ thống hủy hoặc đánh dấu các **NotificationLog** chưa gửi liên quan.
5. Hệ thống hiển thị thông báo hủy thành công.

**Luồng thay thế:**

* Nếu appointment đang ở trạng thái “Đang khám” hoặc “Chờ thanh toán”, hệ thống cảnh báo và yêu cầu **xác nhận từ quản lý**.
* Nếu hóa đơn đã thanh toán hoặc appointment đã hủy ở trước đó, hiện thông báo.

**Sequence Diagram:**  
****

**State Diagram**

****

### ****3.2.3 UC-03: Check-in Bệnh nhân (Khi đến khám)****

**Mô tả ngắn:**  
Nhân viên lễ tân thực hiện check-in khi bệnh nhân đến phòng khám theo lịch hẹn.

**Actor chính:**  
Nhân viên lễ tân

**Điều kiện tiên quyết:**

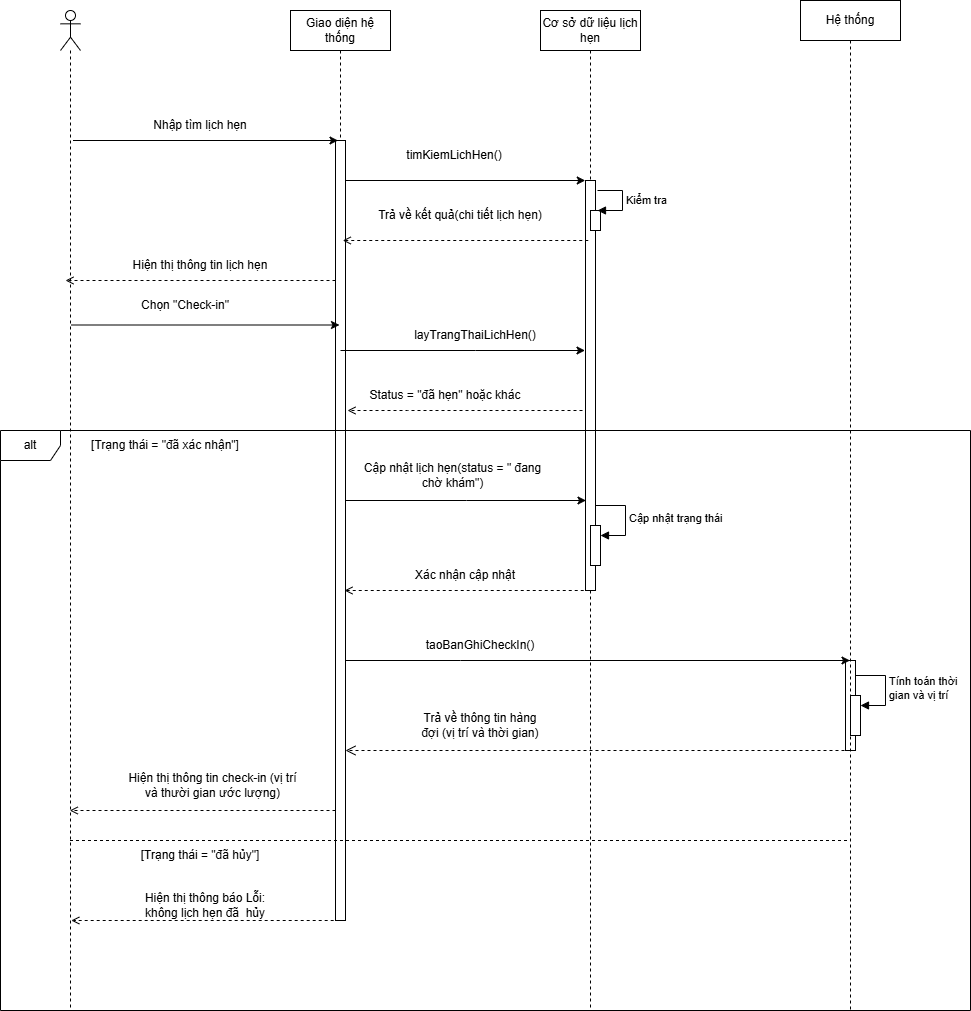
* Appointment tồn tại với status = 'Đã xác nhận'.

**Luồng sự kiện chính:**

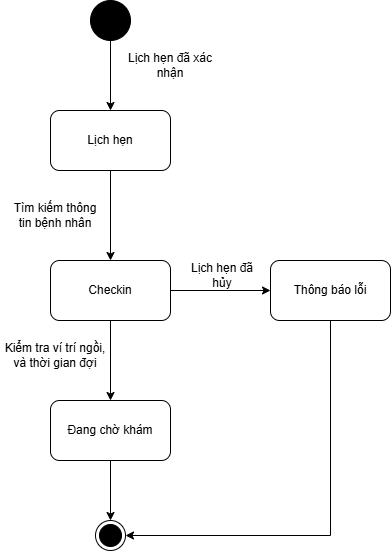
1. Nhân viên tìm appointment (theo tên, mã, hoặc quét mã QR trên phiếu hẹn).
2. Nhân viên bấm **“Check-in”**.
3. Hệ thống tạo bản ghi **CheckinQueue** (queued\_at = now) và cập nhật Appointment.status = 'Đang chờ khám'.
4. Bệnh nhân xuất hiện trong **hàng đợi điện tử**, hiển thị vị trí và thời gian ước lượng chờ.
5. Hệ thống có thể gửi thông báo nội bộ đến phòng khám (nếu có màn hình hiển thị hàng đợi).

**Luồng thay thế:**

* Nếu appointment đã bị hủy, hệ thống thông báo lỗi và không cho check-in.

**Sequence Diagram:**  
****

**State Diagram**

****

### ****3.2.4 UC-04: Ghi nhận Phiếu Khám (Medical Record)****

**Mô tả ngắn:**  
Nha sĩ ghi nhận phiếu khám và các dịch vụ thực hiện trong quá trình khám bệnh.

**Actor chính:**  
Nha sĩ

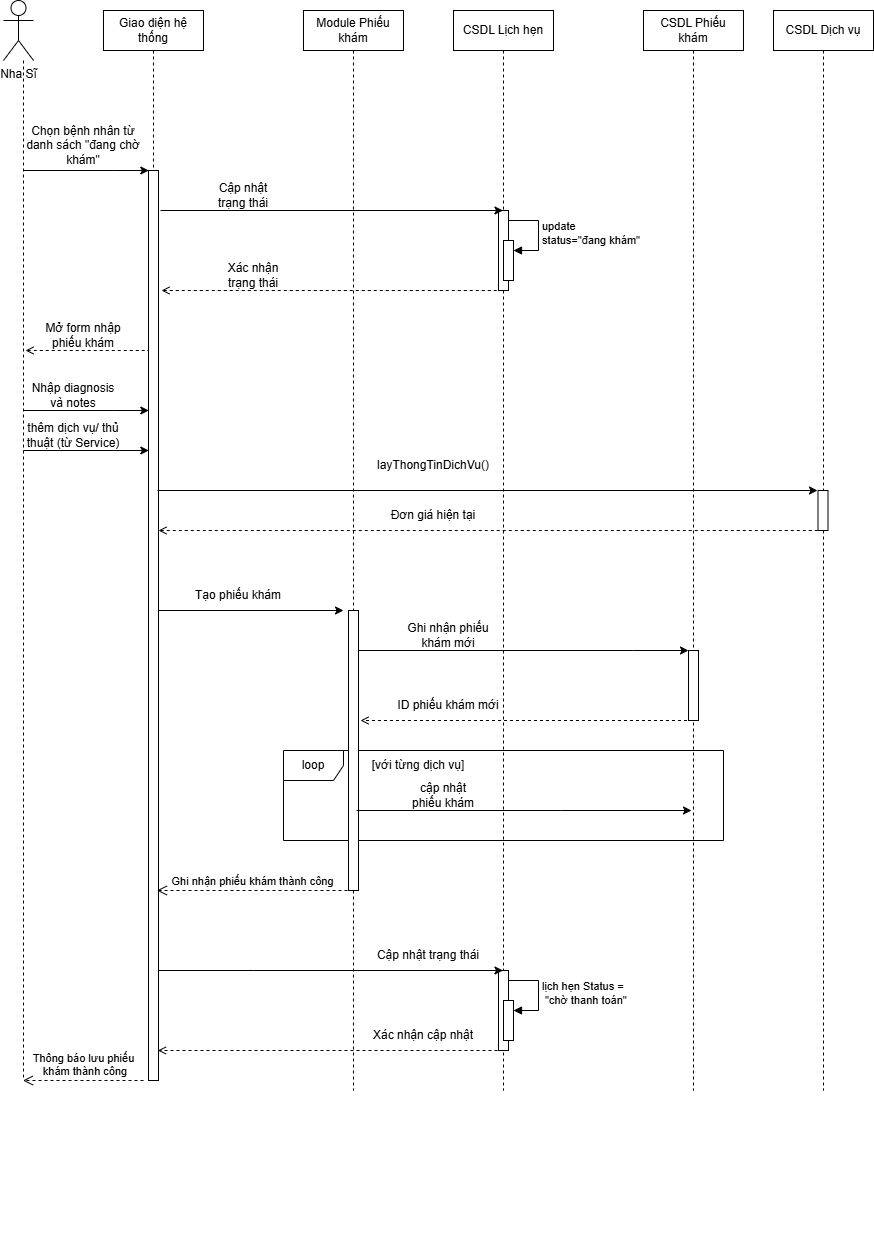
**Điều kiện tiên quyết:**

* Appointment đang ở trạng thái 'Đang khám' hoặc nha sĩ mở trực tiếp từ danh sách bệnh nhân chờ.

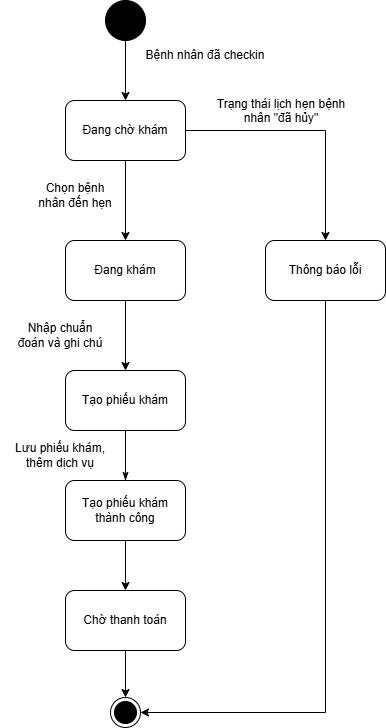
**Luồng sự kiện chính:**

1. Nha sĩ chọn bệnh nhân từ danh sách **“Đang chờ khám”** → hệ thống chuyển Appointment.status → 'Đang khám'.
2. Nha sĩ tạo **MedicalRecord**: nhập chẩn đoán (diagnosis) và ghi chú (notes).
3. Nha sĩ thêm **Dịch vụ/Thủ thuật** (chọn từ danh mục Service) → hệ thống lưu đơn giá tại thời điểm hiện tại (unit\_price).
4. Nha sĩ lưu phiếu khám → hệ thống chuyển Appointment.status → 'Chờ thanh toán'.
5. Hệ thống tạo bản ghi **MedicalRecordService** cho từng dịch vụ.

Sequence Diagram



**State Diagram**

****

### ****3.2.5 UC-05: Thanh toán & Tạo Hóa đơn****

**Mô tả ngắn:**  
Nhân viên lễ tân xử lý thanh toán và hệ thống tạo hóa đơn cho bệnh nhân sau khi khám.

**Actor chính:**  
Nhân viên lễ tân

**Điều kiện tiên quyết:**

* Appointment có status = 'Chờ thanh toán'.
* MedicalRecord đã tồn tại.

**Luồng sự kiện chính:**

1. Nhân viên mở chi tiết appointment hoặc phiếu khám.
2. Hệ thống hiển thị **chi tiết dịch vụ** và **tổng tiền cần thanh toán**.
3. Nhân viên nhận tiền, chọn **phương thức thanh toán** (Tiền mặt / Chuyển khoản / Khác).
   * Nếu bệnh nhân chỉ trả một phần -> ghi paid\_amount < total\_amount.
4. Nhân viên bấm **“Xác nhận thanh toán”** -> hệ thống tạo **Invoice**, cập nhật:
   * Appointment.status = 'Hoàn thành'
   * Cập nhật patient.balance nếu có ghi nợ.
5. Hệ thống cho phép **in hóa đơn / biên lai**.

**Luồng thay thế:**

* Thanh toán nhiều lần → hệ thống cho phép tạo nhiều bản ghi thanh toán (phiếu thu tạm).

### ****3.2.6 UC-06: Quản lý Danh mục Dịch vụ****

**Mô tả ngắn:**  
Quản lý phòng khám thêm, sửa hoặc vô hiệu hóa dịch vụ nha khoa và đơn giá.

**Actor chính:**  
Quản lý phòng khám (Admin)

**Điều kiện tiên quyết:**

* Quản lý đã đăng nhập bằng tài khoản có quyền **Admin**.

**Luồng sự kiện chính:**

1. Quản lý truy cập **màn hình Quản lý Dịch vụ**.
2. Thực hiện thêm, sửa hoặc xóa dịch vụ.
3. Hệ thống kiểm tra ràng buộc:
   * Không được xóa dịch vụ đã sử dụng trong **MedicalRecord**.
   * Chỉ cho phép **vô hiệu hóa (Deactivate)** dịch vụ thay vì xóa cứng.
4. Hệ thống lưu thay đổi và cập nhật danh mục hiển thị.

### ****3.2.7 UC-07: Quản lý Nhân viên****

**Mô tả ngắn:**  
Quản lý phòng khám (Admin) thực hiện thêm mới, chỉnh sửa hoặc vô hiệu hóa tài khoản nhân viên trong hệ thống. Mỗi nhân viên có thông tin cá nhân cơ bản, tài khoản đăng nhập, và vai trò (Lễ tân / Nha sĩ ).

**Actor chính:**  
Quản lý phòng khám (Admin)

**Điều kiện tiên quyết :**

* Quản lý đã đăng nhập bằng tài khoản có quyền “Admin”.
* Hệ thống có sẵn danh sách vai trò (Role) và bảng nhân viên (Employee).

**Luồng sự kiện chính :**

1. Quản lý chọn menu **“Quản lý Nhân viên”**.
2. Hệ thống hiển thị danh sách nhân viên hiện tại, gồm: Mã nhân viên, Họ tên, Vai trò, Trạng thái (Hoạt động / Vô hiệu).

**Quản lý có thể thực hiện một trong ba tác vụ sau:**

**a. Thêm nhân viên mới:**

* Nhập thông tin cá nhân (Họ tên, Giới tính, Số điện thoại, Tên đăng nhập, Mật khẩu, Vai trò).
* Hệ thống kiểm tra trùng username.
* Nếu hợp lệ → Lưu và hiển thị trong danh sách.

b. **Sửa thông tin nhân viên:**

* Chọn một nhân viên → sửa thông tin (số điện thoại, vai trò, mật khẩu, v.v.).
* Hệ thống ghi lại lịch sử thay đổi (AuditLog).

c. **Vô hiệu hóa nhân viên:**

* Chọn nhân viên → chọn “Vô hiệu hóa”.
* Hệ thống cập nhật trạng thái status = Inactive nhưng vẫn giữ dữ liệu.

1. Sau khi thực hiện, hệ thống hiển thị thông báo thành công và cập nhật danh sách hiển thị.

**Luồng thay thế (Alternative Flows):**

* **A1:** Nếu username đã tồn tại → Hệ thống báo lỗi “Tên đăng nhập đã tồn tại”, yêu cầu nhập lại.
* **A2:** Nếu quản lý cố xóa nhân viên đang có lịch hẹn hoặc phiếu khám liên quan → Hệ thống chặn thao tác và đề nghị “Vô hiệu hóa thay vì xóa”.
* **A3:** Nếu mất kết nối cơ sở dữ liệu → Hệ thống hiển thị thông báo lỗi và hủy thao tác.

## 3.3. Yêu cầu Phi chức năng (Non-functional Requirements)

* **PER-01 (Hiệu năng):** Thời gian tải và hiển thị giao diện lịch làm việc của nha sĩ (một tuần, hiển thị các appointment) phải ≤ 3 giây trên kết nối broadband tiêu chuẩn, với dữ liệu lên đến 2,000 appointment/tháng.
* **PER-02 (Tìm kiếm):** Thời gian phản hồi cho thao tác tìm kiếm bệnh nhân hoặc lịch hẹn phải < 2 giây với dữ liệu 10,000 bệnh nhân giả lập.
* **USA-01 (Tính khả dụng / Usability):** Một nhân viên lễ tân mới, sau khi được hướng dẫn không quá 5 phút, phải có khả năng thực hiện thành thạo 2 nghiệp vụ chính (Đặt lịch, Check-in) mà không cần trợ giúp.
* **SEC-01 (Bảo mật):** Chỉ role='Nha\_si' được phép xem và chỉnh sửa phần Chuẩn đoán/Ghi chú trong MedicalRecord; lễ tân chỉ xem được thông tin tổng quan. Mật khẩu phải được lưu dưới dạng hash an toàn; truyền tải dữ liệu nhạy cảm qua HTTPS.
* **REL-01 (Độ tin cậy / Backup):** Hệ thống phải đảm bảo uptime ≥ 99.5%. Dữ liệu sao lưu tự động hàng ngày; khả năng phục hồi (restore) dữ liệu trong vòng ≤ 4 giờ.
* **AUD-01 (Theo dõi thay đổi):** Mọi thay đổi quan trọng (tạo/hủy lịch, chỉnh sửa phiếu khám, thanh toán) phải được ghi nhận trong AuditLog với thông tin staff\_id và timestamp để kiểm chứng.
* **INT-01 (Tương thích):** Ứng dụng phải hoạt động trên trình duyệt hiện đại (Chrome, Edge, Firefox) và có giao diện responsive cho tablet.

## 3.4. Yêu cầu về Cơ sở dữ liệu Logic

Hệ thống yêu cầu một CSDL quan hệ để quản lý dữ liệu một cách nhất quán và toàn vẹn.

## 3.4.1 Thực thể & Thuộc tính (Entities & Attributes)

Danh sách các bảng chính (thực thể) và thuộc tính quan trọng (ghi chú PK = Primary Key, FK = Foreign Key):

1. **Patient (Bệnh nhân)**

* patient\_id (PK, INT, auto-increment) — Mã bệnh nhân
* full\_name (VARCHAR) — Họ tên
* date\_of\_birth (DATE) — Ngày sinh
* gender (ENUM: 'Nam','Nữ','Khác') — Giới tính
* phone (VARCHAR) — Số điện thoại
* address (VARCHAR) — Địa chỉ
* medical\_history (TEXT) — Ghi chú tiền sử bệnh lý
* status (ENUM: 'Đang hoạt động','Ngưng') — Trạng thái tài khoản
* created\_at (DATETIME)
* updated\_at (DATETIME)

1. **Staff (Nhân viên)**

* staff\_id (PK, INT, auto-increment)
* username (VARCHAR, unique)
* password\_hash (VARCHAR)
* full\_name (VARCHAR)
* role (ENUM: 'Nha\_si','Le\_tan','Quan\_ly')
* phone (VARCHAR)
* email (VARCHAR)
* active (BOOLEAN)
* created\_at (DATETIME)

1. **Service (Dịch vụ)**

* service\_id (PK, INT, auto-increment)
* code (VARCHAR, unique)
* name (VARCHAR)
* description (TEXT)
* price (DECIMAL) — Đơn giá
* duration\_minutes (INT) — (tùy chọn, nếu muốn gắn thời lượng)
* active (BOOLEAN)

1. **Appointment (Lịch hẹn)**

* appointment\_id (PK, INT, auto-increment)
* patient\_id (FK -> Patient.patient\_id)
* staff\_id (FK -> Staff.staff\_id) — nha sĩ được chỉ định
* scheduled\_start (DATETIME) — Thời gian bắt đầu
* scheduled\_end (DATETIME) — Thời gian kết thúc (thường +30 phút)
* reason (VARCHAR) — Lý do khám
* status (ENUM: 'Đã xác nhận','Đã hủy','Đang chờ khám','Đang khám','Chờ thanh toán','Hoàn thành')
* created\_by (FK -> Staff.staff\_id) — ai đặt lịch
* created\_at (DATETIME)
* cancelled\_at (DATETIME, nullable)

1. **CheckinQueue (Hàng đợi / Check-in)**

* checkin\_id (PK, INT, auto-increment)
* appointment\_id (FK -> Appointment.appointment\_id)
* queued\_at (DATETIME)
* status (ENUM: 'Chờ','Đã gọi','Bỏ qua')

1. **MedicalRecord (Phiếu khám / Hồ sơ khám)**

* record\_id (PK, INT, auto-increment)
* appointment\_id (FK -> Appointment.appointment\_id, unique) — 1 phiếu khám ứng với 1 lịch hẹn
* staff\_id (FK -> Staff.staff\_id) — nha sĩ tạo phiếu
* diagnosis (TEXT)
* notes (TEXT)
* created\_at (DATETIME)

1. **MedicalRecordService (Chi tiết dịch vụ trong phiếu khám)**

* id (PK, INT, auto-increment)
* record\_id (FK -> MedicalRecord.record\_id)
* service\_id (FK -> Service.service\_id)
* quantity (INT, default 1)
* unit\_price (DECIMAL) — lưu lại giá tại thời điểm khám

1. **Invoice (Hóa đơn)**

* invoice\_id (PK, INT, auto-increment)
* appointment\_id (FK -> Appointment.appointment\_id)
* patient\_id (FK -> Patient.patient\_id)
* total\_amount (DECIMAL)
* paid\_amount (DECIMAL)
* payment\_method (ENUM: 'Tiền mặt','Chuyển khoản','Khác')
* issued\_by (FK -> Staff.staff\_id)
* issued\_at (DATETIME)

1. **NotificationLog (Nhắc lịch - Log gửi SMS/Zalo)**

* notification\_id (PK, INT, auto-increment)
* appointment\_id (FK -> Appointment.appointment\_id)
* channel (ENUM: 'SMS','Zalo')
* status (ENUM: 'Pending','Sent','Failed')
* scheduled\_send\_at (DATETIME)
* sent\_at (DATETIME, nullable)
* response\_message (TEXT, nullable)

### 3.4.2. Mối quan hệ & Khóa ngoại (Relationships & Foreign Keys)

Liệt kê các mối quan hệ chính và các khoá ngoại tương ứng:

* Patient (1) - (N) Appointment
  + Appointment.patient\_id FK -> Patient.patient\_id
  + Ý nghĩa: Một bệnh nhân có thể có nhiều lịch hẹn; mỗi lịch hẹn thuộc về đúng một bệnh nhân.
* Staff (1) - (N) Appointment
  + Appointment.staff\_id FK -> Staff.staff\_id
  + Ý nghĩa: Một nha sĩ có nhiều lịch hẹn; mỗi lịch hẹn gán cho một nha sĩ.
* Staff (1) - (N) MedicalRecord
  + MedicalRecord.staff\_id FK -> Staff.staff\_id
  + Ý nghĩa: Một nha sĩ có thể tạo nhiều phiếu khám.
* Appointment (1) - (0..1) MedicalRecord
  + MedicalRecord.appointment\_id FK -> Appointment.appointment\_id (UNIQUE)
  + Ý nghĩa: Mỗi lịch hẹn có thể có tối đa 1 phiếu khám (chỉ khi bệnh nhân được khám tạo phiếu).
* MedicalRecord (1) - (N) MedicalRecordService
  + MedicalRecordService.record\_id FK -> MedicalRecord.record\_id
  + Ý nghĩa: Một phiếu khám có thể chứa nhiều dịch vụ.
* Service (1) - (N) MedicalRecordService
  + MedicalRecordService.service\_id FK -> Service.service\_id
  + Ý nghĩa: Một dịch vụ có thể xuất hiện trong nhiều phiếu khám.
* Appointment (1) - (0..1) CheckinQueue
  + CheckinQueue.appointment\_id FK -> Appointment.appointment\_id
  + Ý nghĩa: Một lịch hẹn có thể được check-in (tạo bản ghi hàng đợi) hoặc chưa check-in.
* Appointment (1) - (0..1) Invoice
  + Invoice.appointment\_id FK -> Appointment.appointment\_id
  + Ý nghĩa: Một lịch hẹn có thể phát sinh hóa đơn khi thanh toán.
* Patient (1) - (N) Invoice
  + Invoice.patient\_id FK -> Patient.patient\_id
  + Ý nghĩa: Một bệnh nhân có 1 hoặc nhiều hóa đơn
* Appointment (1) - (N) NotificationLog
  + NotificationLog.appointment\_id FK -> Appointment.appointment\_id
  + Ý nghĩa: Các lời nhắc có thể gửi nhiều lần (ví dụ nhắc 24h, nhắc 1h).

### 3.4.3 Cardinality & Optionality (Cách hiểu bằng phát biểu và ký hiệu)

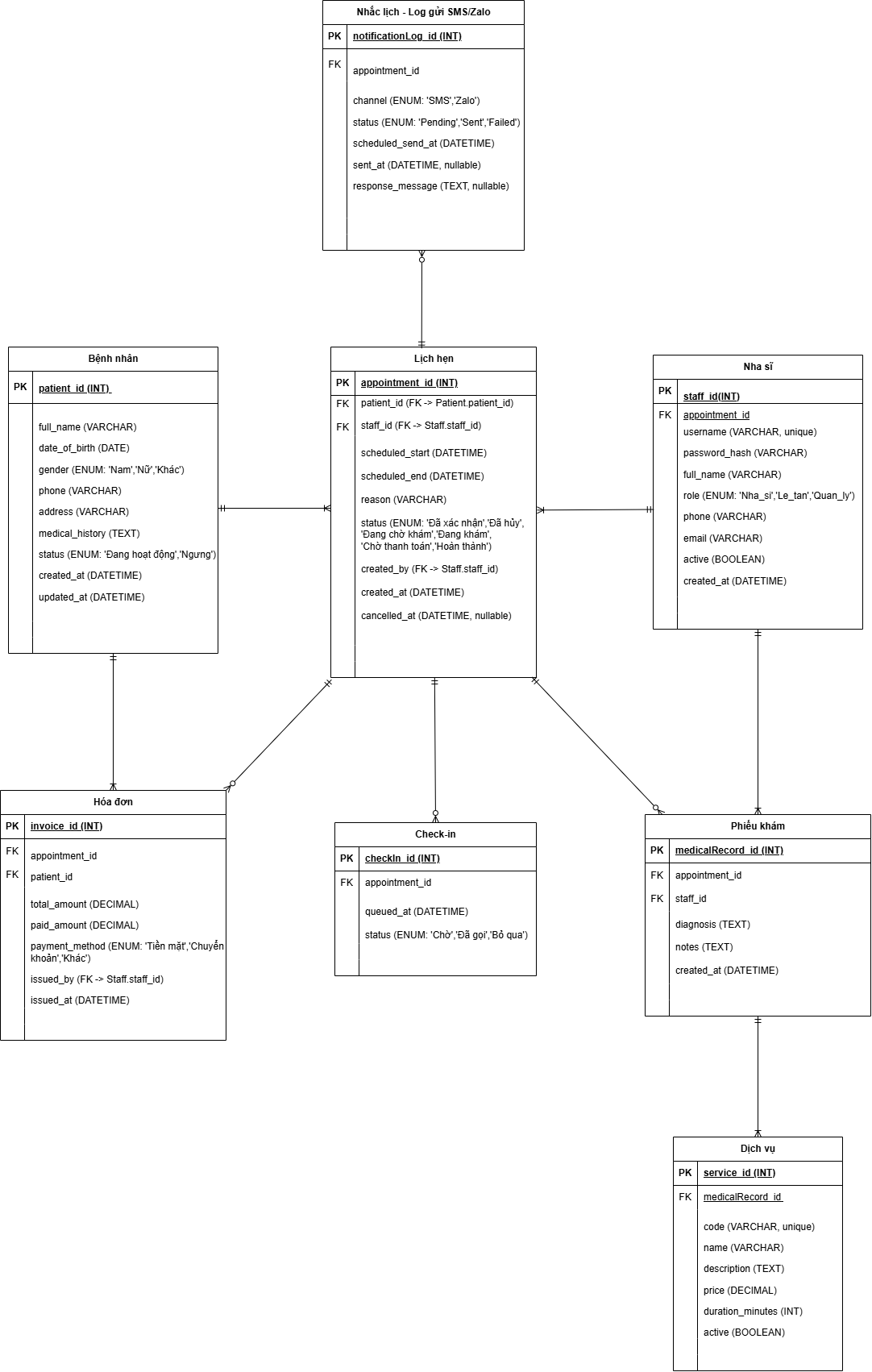
Dưới đây dùng ký hiệu (min..max):

* **Patient - Appointment:** (1..1) Patient -> (0..N) Appointment
  + Một Patient bắt buộc tồn tại (PK), có thể có 0 hoặc nhiều Appointment.
  + Optionality: Appointment.patient\_id **required** (không null) — mỗi lịch hẹn phải gắn bệnh nhân.
* **Staff - Appointment (Nha sĩ):** (1..1) Staff -> (0..N) Appointment
  + appointment.staff\_id **required** — mỗi lịch hẹn phải gán nha sĩ.
* **Appointment - MedicalRecord:** Appointment (1..1) -> MedicalRecord (0..1)
  + Một Appointment có thể có một MedicalRecord hoặc chưa có (trước khi khám). medicalrecord.appointment\_id **unique** và **nullable until created**.
* **MedicalRecord - MedicalRecordService:** (1..1) MedicalRecord -> (1..N) MedicalRecordService
  + Một phiếu khám nếu tồn tại thì thường có ít nhất 1 dịch vụ (nếu muốn cho phép 0 dịch vụ thì chỉnh thành 0..N).
* **Service - MedicalRecordService:** (1..1) Service -> (0..N) MedicalRecordService
  + MedicalRecordService.service\_id **required**.
* **Appointment - CheckinQueue:** Appointment (1..1) -> CheckinQueue (0..1)
  + CheckinQueue.appointment\_id **nullable** (chỉ tạo khi check-in).
* **Appointment - Invoice:** Appointment (1..1) -> Invoice (0..1..N)
  + Nếu hệ thống cho phép nhiều hóa đơn cho một lịch hẹn (ví dụ thanh toán chia nhỏ), thì là 0..N; nếu 1 lịch hẹn chỉ 1 hóa đơn, thì 0..1.
* **NotificationLog:** Một Appointment có thể có nhiều NotificationLog (0..N).

**Ghi chú về ràng buộc nghiệp vụ (business constraints):**

* Một khung giờ của một nha sĩ chỉ có thể có một Appointment ở trạng thái 'Đã xác nhận'. (Yêu cầu kiểm tra tại tầng ứng dụng hoặc DB unique index trên (staff\_id, scheduled\_start) khi trạng thái = 'Đã xác nhận').
* Appointment intervals should be non-overlapping per staff for confirmed appointments.

**Sơ đồ ERD**



**Chuẩn hóa**

* **1NF:** Tất cả thuộc tính có giá trị nguyên tố (không lặp, không mảng).
* **2NF:** Mỗi bảng có khóa chính duy nhất, các thuộc tính phụ thuộc hoàn toàn vào khóa chính.
* **3NF:** Không có phụ thuộc bắc cầu (VD: Tổng tiền không lưu thẳng trong TreatmentDetail mà tính từ Price \* Quantity).

-> Mô hình đã đạt **chuẩn 3NF**, đảm bảo dữ liệu không dư thừa và dễ bảo trì.

# PHỤ LỤC (APPENDIX)

## *A* Biểu đồ Lớp (Class Diagram - Domain Model)

Mô hình này mô tả các thực thể nghiệp vụ cốt lõi và mối quan hệ logic giữa chúng từ góc nhìn hướng đối tượng, làm nền tảng cho việc thiết kế Tầng Nghiệp vụ.

**Các lớp và thuộc tính**

|  |  |  |
| --- | --- | --- |
| **Lớp** | **Thuộc tính chính** | **Phương thức (method)** |
| **Patient** | patientID, fullName, dob, gender, phone, address, medicalHistory, status | addPatient(), updatePatient(), searchPatient() |
| **Dentist** | dentistID, fullName, specialty, phone, status | assignAppointment(), viewSchedule() |
| **Receptionist** | receptionistID, fullName, username, password | createAppointment(), cancelAppointment(), checkInPatient(), processPayment() |
| **Service** | serviceID, serviceName, price, description | updatePrice(), addService(), removeService() |
| **Appointment** | appointmentID, patient, dentist, date, time, reason, status | confirmAppointment(), cancelAppointment(), checkIn() |
| **TreatmentRecord** | recordID, appointment, dentist, diagnosis, notes, date | addServiceDetail(), saveRecord() |
| **TreatmentDetail** | detailID, service, quantity, subtotal | calculateSubtotal() |
| **Invoice** | invoiceID, treatmentRecord, receptionist, totalAmount, paymentDate, status | generateInvoice(), markAsPaid() |

**Mối quan hệ giữa các lớp (Multiplicity)**

|  |  |  |
| --- | --- | --- |
| **Quan hệ** | **Multiplicity** | **Giải thích** |
| Patient — Appointment | 1..\* | Một bệnh nhân có thể có nhiều lịch hẹn. |
| Dentist — Appointment | 1..\* | Một nha sĩ có nhiều lịch hẹn. |
| Appointment — TreatmentRecord | 1..1 | Mỗi lịch hẹn gắn với một phiếu khám. |
| TreatmentRecord — TreatmentDetail | 1..\* | Một phiếu khám có nhiều dịch vụ. |
| TreatmentRecord — Invoice | 0..1 | Có thể đã hoặc chưa thanh toán. |
| Receptionist — Invoice | 1..\* | Một lễ tân có thể lập nhiều hóa đơn. |
| Service — TreatmentDetail | 1..\* | Một dịch vụ có thể xuất hiện trong nhiều phiếu khám. |

## B Biểu đồ Class Diagram

