**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**





**BÁO CÁO:BÀI TẬP LỚN**

**HỌC PHẦN: PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN**

**Giảng viên: TS. Đào Ngọc Phong**

**Nhóm : 11**

| **Họ tên** | **MSV** |
| --- | --- |
| Nguyễn Hữu Văn (NT) | K23DTCN247 |
| Nguyễn Anh Văn | K23DTCN248 |
| Hồ Thân Chính | K23DTCN086 |
| Hà Dương Kim Tuyến | K23DTCN243 |

**Hà Nội, 2025**

[**I. CÔNG VIỆC CHUNG CỦA NHÓM 4**](#_wkl6ov2as162)

[1. Mô tả yêu cầu bài toán, yêu cầu người dùng 4](#_lic3mgcyolaq)

[a. Mô tả bằng ngôn ngữ tự nhiên 4](#_fh9cp7ze52zb)

[b. Mô tả đủ các yêu cầu mà "người dùng" cần 4](#_hu3wo46madss)

[c. Các ràng buộc nghiệp vụ 5](#_djdv9cngufl9)

[2. Mô tả yêu cầu phần mềm 6](#_13jtpbma3kt1)

[a. Phân tích và xác định actor 6](#_6s4vqb11vhvs)

[b. Xác định yêu cầu chức năng: các Usecase 6](#_m0hirbws3aap)

[c. Xác định các yêu cầu phi chức năng 8](#_qj11fgoi4lig)

[**3. Phân tích, thiết kế kiến trúc 9**](#_vhwh1ik5mdcx)

[a. Phân tích yêu cầu hệ thống: 10](#_lla6geviuezi)

[b. Lựa chọn kiến trúc: Microservices Architecture 10](#_m7l16bewx0cv)

[4. Xây dựng biểu đồ lớp thực thể 11](#_9v2rwkr0rmdw)

[4.1. Phân tích và xác định các thực thể 11](#_wm0w6ep1zv0g)

[4.2. Mô tả chi thiết các thực thể 12](#_ssdw3l6664l)

[**II. USECASE TỪNG THÀNH VIÊN 17**](#_dcx8uh5denlw)

[**Nguyễn Hữu Văn - Usecase tạo hoá đơn trả trước và trả sau 17**](#_4ahbqy6ejw0q)

[1. Biểu đồ Use case 17](#_4oz08iezj9wn)

[2. Viết đặc tả UC 17](#_h02wwqh42dvy)

[3. Biểu đồ lớp UC ở pha phân tích 20](#_jaxthqwk19p8)

[4. Biểu đồ lớp UC ở pha thiết kế 24](#_jlwi5n5l6cx5)

[**Hồ Thân Chính - Quản lý gói đăng ký với khách hàng 25**](#_clw3uyudz1k6)

[1. Biểu đồ Use case 25](#_w1c5k32zthrj)

[2. Viết đặc tả UC 25](#_n0zzy75xc53v)

[2.1. Use case đăng ký dịch vụ 25](#_o7n7eu55slp4)

[2.2. Use case thay đổi gói dịch vụ 28](#_24a9mx3xznkh)

[2.3. Use case hủy gói dịch vụ 32](#_duuvplh7e1li)

[3. Biểu đồ lớp UC ở pha phân tích 34](#_80j7vn3xt8o0)

[4. Biểu đồ lớp UC ở pha thiết kế 36](#_1z5t4yht4lar)

[**Hà Dương Kim Tuyến - Usecase Quản lý thông tin công ty 39**](#_cghc1dsbcg5r)

[1. Biểu đồ Use case 39](#_p45b4xcpw41l)

[2. Viết đặc tả UC 39](#_mxhb629vclpg)

[3. Biểu đồ lớp UC ở pha phân tích 41](#_y3gwpi3z8ws5)

[4. Biểu đồ lớp UC ở pha thiết kế 43](#_wg2gzl5hhdhl)

[**Nguyễn Anh Văn - Usecase Tạo báo cáo 45**](#_i5gr0wp8cdy)

[1. Biểu đồ Use case 45](#_313lnxygy42c)

[2. Viết đặc tả UC 45](#_nls2wn5hxdw0)

[3. Biểu đồ lớp UC ở pha phân tích 49](#_fk189b9xxdlr)

[4. Biểu đồ lớp UC ở pha thiết kế 51](#_4le6jswaawn)

# **I. CÔNG VIỆC CHUNG CỦA NHÓM**

## 1. Mô tả yêu cầu bài toán, yêu cầu người dùng

### a. Mô tả bằng ngôn ngữ tự nhiên

Trong bối cảnh hiện đại, việc chăm sóc sức khỏe nhân viên đang trở thành ưu tiên quan trọng của các doanh nghiệp. Nhiều công ty muốn đầu tư vào dịch vụ gym cho nhân viên nhằm nâng cao sức khỏe, cải thiện tinh thần làm việc và tăng năng suất lao động. Tuy nhiên, việc quản lý và thanh toán dịch vụ gym theo mô hình truyền thống gặp nhiều khó khăn.

Hệ thống quản lý phòng gym B2B được xây dựng để giải quyết bài toán này. Thay vì mỗi nhân viên tự đăng ký và thanh toán riêng lẻ, các công ty sẽ mua gói dịch vụ tổng thể cho toàn bộ nhân viên. Hệ thống áp dụng mô hình thanh toán kết hợp: công ty trả trước một khoản credit dựa trên gói đăng ký, sau đó nhân viên sử dụng dịch vụ sẽ được trừ vào credit này. Nếu sử dụng vượt mức credit, công ty sẽ thanh toán thêm phần chênh lệch.

### b. Mô tả đủ các yêu cầu mà "người dùng" cần

**Đối với công ty khách hàng:**

* Cần có thể đăng ký gói dịch vụ phù hợp với số lượng nhân viên của mình
* Muốn quản lý danh sách nhân viên được phép sử dụng dịch vụ
* Cần theo dõi chi tiết việc sử dụng của từng nhân viên và tổng chi phí
* Muốn có thể thay đổi gói dịch vụ theo nhu cầu hàng tháng
* Cần báo cáo chi tiết về việc sử dụng và chi phí để kiểm soát ngân sách

**Đối với nhân viên các công ty:**

* Cần quy trình check-in/check-out đơn giản và nhanh chóng
* Muốn có thể chọn loại phòng tập phù hợp với nhu cầu
* Cần xem được lịch sử sử dụng cá nhân
* Muốn biết thông tin về gói dịch vụ công ty đã đăng ký

**Đối với phòng sales của gym:**

* Cần quản lý thông tin các công ty khách hàng
* Muốn theo dõi việc sử dụng dịch vụ của từng công ty
* Cần hỗ trợ công ty thay đổi hoặc hủy gói dịch vụ
* Muốn có báo cáo doanh số và hiệu quả kinh doanh

**Đối với phòng kế toán của gym:**

* Cần tự động hóa việc tạo hóa đơn hàng tháng
* Muốn theo dõi tình trạng thanh toán của các công ty
* Cần báo cáo tài chính chi tiết và chính xác
* Muốn quản lý công nợ và nhắc nhở thanh toán

### c. Các ràng buộc nghiệp vụ

**Ràng buộc về gói dịch vụ:**

* Hệ thống cung cấp 4 gói cố định: A ($500, 5 người), B ($1000, 8 người), C ($1500, 10 người), D ($2000, 12 người)
* Số lượng nhân viên của công ty không được vượt quá giới hạn của gói đã chọn
* Nếu công ty có nhiều nhân viên hơn gói hiện tại, bắt buộc phải nâng cấp gói

**Ràng buộc về thời gian:**

* Thay đổi gói chỉ được phép thực hiện vào ngày đầu tiên của tháng
* Không cho phép thay đổi hoặc hủy gói giữa tháng
* Thời gian sử dụng được tính tròn lên theo giờ

**Ràng buộc về sử dụng dịch vụ:**

* Nhân viên không bị giới hạn thời gian sử dụng trong ngày
* Có 3 hạng phòng với giá khác nhau: Silver ($10/giờ), Gold ($15/giờ), Platinum ($20/giờ)
* Mỗi lần sử dụng phải có đầy đủ thông tin check-in và check-out

**Ràng buộc về thanh toán:**

* Mỗi tháng công ty phải thanh toán 2 loại hóa đơn: credit trả trước và phí vượt (nếu có)
* Credit của tháng trước không được chuyển sang tháng sau
* Khi hủy gói, công ty vẫn phải thanh toán phí vượt tháng hiện tại nhưng không cần trả credit tháng sau

## 2. Mô tả yêu cầu phần mềm

### a. Phân tích và xác định actor

**Primary Actors (Tác nhân chính):**

1. **Company Admin (Quản trị viên công ty)**
   * Vai trò: Đại diện công ty trong việc quản lý gói dịch vụ và nhân viên
   * Quyền hạn: Đăng ký/cập nhật thông tin công ty, quản lý danh sách nhân viên, xem báo cáo sử dụng
2. **Employee (Nhân viên)**
   * Vai trò: Người sử dụng trực tiếp dịch vụ gym
   * Quyền hạn: Check-in/check-out, chọn hạng phòng, xem lịch sử sử dụng cá nhân
3. **Sales Staff (Nhân viên kinh doanh)**
   * Vai trò: Quản lý các gói đăng ký và quan hệ khách hàng doanh nghiệp
   * Quyền hạn: Tạo/thay đổi/hủy gói đăng ký, theo dõi sử dụng, tạo báo cáo doanh số
4. **Finance Staff (Nhân viên kế toán)**
   * Vai trò: Xử lý các vấn đề tài chính và thanh toán
   * Quyền hạn: Tạo hóa đơn, theo dõi thanh toán, quản lý công nợ, báo cáo tài chính
5. **Finance Manager (Kế toán trưởng)**
   * Vai trò: Xử lý các vấn đề tài chính cấp quản lý
   * Quyền hạn: Duyệt hóa đơn tạo bởi kế toán

**Secondary Actors (Tác nhân phụ):**

1. **System Admin (Quản trị viên hệ thống)**
   * Vai trò: Vận hành và bảo trì hệ thống
   * Quyền hạn: Cấu hình hệ thống, backup dữ liệu, monitor performance
2. **Payment Gateway (Cổng thanh toán)**
   * Vai trò: Xử lý giao dịch thanh toán trực tuyến
   * Tương tác: Nhận yêu cầu thanh toán, trả về kết quả giao dịch

### b. Xác định yêu cầu chức năng: các Usecase

**UC-01: Quản lý thông tin công ty**

* Actor: Company Admin, Sales Staff
* Mô tả: Đăng ký, cập nhật thông tin công ty khách hàng
* Luồng chính: Nhập thông tin → Xác thực → Lưu vào hệ thống

**UC-02: Quản lý danh sách nhân viên**

* Actor: Company Admin
* Mô tả: Thêm, sửa, xóa nhân viên trong danh sách được phép sử dụng
* Luồng chính: Chọn thao tác → Nhập thông tin nhân viên → Kiểm tra ràng buộc → Cập nhật

**UC-03: Đăng ký gói dịch vụ**

* Actor: Sales Staff
* Mô tả: Tạo gói đăng ký mới cho công ty
* Luồng chính: Chọn công ty → Kiểm tra số lượng nhân viên → Chọn gói phù hợp → Tạo subscription

**UC-04: Thay đổi gói dịch vụ**

* Actor: Sales Staff
* Mô tả: Thay đổi gói đăng ký vào đầu tháng
* Luồng chính: Kiểm tra thời gian → Chọn gói mới → Cập nhật subscription → Điều chỉnh billing

**UC-05: Hủy gói dịch vụ**

* Actor: Sales Staff
* Mô tả: Hủy gói đăng ký của công ty
* Luồng chính: Xác nhận hủy → Tính toán phí vượt → Tạo hóa đơn cuối → Vô hiệu hóa gói

**UC-06: Check-in vào gym**

* Actor: Employee
* Mô tả: Nhân viên check-in để bắt đầu sử dụng dịch vụ
* Luồng chính: Xác thực nhân viên → Chọn hạng phòng → Ghi nhận thời gian bắt đầu

**UC-07: Check-out khỏi gym**

* Actor: Employee
* Mô tả: Nhân viên check-out khi kết thúc sử dụng
* Luồng chính: Ghi nhận thời gian kết thúc → Tính thời gian sử dụng → Tính chi phí → Lưu session

**UC-08: Xem lịch sử sử dụng**

* Actor: Employee, Company Admin
* Mô tả: Xem chi tiết lịch sử sử dụng dịch vụ
* Luồng chính: Chọn khoảng thời gian → Lấy dữ liệu → Hiển thị báo cáo

**UC-09: Tạo hóa đơn credit**

* Actor: Accounting Staff, System
* Mô tả: Tạo hóa đơn trả trước đầu mỗi tháng
* Luồng chính: Lấy thông tin gói → Tính credit → Tạo hóa đơn → Gửi thông báo

**UC-10: Tạo hóa đơn phí vượt**

* Actor: Accounting Staff, System
* Mô tả: Tạo hóa đơn cho phần sử dụng vượt credit
* Luồng chính: Tính tổng usage → So sánh với credit → Tạo hóa đơn phí vượt → Gửi thông báo

**UC-11: Xử lý thanh toán**

* Actor: Accounting Staff, Payment Gateway
* Mô tả: Xử lý thanh toán hóa đơn
* Luồng chính: Nhận thanh toán → Xác thực giao dịch → Cập nhật trạng thái → Gửi xác nhận

**UC-12: Tạo báo cáo**

* Actor: Company Admin, Sales Staff, Accounting Staff
* Mô tả: Tạo các loại báo cáo theo yêu cầu
* Luồng chính: Chọn loại báo cáo → Thiết lập tham số → Tạo báo cáo → Xuất file

### c. Xác định các yêu cầu phi chức năng

**Performance (Hiệu năng):**

* Thời gian check-in/check-out: < 2 giây
* Thời gian tải báo cáo: < 5 giây
* Hệ thống phải xử lý đồng thời 1000+ check-in/check-out
* Hỗ trợ 10,000+ nhân viên active cùng lúc

**Reliability (Độ tin cậy):**

* Uptime: 99.9% (tối đa 8.76 giờ downtime/năm)
* Độ chính xác tính toán thời gian: 100%
* Độ chính xác tính toán hóa đơn: 99.99%
* Backup dữ liệu tự động hàng ngày

**Security (Bảo mật):**

* Xác thực 2 lớp (2FA) cho admin
* Mã hóa dữ liệu nhạy cảm (thông tin tài chính)
* Phân quyền rõ ràng theo vai trò
* Audit trail cho tất cả thao tác quan trọng
* Session timeout sau 30 phút không hoạt động

**Scalability (Khả năng mở rộng):**

* Kiến trúc microservices cho phép scale từng module
* Hỗ trợ horizontal và vertical scaling
* Database partitioning theo công ty
* Load balancing và auto-scaling

**Usability (Khả năng sử dụng):**

* Giao diện trực quan, dễ sử dụng
* Responsive design cho mobile và desktop
* Hỗ trợ đa ngôn ngữ (Tiếng Việt, English)
* Tuân thủ chuẩn accessibility WCAG 2.1

**Maintainability (Khả năng bảo trì):**

* Code coverage > 80%
* Comprehensive documentation
* Real-time monitoring và alerting
* Centralized logging system

## 3. Phân tích, thiết kế kiến trúc

Phân tích các yếu tố để lựa chọn mô hình kiến trúc triển khai

### a. Phân tích yêu cầu hệ thống:

**1. Tính phức tạp của nghiệp vụ:**

* Hệ thống có nhiều module độc lập: quản lý người dùng, quản lý gói, thanh toán, báo cáo
* Mỗi module có logic nghiệp vụ riêng biệt và có thể phát triển độc lập
* Cần tích hợp với các hệ thống bên ngoài (payment gateway, notification service)

**2. Yêu cầu về performance và scalability:**

* Cần xử lý đồng thời nhiều check-in/check-out (1000+ concurrent users)
* Khối lượng dữ liệu lớn với nhiều công ty và nhân viên (10,000+ active users)
* Tần suất truy cập không đều (peak hours vào buổi sáng và chiều)

**3. Yêu cầu về reliability và availability:**

* Uptime cao (99.9%) đòi hỏi khả năng fault tolerance
* Dữ liệu giao dịch quan trọng không được mất
* Cần khả năng recovery nhanh chóng khi có sự cố

**4. Yêu cầu về maintenance và deployment:**

* Cần deploy các module độc lập để giảm thiểu downtime
* Team phát triển có thể làm việc song song trên các module khác nhau
* Cần monitor và debug từng module riêng biệt

**5. Yêu cầu về security:**

* Dữ liệu tài chính nhạy cảm cần bảo mật cao
* Phân quyền phức tạp với nhiều vai trò người dùng
* Audit trail cho tất cả giao dịch quan trọng

### b. Lựa chọn kiến trúc: Microservices Architecture

Dựa trên phân tích trên, nhóm chọn **Microservices Architecture** với các lý do sau:

**Ưu điểm phù hợp:**

* **Scalability**: Có thể scale từng service độc lập theo nhu cầu
* **Technology diversity**: Mỗi service có thể sử dụng công nghệ phù hợp nhất
* **Team autonomy**: Các team có thể phát triển và deploy độc lập
* **Fault isolation**: Lỗi ở một service không ảnh hưởng toàn bộ hệ thống
* **Maintainability**: Dễ dàng maintain và update từng service

**Nhược điểm cần xử lý:**

* **Complexity**: Cần quản lý nhiều service và communication giữa chúng
* **Data consistency**: Cần implement distributed transaction patterns
* **Network latency**: Cần optimize inter-service communication
* **Monitoring**: Cần monitoring system phức tạp hơn

## 4. Xây dựng biểu đồ lớp thực thể

### 4.1. Phân tích và xác định các thực thể

Dựa trên mô hình kinh doanh B2B Gym Management System, hệ thống cần quản lý các thực thể sau:

a. Nhóm thực thể quản lý công ty và gói dịch vụ:

* **CongTy**: Đại diện cho các công ty khách hàng sử dụng dịch vụ gym B2B
* **GoiDangKy**: Gói dịch vụ mà công ty đăng ký (Package A, B, C, D với giới hạn số người và giá cụ thể)
* **LichSuThayDoiGoi**: Lịch sử thay đổi gói đăng ký của công ty
* **HuyDangKy**: Thông tin về việc hủy đăng ký gói dịch vụ

b. Nhóm thực thể quản lý hóa đơn:

* **HoaDonTraTruoc**: Hóa đơn thanh toán trước credit cho tháng tiếp theo
* **ChiTietTraTruoc**: Chi tiết các line item trong hóa đơn trả trước (phí gói, giảm giá, phụ phí)
* **NoteGiamGia**: Ghi chú giảm giá/phụ phí từ Sales staff

c. Nhóm thực thể quản lý nhân viên và người dùng:

* **NhanVien**: Nhân viên hệ thống bao gồm:
  + Kế toán: Tạo hóa đơn
  + Kế toán trưởng: Phê duyệt hóa đơn
  + Sales Staff: Quản lý gói đăng ký, tạo voucher/note giảm giá
  + Company Admin: Quản lý thông tin công ty
* **TaiKhoan**: Tài khoản đăng nhập của người dùng trong hệ thống

d. Nhóm thực thể lịch sử và audit:

* **LichSuCapNhat**: Lịch sử cập nhật các thông tin quan trọng (credit balance, trạng thái hóa đơn, thay đổi gói)

### 4.2. Mô tả chi thiết các thực thể

**4.2.1 CongTy (Company)**

**Mục đích**: Lưu trữ thông tin các công ty khách hàng

**Thuộc tính:**

* maCongTy: UUID - Mã định danh công ty (Primary Key)
* tenCongTy: String - Tên công ty
* email: String - Email liên hệ
* soDuCredit: BigDecimal - Số dư credit hiện tại
* trangThai: String - Trạng thái (HOAT\_DONG, KHONG\_HOAT\_DONG, TAM\_DUNG, DUNG\_HOAT\_DONG)

**Quan hệ:**

* 1 CongTy có nhiều GoiDangKy (1:N)
* 1 CongTy có nhiều HoaDonTraTruoc (1:N)
* 1 CongTy có nhiều NoteGiamGia (1:N)

**4.2.2 Subscription (GoiDangKy)**

**Mục đích:** Lưu thông tin gói dịch vụ mà công ty đăng ký

**Thuộc tính:**

* maGoi: UUID - Mã gói đăng ký (Primary Key)
* tenGoi: String - Tên gói (Package A/B/C/D)
* giaGoi: BigDecimal - Giá gói theo tháng
* soNVMax: Integer - Số nhân viên tối đa
* trangThai: String - Trạng thái gói (HOAT\_DONG, HET\_HAN, DA\_HUY, CHO\_XU\_LY)
* thangNam: String - Tháng năm áp dụng gói
* maCongTy: UUID - Mã công ty (Foreign Key)

**Quan hệ:**

* N GoiDangKy thuộc về 1 CongTy (N:1)
* 1 GoiDangKy có nhiều LichSuThayDoiGoi (1:N)

**4.2.3. HoaDonTraTruoc (Prepaid Invoice)**

**Mục đích:** Lưu thông tin hóa đơn trả trước credit

**Thuộc tính:**

* maHoaDonTraTruoc: UUID - Mã hóa đơn (Primary Key)
* maCongTy: UUID - Mã công ty (Foreign Key)
* tongTien: BigDecimal - Tổng tiền hóa đơn
* trangThai: String - Trạng thái (NHAP, CHO\_DUYET, DA\_DUYET, BI\_TU\_CHOI)
* maNguoiTao: UUID - Mã nhân viên tạo (Foreign Key)
* maNguoiDuyet: UUID - Mã nhân viên duyệt (Foreign Key)

**Quan hệ:**

* N HoaDonTraTruoc thuộc về 1 CongTy (N:1)
* 1 HoaDonTraTruoc có nhiều ChiTietTraTruoc (1:N)
* N HoaDonTraTruoc được tạo bởi 1 NhanVien (N:1)
* N HoaDonTraTruoc được duyệt bởi 1 NhanVien (N:1)

**4.2.4. ChiTietTraTruoc (Prepaid Line Item)**

**Mục đích:** Lưu các line item trong hóa đơn trả trước

**Thuộc tính:**

* maChiTietTT: UUID - Mã chi tiết (Primary Key)
* maHoaDonTraTruoc: UUID - Mã hóa đơn (Foreign Key)
* moTa: String - Mô tả line item
* soTien: BigDecimal - Số tiền (có thể âm nếu là giảm giá)

**Quan hệ:**

* N ChiTietTraTruoc thuộc về 1 HoaDonTraTruoc (N:1)

**4.2.5. NoteGiamGia (Sales Note)**

**Mục đích:** Lưu ghi chú giảm giá/phụ phí từ Sales

**Thuộc tính:**

* maNote: UUID - Mã note (Primary Key)
* noiDung: String - Nội dung ghi chú

**Quan hệ:**

* N NoteGiamGia thuộc về 1 CongTy (quan hệ gián tiếp)

**4.2.6. NhanVien (Employee)**

**Mục đích:** Lưu thông tin nhân viên hệ thống

**Thuộc tính:**

* maNhanVien: UUID - Mã nhân viên (Primary Key)
* hoTen: String - Họ tên
* email: String - Email
* viTri: String - Vị trí (KE\_TOAN, KE\_TOAN\_TRUONG, NHAN\_VIEN\_BAN\_HANG, QUAN\_TRI)

**Quan hệ:**

* 1 NhanVien tạo nhiều HoaDonTraTruoc (1:N)
* 1 NhanVien duyệt nhiều HoaDonTraTruoc (1:N)
* 1 NhanVien có 1 TaiKhoan (1:1)

**4.2.7. TaiKhoan (Account)**

**Mục đích:** Lưu thông tin tài khoản đăng nhập

**Thuộc tính:**

* maTaiKhoan: UUID - Mã tài khoản (Primary Key)
* tenTaiKhoan: String - Username
* matKhau: String - Password (đã mã hóa)
* ngayTao: DateTime - Ngày tạo tài khoản

**Quan hệ:**

* 1 TaiKhoan thuộc về 1 NhanVien (1:1)

**4.2.8. LichSuThayDoiGoi (Subscription Change History)**

**Mục đích:** Lưu lịch sử thay đổi gói đăng ký

**Thuộc tính:**

* maChange: UUID - Mã lịch sử (Primary Key)
* maSubcription: UUID - Mã gói đăng ký (Foreign Key)
* ngayThayDoi: DateTime - Ngày thay đổi
* loaiThayDoi: Enum - Loại thay đổi
* maGoiCu: UUID - Mã gói cũ
* maGoiMoi: UUID - Mã gói mới
* ghiChu: String - Ghi chú

**Quan hệ:**

* N LichSuThayDoiGoi thuộc về 1 GoiDangKy (N:1)

**4.2.9. HuyDangKy (Cancellation)**

**Mục đích:** Lưu thông tin hủy đăng ký

**Thuộc tính:**

* maHuy: UUID - Mã hủy (Primary Key)
* maSubcription: UUID - Mã gói đăng ký (Foreign Key)
* ngayYeuCau: DateTime - Ngày yêu cầu hủy
* lyDo: String - Lý do hủy
* trangThai: Enum - Trạng thái (CHO\_XU\_LY, DA\_DUYET, BI\_TU\_CHOI)
* ngayXuLy: DateTime - Ngày xử lý
* maNhanVienXuLy: UUID - Mã nhân viên xử lý

**Quan hệ:**

* N HuyDangKy thuộc về 1 GoiDangKy (N:1)

**4.2.10. LichSuCapNhat (Update History)**

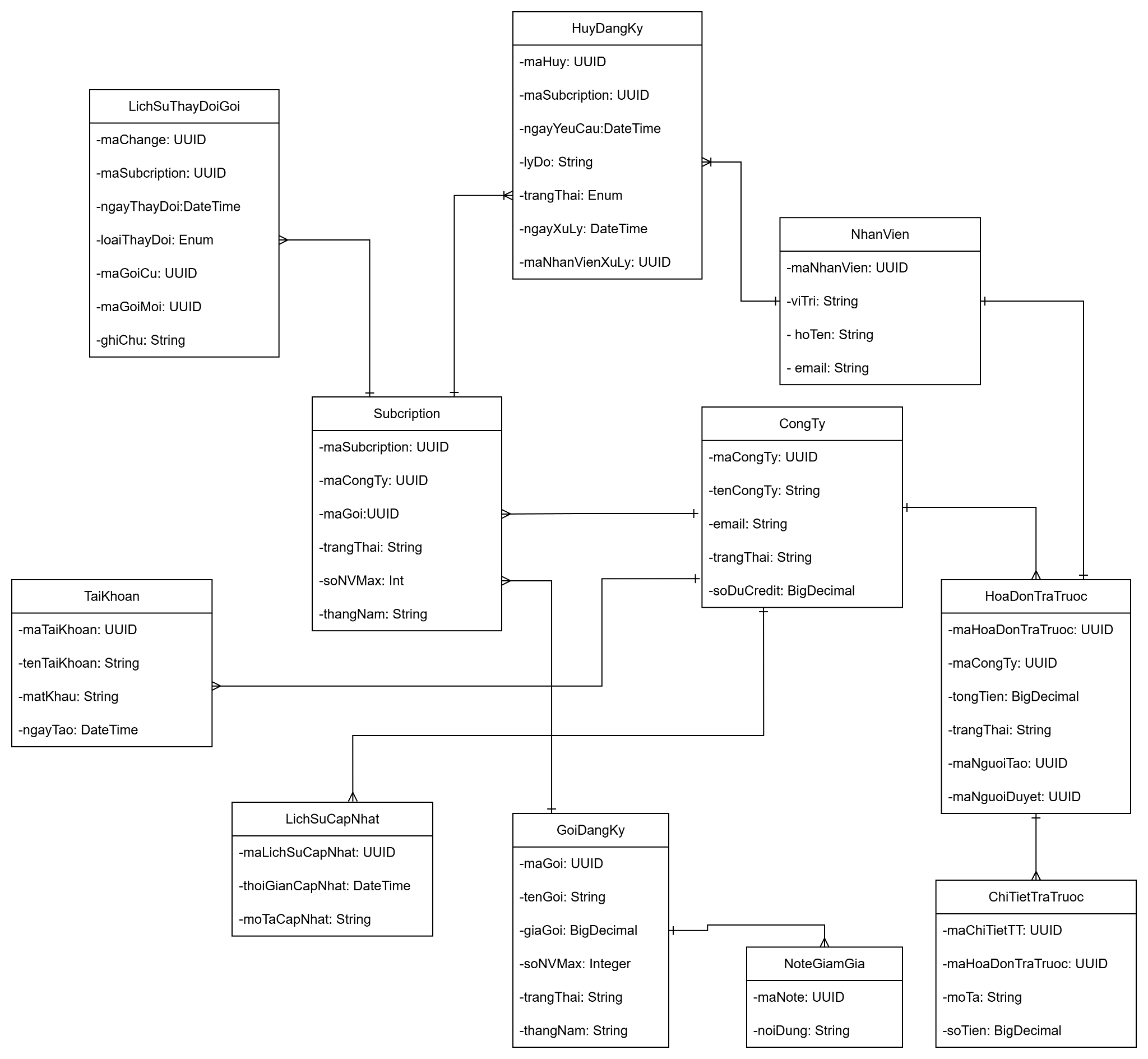
**Mục đích:** Lưu lịch sử cập nhật các thay đổi quan trọng

**Thuộc tính:**

* maLichSuCapNhat: UUID - Mã lịch sử (Primary Key)
* thoiGianCapNhat: DateTime - Thời gian cập nhật
* moTaCapNhat: String - Mô tả cập nhật

**Quan hệ:**

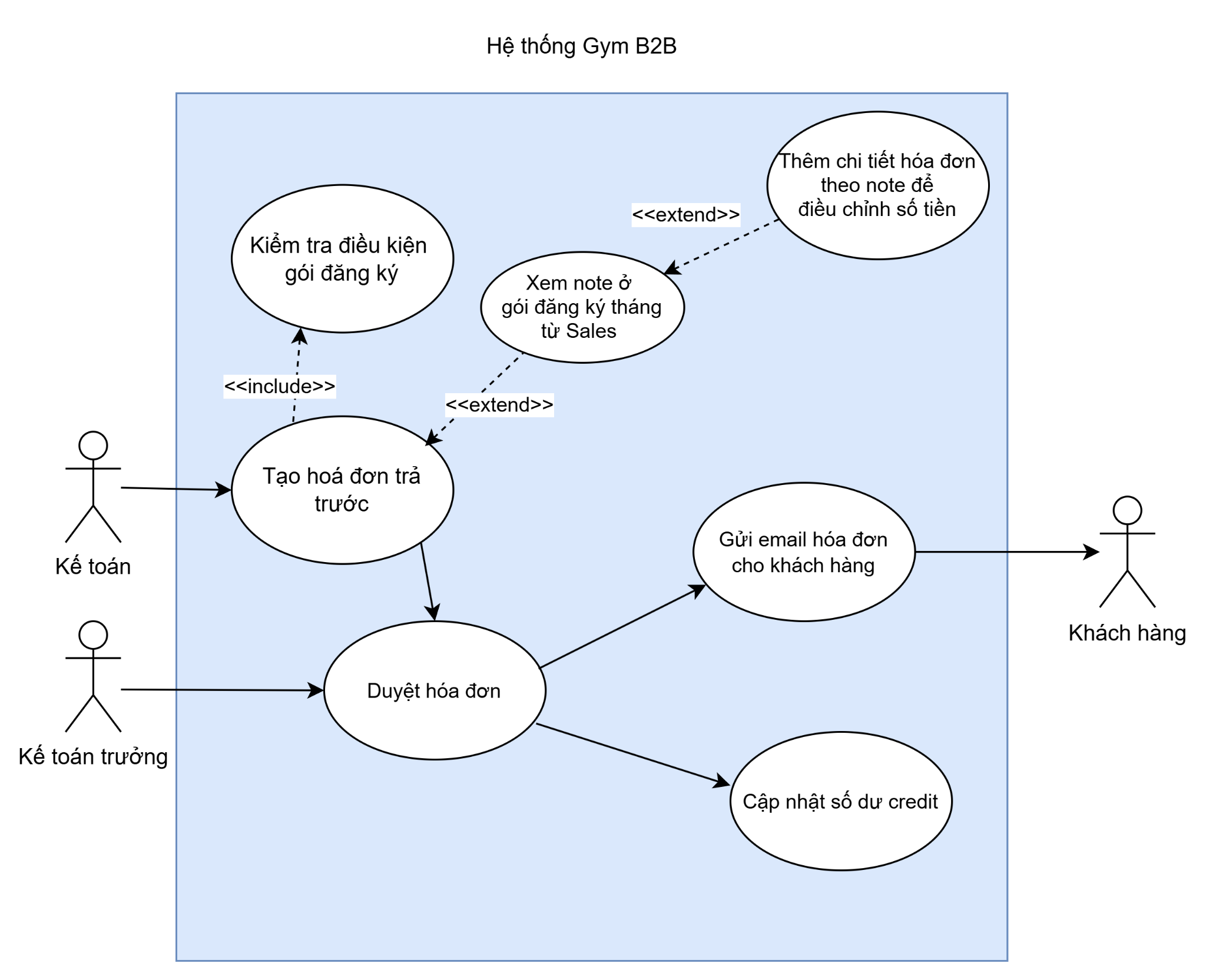
* Có thể liên kết với nhiều thực thể khác nhau



# **II. USECASE TỪNG THÀNH VIÊN**

# **Nguyễn Hữu Văn - Usecase tạo hoá đơn trả trước và trả sau**

## 1. Biểu đồ Use case



## 2. Viết đặc tả UC

| Use Case Specification | | | |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-PREPAID-INVOICE | | |
| **Use Case Name:** | Tạo hoá đơn trả trước | | |
| **Created By:** | Nguyễn Hữu Văn | | |
| **Primary Actor:** | Kế toán | **Secondary Actors:** | Kế toán trưởng |
| **Trigger:** | - Đã đến thời điểm tạo hóa đơn (từ ngày 25 tháng trước đến ngày 5 tháng hiện tại) - Có ít nhất 1 công ty có gói đăng ký active | | |
| **Description:** | Là kế toán của hệ thống quản lý phòng gym, tới cuối tháng, tôi cần tạo hoá đơn trả trước cho phí dịch vụ tháng sau của người dùng | | |
| **Preconditions:** | - Kế toán đã đăng nhập vào hệ thống  - Đã đến chu kỳ tạo hóa đơn (từ ngày 25 tháng trước đến ngày 5 tháng hiện tại)  - Công ty đã có thông tin đăng ký trong hệ thống | | |
| **Post–conditions:** | - **Thành công:**   * Hóa đơn được tạo với status "APPROVED" * Credit balance của công ty được cập nhật * Email hóa đơn được gửi cho công ty   **Thất bại:** Hệ thống ghi log lỗi, thông báo cho kế toán | | |
| **Normal Flow:** | 1. Kế toán khởi tạo quy trình tạo hóa đơn trả trước 2. Hệ thống hiển thị danh sách các công ty cần tạo hóa đơn 3. Kế toán chọn công ty cụ thể 4. Hệ thống kiểm tra điều kiện gói đăng ký tháng sau: - Truy xuất thông tin subscription của công ty - Phân loại trạng thái: "Có gói", "Chưa có gói", "Đã hủy" 5. Hệ thống hiển thị thông tin gói và giá cơ bản 6. Kế toán kiểm tra ghi chú giảm giá từ Sales (nếu có) 7. Kế toán thêm 1 line item giảm giá vào hoá đơn 8. Kế toán xác nhận tạo hóa đơn 9. Hệ thống tạo hóa đơn với status "SUBMITTED" 10. Hệ thống gửi notification cho Kế toán trưởng để duyệt 11. Kế toán trưởng xem chi tiết hóa đơn và approve 12. Hệ thống cập nhật status thành "APPROVED" 13. Hệ thống cập nhật credit balance của công ty 14. Hệ thống gửi email hóa đơn cho công ty | | |
| **Alternative Flows:** | 4a: Công ty chưa có gói đăng ký tháng sau   * 4a.1 Hệ thống phát hiện công ty chưa có gói tháng sau * 4a.2 Hệ thống thông báo cho kế toán * 4a.3 Kế toán liên hệ Sales để xác nhận * 4a.4 Use case tạm dừng cho đến khi có xác nhận   11a: Kế toán trưởng từ chối duyệt hóa đơn   * 11a.1 Kế toán trưởng từ chối với lý do cụ thể * 11a.2 Hệ thống cập nhật status thành "REJECTED" * 11a.3 Gửi notification cho kế toán với lý do từ chối * 11a.4 Kế toán chỉnh sửa và submit lại | | |
| **Exceptions:** | **E1: Lỗi hệ thống khi tạo hóa đơn**   * **9e.1** Hệ thống gặp lỗi technical * **9e.2** Rollback các thao tác đã thực hiện * **9e.3** Ghi log lỗi chi tiết * **9e.4** Thông báo lỗi cho kế toán * **9e.5** Use case kết thúc   **E2: Lỗi gửi email**   * **14e.1** Hệ thống không thể gửi email * **14e.2** Ghi log cảnh báo * **14e.3** Đánh dấu cần gửi lại sau * **14e.4** Use case vẫn được coi là thành công | | |
| **Priority:** | Cao | | |
| **Frequency of Use** | Hàng tháng (chu kỳ từ ngày 25-5) | | |
| **Business Rules:** | **Quy tắc thời gian:**   * Chỉ được tạo hóa đơn trả trước trong khoảng từ ngày 25 tháng trước đến ngày 5 tháng hiện tại * Hóa đơn được tạo cho gói đăng ký của tháng tiếp theo   **Quy tắc gói đăng ký:**   * Công ty phải có gói đăng ký hợp lệ cho tháng cần tạo hóa đơn * Nếu chưa có gói hoặc đã hủy gói, cần xác nhận với Sales trước khi tạo hóa đơn   **Quy tắc phê duyệt:**   * Tất cả hóa đơn trả trước phải được Kế toán trưởng phê duyệt * Không được cập nhật credit balance khi chưa có phê duyệt * Hóa đơn bị từ chối có thể chỉnh sửa và submit lại   **Quy tắc bảo mật:**   * Chỉ Kế toán mới có quyền tạo hóa đơn trả trước * Chỉ Kế toán trưởng mới có quyền phê duyệt   **Quy tắc nghiệp vụ:**   * Một công ty chỉ có thể có một hóa đơn trả trước cho một tháng cụ thể * Không được tạo hóa đơn trùng lặp * Hóa đơn đã approve không được chỉnh sửa hoặc xóa * Credit balance sẽ sử dụng cho phí sử dụng phòng gym của từng thành viên công ty khách hàng. Tính chi tiết hơn ở Hóa đơn trả sau | | |

## 3. Biểu đồ lớp UC ở pha phân tích

* Bước 1: Ban đầu, giao diện chính của kế toán → đề xuất lớp GDChinhKeToan, cần các thành phần:
  + Chọn "Tạo hóa đơn trả trước": kiểu submit
* Bước 2: Giao diện danh sách công ty cần tạo hóa đơn → đề xuất lớp GDDanhSachCongTy, cần có các thành phần:
  + Bảng danh sách công ty cần tạo hóa đơn: vừa output, vừa submit
  + Nút chọn công ty: kiểu submit
* Để có danh sách công ty cần tạo hóa đơn ở bước 2, cần xử lí dưới hệ thống:
  + Tìm các công ty cần tạo hóa đơn trong chu kỳ hiện tại
  + Input: thời điểm hiện tại
  + Output: danh sách CongTy
  + Đề xuất phương thức getCongTyCanTaoHoaDon(), gán cho lớp CongTy
* Bước 3: Giao diện kiểm tra gói đăng ký
* Bước 4, giao diện hiển thị thông tin gói → đề xuất lớp GDThongTinGoi, cần có các thành phần:
  + Hiển thị thông tin gói đăng ký: kiểu output
  + Hiển thị giá cơ bản: kiểu output
  + Hiển thị note giảm giá/ thu thêm phí: kiểu output
  + Nút tiếp tục: kiểu submit
* Để kiểm tra gói đăng ký tháng sau, cần xử lí bên dưới hệ thống:
  + Tìm gói đăng ký của công ty cho tháng tiếp theo
  + Input: công ty, tháng tiếp theo
  + Output: GoiDangKy
  + Đề xuất phương thức getGoiThangSau(), gán vào lớp GoiDangKy
* Để có note giảm giá/thu thêm phí từ Sales, cần xử lí bên dưới hệ thống:
* Tìm ghi chú đặc biệt từ Sales cho công ty trong tháng
* Input: công ty, tháng
* Output: NoteGiamGia (có thể null)
* Đề xuất phương thức getNoteGiamGiaChoThang(), gán cho lớp NoteGiamGia
* Bước 5, giao diện xác nhận tạo hóa đơn → đề xuất lớp GDTaoHoaDon, cần có:
  + Ô nhập thêm chi tiết hóa đơn gồm mô tả, số tiền: vừa input, vừa output
  + Hiển thị tổng kết thông tin hóa đơn: kiểu output
  + Nút xác nhận tạo: kiểu submit
  + Nút hủy: kiểu submit
* Đề xuất thêm lớp entity ChiTietTraTruoc để lưu các line item bổ sung:
  + Thuộc tính: mô tả, số tiền, loại (giảm/tăng)
  + Đề xuất phương thức taoChiTietBoSung(), gán cho lớp ChiTietTraTruoc
* Để tạo hóa đơn mới, cần xử lí từ dưới hệ thống:
* Tạo hóa đơn với thông tin đã nhập
* Input: công ty, gói đăng ký, chi tiết hóa đơn bổ sung (nếu có)
* Output: HoaDonTraTruoc
* Đề xuất phương thức taoHoaDonMoi(), gán cho lớp HoaDonTraTruoc
* Bước 6, giao diện phê duyệt → đề xuất lớp GDPheDuyet, cần có:
  + Hiển thị chi tiết hóa đơn cần duyệt: kiểu output
  + Nút phê duyệt: kiểu submit
  + Nút từ chối: kiểu submit
  + Ô nhập lý do từ chối: kiểu input
* Để gửi thông báo phê duyệt, cần xử lí bên dưới hệ thống:
  + Tìm kế toán trưởng để gửi thông báo
  + Input: hóa đơn
  + Output: KeToanTruong
  + Đề xuất phương thức getKeToanTruongCanDuyet(), gán cho lớp NhanVien
* Bước 7, giao diện thông báo → đề xuất lớp GDThongBao, cần có:
  + Hiển thị thông báo kết quả: kiểu output
  + Nút hoàn thành: kiểu submit
* Để cập nhật credit balance, cần xử lí bên dưới hệ thống:
* Cập nhật số dư credit cho công ty
* Input: công ty, số tiền hóa đơn
* Output: boolean (thành công/thất bại)
* Đề xuất phương thức capNhatSoDuCredit(), gán cho lớp CongTy
* Để gửi email hóa đơn, cần xử lí bên dưới hệ thống:
* Gửi email hóa đơn cho công ty
* Input: hóa đơn, email công ty
* Output: boolean (thành công/thất bại)
* Đề xuất phương thức guiEmailHoaDon(), gán cho lớp HoaDonTraTruoc

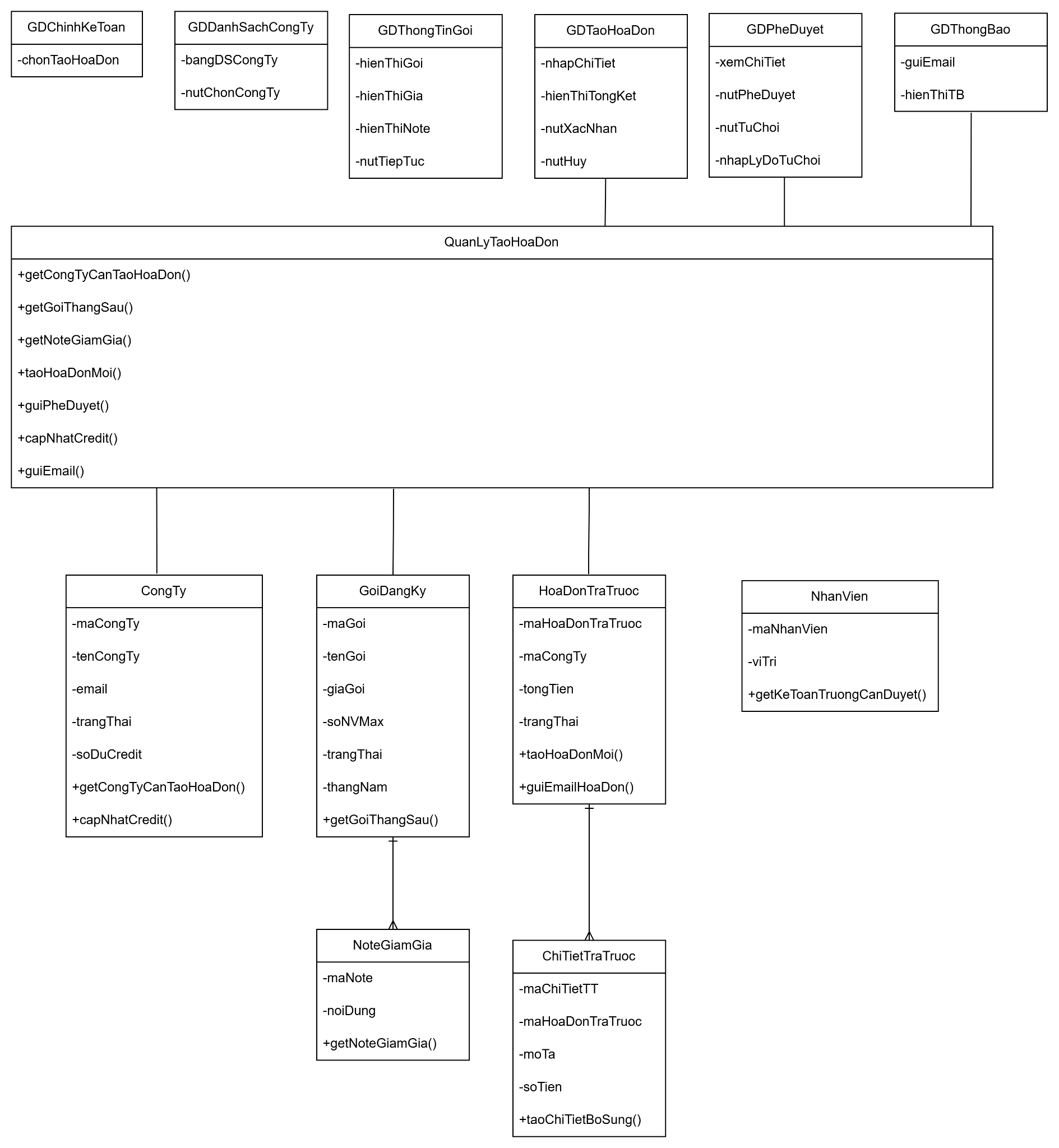
Tóm tắt các lớp và phương thức đề xuất:

Boundary Classes:

* GDChinhKeToan, GDDanhSachCongTy, GDThongTinGoi, GDGiamGia, GDTaoHoaDon, GDPheDuyet, GDThongBao

Entity Classes với phương thức:

* CongTy: getCongTyCanTaoHoaDon(), capNhatSoDuCredit()
* GoiDangKy: getGoiThangSau()
* NoteGiamGia getNoteGiamGia()
* ChiTietTraTruoc: taoChiTietBoSung()
* HoaDonTraTruoc: taoHoaDonMoi(), guiEmailHoaDon()
* NhanVien: getKeToanTruongCanDuyet()

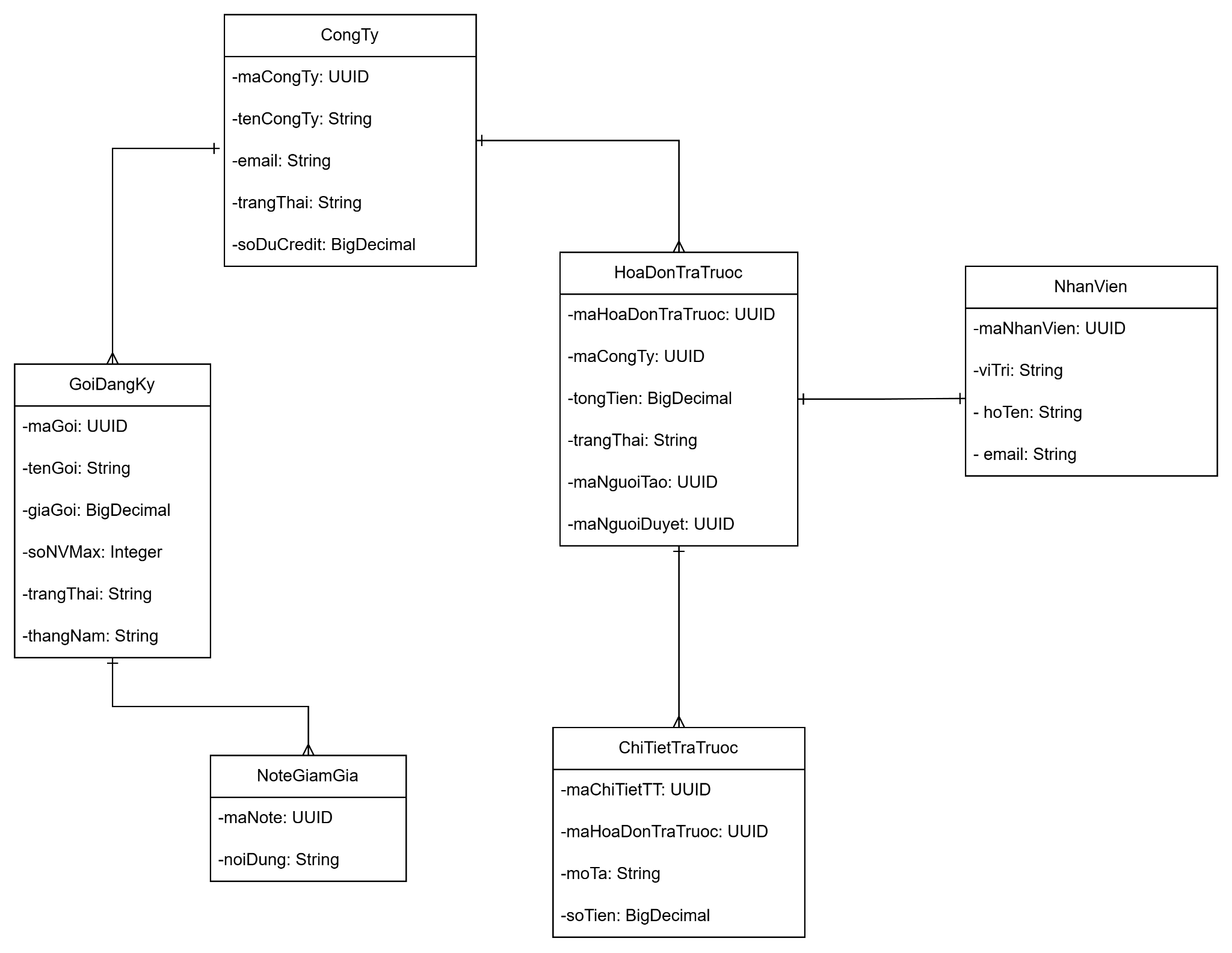


## 4. Biểu đồ lớp UC ở pha thiết kế

Đầu vào: Biểu đồ lớp thực thể pha phân tích

CongTy, GoiDangKy, HoaDonTraTruoc, ChiTietTraTruoc, NoteGiamGia, NhanVien

* Bước 1: Bổ sung kiểu dữ liệu theo Java
  + Thuộc tính của các lớp được bổ sung kiểu dữ liệu Java (UUID, String, BigDecimal, LocalDateTime, etc.)
* Bước 2: Chuyển đổi quan hệ association sang aggregation/composition
* Quan hệ CongTy - GoiDangKy: GoiDangKy chứa CongTy
* Quan hệ CongTy - HoaDonTraTruoc: HoaDonTraTruoc chứa CongTy
* Quan hệ HoaDonTraTruoc - ChiTietTraTruoc: HoaDonTraTruoc chứa List<ChiTietTraTruoc>
* Quan hệ GoiDangKy- NoteGiamGia: GoiDangKy chứa NoteGiamGia
* Quan hệ HoaDonTraTruoc - NhanVien: HoaDonTraTruoc chứa NhanVien (người tạo, người duyệt)



# **Hồ Thân Chính - Quản lý gói đăng ký với khách hàng**

## 1. Biểu đồ Use case

## 2. Viết đặc tả UC

### 2.1. Use case đăng ký dịch vụ

| Use Case Specification | | | |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-SERVICE-REGISTER | | |
| **Use Case Name:** | Đăng ký dịch vụ | | |
| **Created By:** | Hồ Thân Chính | | |
| **Primary Actor:** | Nhân viên kinh doanh | **Secondary Actors:** | Quản trị viên công ty |
| **Trigger:** | - Nhân viên kinh doanh thực hiện sale, đăng ký bán gói cho khách hàng, công ty. - Có công ty khách hàng chủ động liên hệ mua gói tập cho nhân viên. | | |
| **Description:** | Nhân viên kinh doanh tạo mới một gói đăng ký cho công ty, hệ thống kiểm tra ràng buộc (số lượng nhân viên ≤ sức chứa gói) và tùy theo lựa chọn thanh toán sẽ cấp credit hoặc chờ kế toán/phần thanh toán. | | |
| **Preconditions:** | - Nhân viên kinh doanh đã đăng nhập và có quyền thao tác trên khách hàng doanh nghiệp.  - Hồ sơ Công ty khách hàng đã tồn tại trong hệ thống  - Thông tin số lượng nhân viên đã được cập nhật/đồng bộ | | |
| **Post–conditions:** | - **Thành công:**   * Dữ liệu đăng ký gói tập được tạo với trạng thái phù hợp ACTIVE. * Email/notification gửi cho quản trị viên công ty khách hàng.   **Thất bại:** Hệ thống ghi log lỗi, thông báo cho nhân viên kinh doanh. | | |
| **Normal Flow:** | 1. Nhân viên kinh doanh mở màn hình “Tạo gói đăng ký” (GDSales → chọn Công ty). 2. Hệ thống hiển thị thông tin công ty và **số lượng nhân viên hiện tại**. 3. Nhân viên kinh doanh chọn 1 trong 4 gói cố định (A, B, C, D). Hệ thống hiển thị chi tiết gói: giá gói, dung lượng (A: $500, 5 người; B: $1000, 8; C: $1500, 10; D: $2000, 12), credit được cấp (theo gói). 4. Nhân viên kinh doanh xác nhận (và chọn tùy chọn thanh toán: a) thanh toán ngay bằng Payment Gateway, hoặc b) tạo subscription và yêu cầu Accounting phát hành hóa đơn trả trước). 5. Nếu Sales chọn **thanh toán ngay**: Hệ thống gọi Payment Gateway để xử lý thanh toán.  5.1. Nếu thanh toán thành công → Hệ thống tạo subscription với trạng thái **ACTIVE**, cập nhật credit\_balance = credit\_of\_package, gửi thông báo/email xác nhận cho Company Admin, tạo transaction record.  5.2. Nếu thanh toán thất bại → Hệ thống tạo subscription với trạng thái **PENDING\_PAYMENT**, ghi note, gửi thông báo cho Sales và Company Admin, tạo task cho Accounting follow-up. 6. Nếu Sales chọn **tạo subscription & báo Accounting** → Hệ thống tạo subscription ở trạng thái **PENDING\_APPROVAL**, tạo bản ghi yêu cầu tạo hóa đơn trả trước (danh sách cho Accounting). Gửi notification tới Accounting Staff. 7. Use case kết thúc (subscription có trạng thái phù hợp, thông báo gửi các bên). | | |
| **Alternative Flows:** | **3a — Số lượng nhân viên > sức chứa gói**  3a.1 Hệ thống phát hiện employee\_count vượt giới hạn gói.  3a.2 Hệ thống hiển thị cảnh báo và gợi ý các gói phù hợp.  3a.3 Nhân viên kinh doanh chọn gói khác (quay về bước 3) hoặc hủy thao tác.  **4a — Công ty đã có subscription active cho cùng kỳ**  4a.1 Hệ thống phát hiện tồn tại subscription active cùng kỳ.  4a.2 Hệ thống hỏi Sales: “Tạo subscription mới sẽ ghi đè/tiếp nối hay tạo subscription bổ sung?”  4a.3 Nếu tạo đè → hệ thống yêu cầu confirm/hủy lịch sử; nếu không → hủy thao tác. | | |
| **Exceptions:** | * **E1 — Lỗi lưu dữ liệu (DB)**  E1.1 Hệ thống gặp lỗi khi ghi subscription → rollback.  E1.2 Ghi log lỗi chi tiết; thông báo lỗi cho nhân viên kinh doanh. * **E2 — Lỗi Payment Gateway**  E2.1 Giao dịch bị từ chối/timeout → hệ thống thông báo “Thanh toán thất bại”, tạo subscription ở trạng thái PENDING\_PAYMENT (nếu đã tạo). * **E3 — Thiếu thông tin nhân viên**  E3.1 Nếu hệ thống không có employee\_count đáng tin cậy → hiển thị cảnh báo, yêu cầu Sales xác nhận hoặc liên hệ Company Admin để cập nhật trước khi tạo. | | |
| **Priority:** | Cao | | |
| **Frequency of Use** | Khi có khách hàng doanh nghiệp mới muốn đăng ký hoặc doanh nghiệp mở rộng gói (không cố định). | | |
| **Business Rules:** | * Hệ thống chỉ cung cấp 4 gói cố định (A/B/C/D) với thông số đã nêu. * Một công ty không có 2 subscription active cho cùng 1 kỳ (month). * Nếu thanh toán ngay thành công → credit balance được cập nhật tức thời. Nếu không, subscription ở trạng thái chờ và Accounting sẽ xử lý hóa đơn. * Chỉ nhân viên kinh doanh có quyền chính để tạo subscription; Company Admin không thể bypass quy trình thanh toán do sales thiết lập. | | |

### 2.2. Use case thay đổi gói dịch vụ

| Use Case Specification | | | |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-SERVICE-CHANGE | | |
| **Use Case Name:** | Thay đổi dịch vụ | | |
| **Created By:** | Hồ Thân Chính | | |
| **Primary Actor:** | Nhân viên kinh doanh | **Secondary Actors:** | Quản trị viên công ty |
| **Trigger:** | - Công ty khách hàng yêu cầu thay đổi gói dịch vụ. | | |
| **Description:** | Nhân viên kinh doanh thực hiện thay đổi gói dịch vụ cho công ty. Theo ràng buộc nghiệp vụ, \*\*thay đổi chỉ được phép vào **ngày đầu tiên của tháng**. Hệ thống kiểm tra điều kiện, tính toán điều chỉnh billing (nếu cần), cập nhật subscription và thông báo các bên liên quan. | | |
| **Preconditions:** | - Nhân viên kinh doanh đã đăng nhập và có quyền thao tác trên khách hàng doanh nghiệp.  - **Hôm nay là ngày 1 của tháng** (theo ràng buộc: chỉ được thay đổi vào ngày đầu tháng). | | |
| **Post–conditions:** | - **Thành công:**   * Dữ liệu đăng ký gói tập được cập nhật sang gói mới (trạng thái ACTIVE), credit và billing được điều chỉnh theo Business Rules, thông báo gửi tới Accounting và Company Admin. * Email/notification gửi cho quản trị viên công ty khách hàng.   **Thất bại:** Hệ thống ghi log lỗi, thông báo cho nhân viên kinh doanh. | | |
| **Normal Flow:** | 1. Nhân viên kinh doanh chọn công ty cần thay đổi gói. 2. Hệ thống kiểm tra ngày hiện tại: nếu ≠ ngày 1 → từ chối (see Alternative Flow 2a). 3. Hệ thống truy xuất subscription hiện tại, trạng thái credit balance, và usage (nếu có). 4. Nhân viên kinh doanh chọn gói mới (A/B/C/D). 5. Hệ thống kiểm tra: employee\_count ≤ capacity\_new?   5.1 Nếu không → hiển thị yêu cầu nâng cấp hoặc giảm nhân viên, dừng thao tác.   1. Hệ thống tính billing adjustment:   6.1 Nếu price\_new > price\_old → tạo Adjustment Invoice (invoice “chênh lệch gói tháng này”).  6.2 Nếu price\_new < price\_old → theo chính sách hiện hành (tham khảo Business Rules): không hoàn tiền ngay; ghi nhận giảm giá cho kỳ hiện tại/hoặc tạo Credit Note (xem chú ý trong Business Rules).   1. Nhân viên kinh doanh xác nhận thay đổi. 2. Hệ thống cập nhật subscription (gói mới), cập nhật assigned\_credit theo gói mới. 3. Hệ thống tạo record cho Accounting để xử lý invoice điều chỉnh (nếu có) và gửi thông báo cho Company Admin. 4. Use case kết thúc. | | |
| **Alternative Flows:** | * **2a — Không phải ngày 1**  2a.1 Hệ thống từ chối thao tác, hiển thị thông báo “Thay đổi gói chỉ được thực hiện vào ngày đầu tháng”. Use case kết thúc. * **5a — Employee count vượt capacity gói mới**  5a.1 Hệ thống thông báo chi tiết.  5a.2 Sales Staff chọn gói khác hoặc đề xuất nâng cấp số nhân viên (không khả thi) → quay lại bước 4. * **6a — Prepaid invoice cho tháng đã được tạo & approved**  6a.1 Nếu Accounting đã tạo và APPROVED hóa đơn trả trước cho tháng (chu kỳ 25→5) → Hệ thống sẽ:   6a.2 nếu upgrade: tạo invoice bổ sung cho phần chênh lệch;  6a.3 nếu downgrade: tạo ghi chú (Credit Note) hoặc áp dụng giảm cho chu kỳ tiếp theo tùy chính sách. 6a.4 Gửi notification tới Accounting để xử lý phê duyệt lại/adjust. | | |
| **Exceptions:** | * **E1 — Lỗi cập nhật subscription →** rollback, log, thông báo. * **E2 — Lỗi khi tạo adjustment invoice →** ghi log, thông báo Accounting & Sales, subscription vẫn được cập nhật nhưng ghi chú “invoice pending”. * **E3 — Conflicting subscription states (ví dụ: subscription đang trong progress cancel) →** từ chối và yêu cầu xử lý trạng thái trước. | | |
| **Priority:** | Cao | | |
| **Frequency of Use** | Hầu hết diễn ra vào **ngày 1 hàng tháng** (khi có doanh nghiệp muốn thay đổi gói). | | |
| **Business Rules:** | **Thời gian:** Chỉ thực hiện thay đổi gói vào **ngày đầu tiên của tháng**. Không được thay đổi giữa tháng.  **Ràng buộc dung lượng:** Sau khi đổi gói, số nhân viên phải ≤ capacity của gói mới. Nếu vượt, bắt buộc upgrade.  **Billing adjustment:**   * **Upgrade:** tính chênh lệch giá gói cho tháng và tạo invoice bổ sung (không tự động trừ vào credit đã cấp). * **Downgrade:** không hoàn tiền cho phần chênh lệch của tháng nếu công ty đã trả trước; hậu quả tài chính do chính sách sale quy định (ví dụ: áp dụng giảm cho giảm kỳ tiếp theo hoặc tạo credit note — cần quy định rõ trong chính sách doanh nghiệp).   **Invoice lifecycle:** Nếu prepaid invoice đã được tạo và xác nhận, thay đổi phải tạo adjustment invoice / credit note và chờ Accounting xử lý.  **Bảo mật/Quyền:** Chỉ Sales Staff mới có quyền thực hiện thay đổi; mọi thay đổi cần audit trail (ai đổi, lúc nào, lý do). | | |

### 2.3. Use case hủy gói dịch vụ

| Use Case Specification | | | |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-SERVICE-CANCEL | | |
| **Use Case Name:** | Hủy gói dịch vụ | | |
| **Created By:** | Hồ Thân Chính | | |
| **Primary Actor:** | Nhân viên kinh doanh | **Secondary Actors:** | Quản trị viên công ty |
| **Trigger:** | - Công ty yêu cầu chấm dứt subscription/gói dịch vụ. | | |
| **Description:** | Nhân viên kinh doanh tiến hành hủy subscription của công ty. Hệ thống kiểm tra ràng buộc (không hủy giữa tháng — chỉ được hủy theo chính sách; theo dữ liệu bạn cung cấp: **không cho phép hủy giữa tháng**, do đó hủy chỉ thực hiện vào ngày đầu tháng), tính toán phí phát sinh (phí vượt credit nếu có) và tạo hóa đơn cuối cùng cho công ty. | | |
| **Preconditions:** | - Nhân viên kinh doanh đã đăng nhập và có quyền thao tác trên khách hàng doanh nghiệp.  - Công ty đang có subscription với trạng thái ACTIVE.  - Hủy chỉ được phép vào ngày đầu tiên của tháng (theo ràng buộc). | | |
| **Post–conditions:** | - **Thành công:**   * Hệ thống tính phí vượt (nếu có), tạo hóa đơn cuối cùng (OVERAGE INVOICE) với trạng thái SUBMITTED → Accounting xử lý; subscription được set CANCELLED * Email/notification gửi cho quản trị viên công ty khách hàng.   **Thất bại:** Hệ thống ghi log lỗi, thông báo cho nhân viên kinh doanh. | | |
| **Normal Flow:** | 1. Nhân viên kinh doanh chọn công ty cần hủy gói. 2. Hệ thống hiển thị thông tin subscription hiện tại, credit balance, và usage lịch sử (đặc biệt usage của tháng đang xét). 3. Nhân viên kinh doanh xác nhận yêu cầu hủy (nhập lý do, ngày hủy nếu khác mặc định). 4. Hệ thống kiểm tra **hóa đơn & công nợ**: nếu còn hóa đơn unpaid (ở bất kỳ kỳ nào) → chuyển sang Alternative Flow 4a. 5. Hệ thống tính toán phí vượt (nếu used\_hours\_cost > credit\_value) → overage\_amount = used\_cost - credit\_consumed\_value. 6. Hệ thống tạo **Hóa đơn cuối cùng** (Final Invoice) gồm: phí vượt (bắt buộc), các khoản phải thu còn tồn (nếu có), ghi rõ không phải trả credit cho “tháng sau”. Hóa đơn tạo với trạng thái **SUBMITTED**. 7. Hệ thống cập nhật trạng thái subscription = **CANCELLED** (ghi log audit: ai, khi nào, lý do). 8. Hệ thống gửi email/hóa đơn & notification cho Company Admin và thông báo cho Accounting xử lý thanh toán. 9. Use case kết thúc. | | |
| **Alternative Flows:** | **4a —** Công ty còn nợ hóa đơn cũ (unpaid invoices) 4a.1 Hệ thống phát hiện unpaid invoices > 0.  4a.2 Hệ thống thông báo bắt buộc phải thanh toán nợ trước khi hủy.  4a.3 Nhân viên kinh doanh lựa chọn: a) yêu cầu Company thanh toán ngay (giao tiếp/push payment request), hoặc b) tạo thỏa thuận thanh toán (payment plan) → nếu thanh toán hoàn tất → quay lại bước 5; nếu không → hủy thao tác.  **3a —** Yêu cầu hủy vào giữa tháng (không được phép)  3a.1 Nếu không phải ngày 1 → hệ thống từ chối và hiển thị thông báo “Không cho phép hủy giữa tháng. Vui lòng thực hiện vào ngày đầu tháng”. Use case kết thúc.  **6a —** Hệ thống không thể tính usage do thiếu dữ liệu  6a.1 Hệ thống báo lỗi, tạo ticket cho System Admin/Monitoring để điều tra dữ liệu check-in/out, tạm hoãn hủy cho tới khi dữ liệu hoàn tất. | | |
| **Exceptions:** | * **E1 — Lỗi tạo hóa đơn (DB/Service)** → rollback, log, thông báo Sales & Accounting. * **E2 — Lỗi gửi email/hóa đơn** → ghi log cảnh báo; đánh dấu cần gửi lại; use case vẫn coi là thành công về nghiệp vụ (order canceled) nhưng notification bị ghi nhận gửi lại. * **E3 — Payment failure when clearing unpaid invoices** → dừng hủy, thông báo.. | | |
| **Priority:** | Cao | | |
| **Frequency of Use** | Khi công ty chấm dứt hợp đồng / dừng dịch vụ (không thường xuyên). | | |
| **Business Rules:** | **Thời gian:** Không cho phép hủy giữa tháng — hủy chỉ thực hiện vào **ngày đầu tháng** (theo ràng buộc).  **Phí vượt:** Khi hủy, công ty phải thanh toán mọi phí vượt phát sinh trong tháng hiện tại.  **Credit tháng sau:** Công ty **không phải trả** credit cho tháng sau khi hủy (tức không cần prepay cho kỳ tiếp theo).  **Công nợ:** Nếu công ty còn nợ hóa đơn cũ, bắt buộc thanh toán (hoặc thỏa thuận) trước khi hủy.  **Không re-activate:** Subscription khi đã hủy không được kích hoạt lại — nếu công ty muốn trở lại, Sales phải tạo subscription mới.  **Audit:** Mọi thao tác hủy phải được audit (ai hủy, lý do, thời gian). | | |

## 3. Biểu đồ lớp UC ở pha phân tích

**GDChinhSales**

* **Vai trò:** màn hình chính cho Sales (hoặc Sales Staff).
* **Thành phần/controls:** nút chonDangKyGoi(), chonThayDoiGoi(), chonHuyGoi(), bảng danh sách công ty (search/filter).

**GDDangKyGoi**

* **Thuộc tính hiển thị:** thông tin công ty, employee count, list package (A–D), payment method.
* **Phương thức chính:** nhapThongTinDangKy() thực hiện thu dữ liệu -> gọi QuanLyGoiDangKy.dangKyGoi(maCT, maGoi, ngayBD)
* **Validation UI:** kiểm tra format ngày, chọn package. Nếu lỗi (vd: employeeCount > capacity) -> hiển thị message trả về từ control.

**GDThayDoiGoi**

* **Thuộc tính hiển thị:** subscription hiện tại, gói đề xuất, preview điều chỉnh billing.
* **Phương thức:** nhapThongTinThayDoi() → gọi QuanLyGoiDangKy.thayDoiGoi(maSub, goiMoi); previewAdjustment() gọi QuanLyGoiDangKy.kiemTraDieuKien() hoặc BillingService để tính chênh lệch.

**GDHuyGoi**

* **Thuộc tính:** hiển thị subscription, unpaid invoices, overage estimate.
* **Phương thức:** xacNhanHuy() → gọi QuanLyGoiDangKy.huyGoi(maSub). Có confirm dialog & yêu cầu nhập lý do hủy.

**GDThongBao**

* **Chức năng:** hiển thị kết quả, lỗi từ control, link đến invoice/chi tiết.

**QuanLyGoiDangKy**

Là lớp trung tâm thực hiện nghiệp vụ; nó phối hợp entity + repository + dịch vụ ngoại vi (payment, email).

**dangKyGoi(maCT, maGoi, ngayBD)**:

* Lấy CongTy và GoiDichVu từ repository.

Gọi kiemTraDieuKien() (kiểm tra employeeCount ≤ capacity, tồn tại subscription cùng kỳ, hợp lệ ngày bắt đầu).

* Nếu chọn thanh toán online → gọi PaymentService.process(amount, paymentInfo). Nếu thành công: tạo Subscription (status=ACTIVE), cập nhật credit, tạo hóa đơn prepaid (status APPROVED hoặc PAID tùy policy) và gửi email. Nếu payment fail → tạo Subscription (PENDING\_PAYMENT) và tạo invoice DRAFT/SUBMITTED và notify Accounting.
* Trả về Subscription hoặc lỗi.

**thayDoiGoi(maSub, goiMoi)**:

* Kiểm tra ngày cho phép (business rule: ngày 1 của tháng).
* Lấy subscription và công ty; kiểm tra soLuongNV ≤ goiMoi.capacity.
* Tính adjustmentAmount (BillingService): nếu dương → tạo ADJUSTMENT INVOICE; nếu âm → tạo credit note theo policy.

**huyGoi(maSub)**:

* Kiểm tra policy hủy (chỉ ngày 1 — nếu nghiệp vụ khác thì tuỳ policy).
* Lấy unpaid invoices của company. Nếu unpaid>0 → báo bắt buộc thanh toán/hoặc xử lý plan.
* Tính phí vượt tinhPhiVuot(maSub) (dựa trên usage sessions của tháng hiện tại).
* Tạo FINAL Invoice gồm unpaid + overage.
* Đặt subscription.status = CANCELLED, ghi audit, notify Accounting & Company.

**kiemTraDieuKien(maCT, maGoi)**: trả ValidationResult chi tiết (OK hoặc danh sách lỗi).

**tinhPhiVuot(maSub)**: scan UsageSession liên quan, tính tổng chi phí trừ credit, trả overage amount.

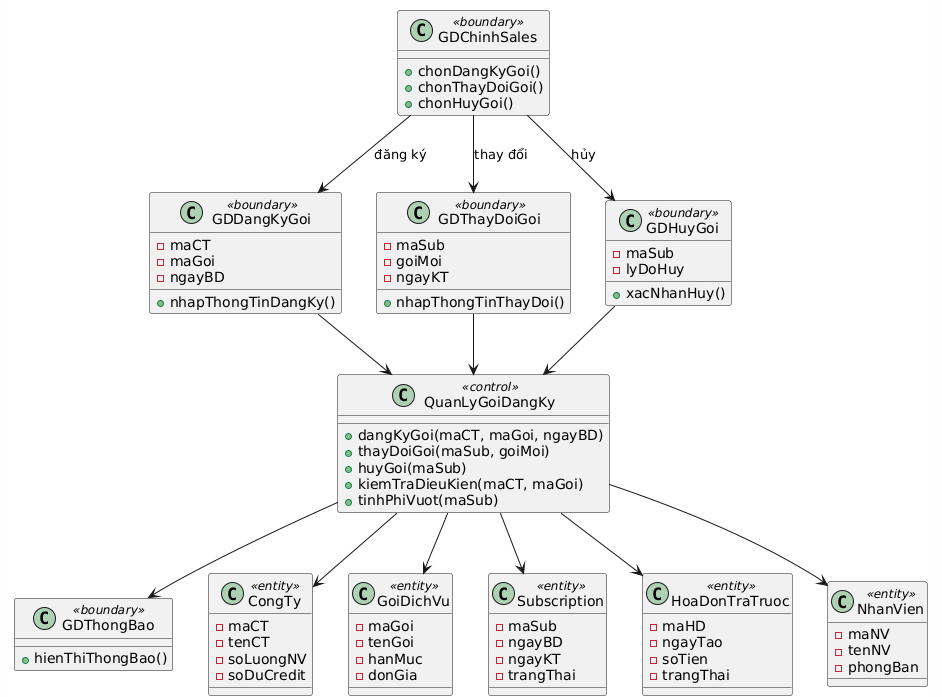
Tóm tắt các lớp và phương thức đề xuất:

Boundary Classes:

* GDChinhSales, GDDangKyGoi, GDThayDoiGoi, GDHuyGoi, GDThongBao

Entity Classes với phương thức:

* QuanLyGoiDangKy: dangKyGoi, thayDoiGoi, huyGoi, kiemTraDieuKien, tinhPhiVuot



## 4. Biểu đồ lớp UC ở pha thiết kế

**Đầu vào:** Biểu đồ lớp thực thể pha phân tích mở rộng với các bảng:

CongTy, GoiDangKy, HoaDonTraTruoc, ChiTietTraTruoc, NoteGiamGia, NhanVien, Subscription, LichSuThayDoiGoi, HuyDangKy.

* Bước 1: Bổ sung kiểu dữ liệu theo Java
  + Thuộc tính của các lớp được bổ sung kiểu dữ liệu Java (UUID, String, BigDecimal, LocalDateTime, etc.)
* Bước 2: Chuyển đổi quan hệ association sang aggregation/composition
* Quan hệ CongTy – Subscription:

→ Một CongTy chứa nhiều Subscription (composition).

* Quan hệ GoiDangKy – Subscription:

→ Một Subscription luôn gắn với một GoiDangKy.

* Quan hệ Subscription – LichSuThayDoiGoi:

→ Một Subscription có nhiều lịch sử thay đổi.

* Quan hệ Subscription – HuyDangKy:

→ Một Subscription có thể có 0 hoặc 1 yêu cầu hủy.

* Quan hệ HuyDangKy – NhanVien:

→ Một NhanVien xử lý nhiều yêu cầu hủy.

* Quan hệ CongTy – HoaDonTraTruoc:

→ Một CongTy chứa nhiều HoaDonTraTruoc.

* Quan hệ HoaDonTraTruoc – ChiTietTraTruoc:

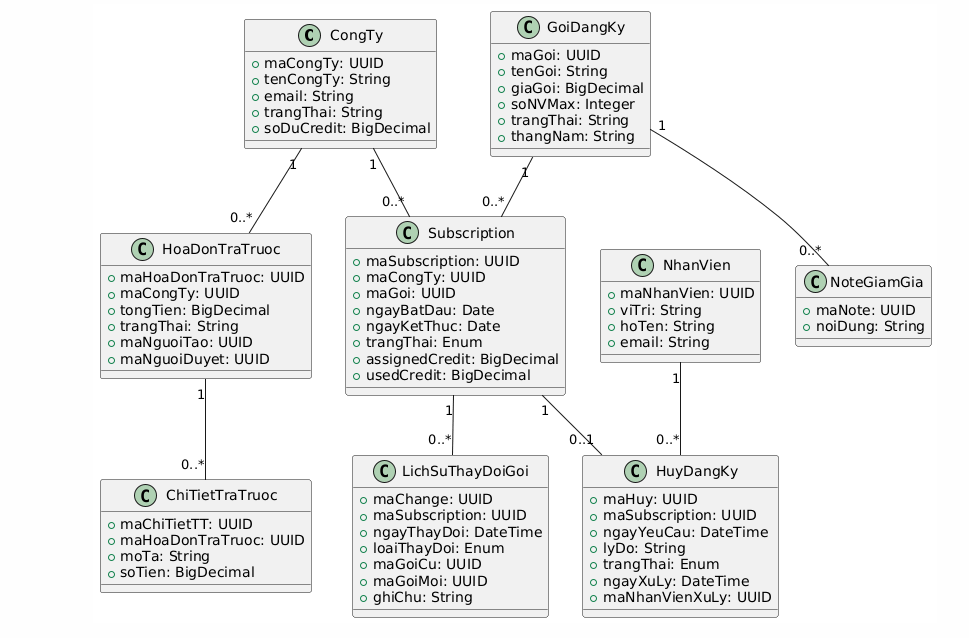
→ Một HoaDonTraTruoc chứa nhiều ChiTietTraTruoc.

* Quan hệ GoiDangKy – NoteGiamGia:

→ Một GoiDangKy có thể chứa nhiều NoteGiamGia.

* Quan hệ HoaDonTraTruoc – NhanVien:

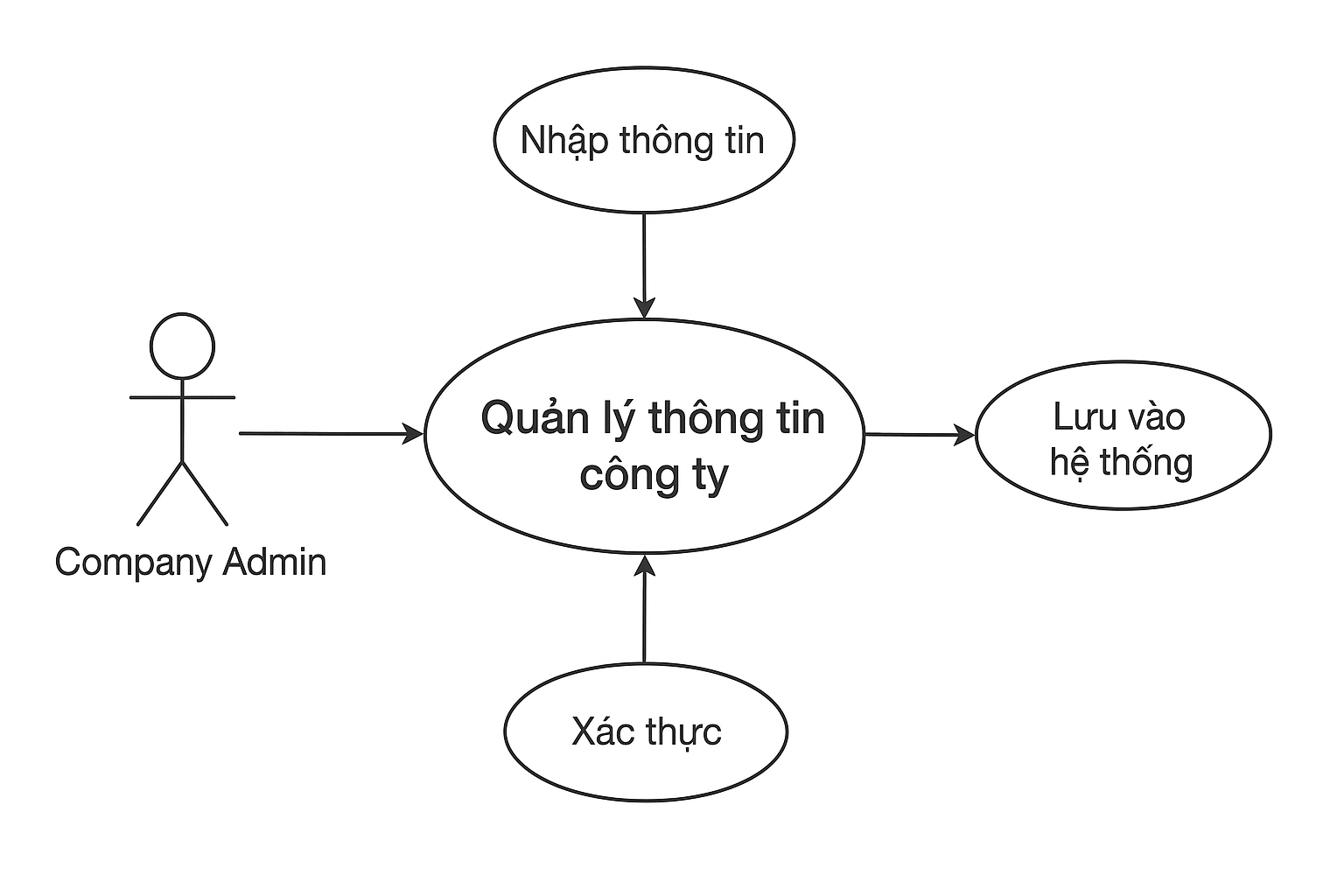
→ HoaDonTraTruoc liên quan đến NhanVien (người tạo, người duyệt).



# 

# **Hà Dương Kim Tuyến - Usecase Quản lý thông tin công ty**

## 1. Biểu đồ Use case

****

## 2. Viết đặc tả UC

| Use Case Specification | | | |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-COMPANY INFORMATION-MANAGE | | |
| **Use Case Name:** | Quản lý thông tin công ty | | |
| **Created By:** | Hà Dương Kim Tuyến | | |
| **Primary Actor:** | Quản trị viên công ty | **Secondary Actors:** | Nhân viên kinh doanh |
| **Trigger:** | Quản trị viên chọn chức năng “Quản lý thông tin công ty” trên giao diện hệ thống | | |
| **Description:** | Use case cho phép quản trị viên thực hiện các thao tác thêm, sửa, xóa và xem thông tin công ty như: tên, mã số thuế, địa chỉ, số điện thoại, email, website, và người đại diện | | |
| **Preconditions:** | - Người dùng đã đăng nhập vào hệ thống với quyền hợp lệ.  - Hệ thống hoạt động bình thường và có kết nối với cơ sở dữ liệu | | |
| **Post–conditions:** | **Thành công:** - Thông tin công ty mới hoặc cập nhật được lưu thành công vào cơ sở dữ liệu.   * Hệ thống hiển thị thông báo xác nhận “Cập nhật thông tin thành công”.   **Thất bại:** - Không có thay đổi nào được lưu vào cơ sở dữ liệu. - Hệ thống hiển thị thông báo lỗi tương ứng (ví dụ: “Thiếu thông tin bắt buộc”, “Lỗi kết nối cơ sở dữ liệu”, “Thông tin không hợp lệ”).   * Người dùng có thể quay lại biểu mẫu để chỉnh sửa và thử lại. | | |
| **Normal Flow:** | 1. Quản trị viên đăng nhập và truy cập mục “Quản lý thông tin công ty” 2. Hệ thống hiển thị danh sách thông tin công ty hiện có. 3. Quản trị viên chọn chức năng: **Thêm mới**, **Chỉnh sửa**, hoặc **Xóa** thông tin 4. Hệ thống hiển thị form nhập/chỉnh sửa thông tin công ty. 5. Quản trị viên nhập hoặc cập nhật thông tin công ty (tên, địa chỉ, mã số thuế, số điện thoại, email, website…). 6. Quản trị viên nhấn Lưu. 7. Hệ thống kiểm tra dữ liệu nhập vào, lưu thông tin vào cơ sở dữ liệu. 8. Hệ thống hiển thị thông báo: “✅ Cập nhật thông tin công ty thành công.” | | |
| **Alternative Flows:** | **5a Dữ liệu không hợp lệ** Hệ thống hiển thị thông báo lỗi: “⚠️ Vui lòng nhập đầy đủ và đúng định dạng thông tin.”  **7a Thông tin trùng lặp** Nếu tên hoặc mã số thuế công ty đã tồn tại, hệ thống cảnh báo: “⚠️ Thông tin công ty đã tồn tại trong hệ thống.” | | |
| **Exceptions:** | * **E1 — Mất kết nối cơ sở dữ liệu** Hệ thống hiển thị thông báo lỗi và ghi log sự cố: “❌ Không thể kết nối đến cơ sở dữ liệu.” * **E2 — Người dùng không có quyền truy cập** Hệ thống thông báo: “🚫 Bạn không có quyền thực hiện thao tác này.” | | |
| **Priority:** | Cao | | |
| **Frequency of Use** | Thỉnh thoảng (chỉ khi cần đăng ký công ty mới hoặc cập nhật thông tin công ty) | | |
| **Business Rules:** | 1. Mỗi công ty phải có **mã số thuế duy nhất**.  2. **Tên công ty** không được trùng với công ty khác trong hệ thống.  3. **Email liên hệ** phải có định dạng hợp lệ (ví dụ: contact@domain.com).  4. Thông tin công ty chỉ có thể được **chỉnh sửa bởi Company Admin**.  5. Hệ thống phải **ghi nhật ký thay đổi (log)** mỗi khi thông tin công ty được cập nhật. | | |

## 3. Biểu đồ lớp UC ở pha phân tích

**Bước 1:** Giao diện chính của người quản lý công ty → đề xuất lớp GDChinhQuanLyCongTy, cần các thành phần:

* 1. Nút “Thêm công ty mới”: kiểu submit
  2. Nút “Cập nhật thông tin công ty”: kiểu submit

**Bước 2:** Giao diện nhập thông tin công ty mới hoặc cập nhật thông tin → Đề xuất lớp GDNhapThongTinCongTy, cần có các thành phần:

* + Ô nhập Tên công ty: kiểu input
  + Ô nhập Mã số thuế: kiểu input
  + Ô nhập Địa chỉ trụ sở: kiểu input
  + Ô nhập Số điện thoại: kiểu input
  + Ô nhập Email liên hệ: kiểu input
  + Ô nhập Người đại diện pháp lý: kiểu input
  + Nút “Lưu thông tin”: kiểu submit
  + Nút “Hủy bỏ”: kiểu submit
* Để xử lý khi người dùng nhấn “Lưu thông tin”, cần xử lý bên dưới hệ thống:

**Kiểm tra dữ liệu hợp lệ**:

* + Input: thông tin công ty (tên, mã số thuế, email, v.v.)
  + Output: boolean (hợp lệ hoặc không)

→ Đề xuất phương thứckiemTraHopLe(), gán cho lớp CongTy

**Bước 3:** Kiểm tra trùng lặp thông tin công ty → Hệ thống cần xử lý bên dưới:

Kiểm tra mã số thuế hoặc tên công ty đã tồn tại

* Input: mã số thuế, tên công ty
* Output: boolean (đã tồn tại hoặc chưa)

→ Đề xuất phương thức kiemTraTonTai(), gán cho lớp CongTy

Nếu thông tin bị trùng → hiển thị cảnh báo

→ Đề xuất lớp GDCanhBaoThongTin, có:

* Hiển thị thông báo lỗi: kiểu output
* Nút “Quay lại chỉnh sửa”: kiểu submit

**Bước 4**: Lưu thông tin hợp lệ vào hệ thống → Đề xuất lớp GDXacNhanLuuThongTin, cần có:

* + Hiển thị thông tin tóm tắt: kiểu **output**
  + Nút “Xác nhận lưu”: kiểu submit

Xử lý hệ thống:

Lưu thông tin công ty mới hoặc cập nhật

* + Input: đối tượng CongTy
  + Output: boolean (thành công/thất bại)

→ Đề xuất phương thức luuThongTin(), gán cho lớp CongTy

**Bước 5**: Gửi thông báo cho Sales Staff → Hệ thống cần xử lý bên dưới:

Gửi thông báo xác nhận công ty đã được thêm/cập nhật

* + Input: thông tin công ty, người phụ trách Sales
  + Output: boolean (gửi thành công/thất bại)

→ Đề xuất phương thức guiThongBaoCapNhat(), gán cho lớp NhanVien

**Bước 6**: Giao diện thông báo kết quả → Đề xuất lớp **GDThongBaoKetQua**, cần có:

* + Hiển thị kết quả “Lưu thành công / thất bại”: kiểu **output**
  + Nút “Hoàn tất”: kiểu submit

Tóm tắt các lớp và phương thức đề xuất:

Boundary Classes:

* GDChinhQuanLyCongTy
* GDNhapThongTinCongTy
* GDCanhBaoThongTin
* GDXacNhanLuuThongTin
* GDThongBaoKetQua

Entity Classes với phương thức:

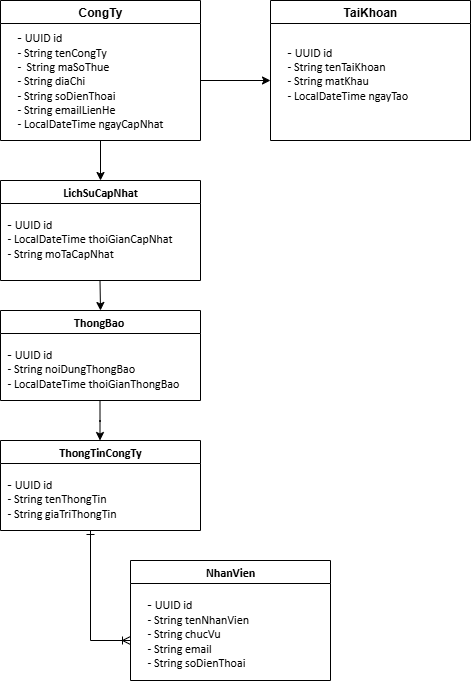
* CongTy: kiemTraHopLe(), kiemTraTonTai(), luuThongTin()
* NhanVien: guiThongBaoCapNhat()



## 4. Biểu đồ lớp UC ở pha thiết kế

Đầu vào: Biểu đồ lớp thực thể pha phân tích

CongTy, NhanVien, ThongTinCongTy, LichSuCapNhat, TaiKhoan, ThongBao

* Bước 1: Bổ sung kiểu dữ liệu theo Java
  + Thuộc tính của các lớp được bổ sung kiểu dữ liệu Java (UUID, String, BigDecimal, LocalDateTime, etc.)
* Bước 2: Chuyển đổi quan hệ association sang aggregation/composition
* Quan hệ CongTy - TaiKhoan: **Aggregation**. Mỗi công ty **CongTy** có thể chứa nhiều tài khoản **TaiKhoan**, nhưng tài khoản có thể tồn tại mà không có công ty
* Quan hệ CongTy - LichSuCapNha: **Aggregation**. Mỗi công ty **CongTy** có thể chứa nhiều lịch sử cập nhật **LichSuCapNhat**, nhưng lịch sử cập nhật có thể tồn tại mà không có công ty.
* Quan hệ CongTy - ThongBao:**Aggregation**. Mỗi công ty **CongTy** có thể liên kết với nhiều thông báo **ThongBao**, nhưng thông báo có thể tồn tại độc lập với công ty
* Quan hệ CongTy- TaiKhoan: **Association**. Mỗi tài khoản **TaiKhoan** liên kết với một công ty **CongTy**, nhưng tài khoản có thể tồn tại mà không có công ty

# **Nguyễn Anh Văn - Usecase Tạo báo cáo**

## 1. Biểu đồ Use case

## 2. Viết đặc tả UC

| **Use Case ID:** | **UC-REPORTING** |
| --- | --- |
| **Use Case Name:** | Tạo báo cáo |
| **Created By:** | Nguyễn Anh Văn |
| **Primary Actor:** | Company Admin, Sales Staff, Accounting Staff |
| **Trigger:** | - Người dùng cần xem báo cáo về sử dụng dịch vụ, doanh số, hoặc tài chính  - Đến kỳ báo cáo định kỳ (tháng/quý) |
| **Description:** | Người dùng lựa chọn loại báo cáo cần tạo (báo cáo sử dụng, doanh số, tài chính), thiết lập các tham số (khoảng thời gian, công ty, chi tiết), hệ thống tạo báo cáo và cho phép xuất file dưới dạng PDF hoặc Excel. |
| **Preconditions:** | - Người dùng đã đăng nhập với quyền phù hợp  - Dữ liệu nguồn đã được đồng bộ (usage sessions, invoices, subscriptions)  - Hệ thống báo cáo hoạt động bình thường |
| **Post-conditions:** | Thành công:  - Báo cáo được tạo thành công với đầy đủ dữ liệu  - File báo cáo được lưu tạm hoặc tải về ngay  - Lịch sử tạo báo cáo được ghi nhận  Thất bại: Hệ thống ghi log lỗi, thông báo cho người dùng về nguyên nhân |

**Normal Flow:**

1. Người dùng đăng nhập và truy cập mục "Báo cáo" (GDChinh → Chọn Báo cáo)
2. Hệ thống hiển thị danh sách các loại báo cáo có sẵn dựa trên vai trò:
   * Company Admin: Báo cáo sử dụng của nhân viên, chi tiết chi phí
   * Sales Staff: Báo cáo doanh số, subscription mới/hủy, conversion rate
   * Accounting Staff: Báo cáo tài chính, công nợ, revenue
3. Người dùng chọn loại báo cáo cần tạo
4. Hệ thống hiển thị form thiết lập tham số (GDThietLapBaoCao):
   * Khoảng thời gian (từ ngày - đến ngày, hoặc chọn preset: tháng này, quý này, năm nay)
   * Phạm vi (tất cả công ty / công ty cụ thể)
   * Mức độ chi tiết (tổng quan / chi tiết)
   * Định dạng xuất (PDF / Excel / CSV)
5. Người dùng nhập các tham số và nhấn "Tạo báo cáo"
6. Hệ thống validate dữ liệu đầu vào:
   * Kiểm tra khoảng thời gian hợp lệ (ngày bắt đầu ≤ ngày kết thúc, không vượt quá 12 tháng)
   * Kiểm tra quyền truy cập dữ liệu (ví dụ: Company Admin chỉ xem được công ty của mình)
7. Hệ thống thu thập dữ liệu từ các nguồn:
   * Báo cáo sử dụng: lấy từ UsageSession, CheckIn/CheckOut records
   * Báo cáo doanh số: lấy từ Subscription, GoiDangKy, revenue data
   * Báo cáo tài chính: lấy từ Invoice (prepaid + overage), Payment transactions
8. Hệ thống xử lý và tính toán:
   * Tổng hợp dữ liệu theo nhóm (công ty, gói, tháng)
   * Tính các chỉ số: tổng giờ sử dụng, credit tiêu thụ, phí vượt, doanh thu, công nợ
   * Tạo biểu đồ/đồ thị (nếu định dạng PDF)
9. Hệ thống tạo file báo cáo theo định dạng đã chọn
10. Hệ thống hiển thị preview báo cáo và nút "Tải xuống" (GDXemBaoCao)
11. Người dùng tải báo cáo về máy hoặc lưu vào hệ thống
12. Hệ thống ghi log: ai tạo báo cáo, loại gì, tham số nào, lúc nào
13. Use case kết thúc

**Alternative Flows:**

6a — Tham số không hợp lệ

* 6a.1 Hệ thống phát hiện khoảng thời gian không hợp lệ (ví dụ: ngày bắt đầu > ngày kết thúc, hoặc khoảng thời gian > 12 tháng)
* 6a.2 Hệ thống hiển thị thông báo lỗi cụ thể: "Vui lòng chọn khoảng thời gian hợp lệ (tối đa 12 tháng)"
* 6a.3 Người dùng chỉnh sửa tham số → quay lại bước 5

7a — Không có dữ liệu trong khoảng thời gian

* 7a.1 Hệ thống phát hiện không có dữ liệu nguồn cho khoảng thời gian đã chọn
* 7a.2 Hệ thống thông báo: "Không có dữ liệu trong khoảng thời gian này. Vui lòng chọn khoảng thời gian khác hoặc kiểm tra dữ liệu nguồn"
* 7a.3 Người dùng có thể: a) chọn khoảng thời gian khác, hoặc b) tạo báo cáo trống (template)

8a — Dữ liệu quá lớn, cần phân trang

* 8a.1 Nếu dữ liệu báo cáo vượt quá ngưỡng (ví dụ: >10,000 records), hệ thống thông báo sẽ tạo báo cáo dạng batch
* 8a.2 Hệ thống tạo job xử lý nền (background job)
* 8a.3 Gửi notification cho người dùng khi báo cáo sẵn sàng
* 8a.4 Người dùng quay lại sau để tải báo cáo từ mục "Lịch sử báo cáo"

**Exceptions:**

E1 — Lỗi truy xuất dữ liệu (database timeout, connection error)

* E1.1 Hệ thống gặp lỗi khi kết nối đến database/data warehouse
* E1.2 Hệ thống retry 3 lần với backoff
* E1.3 Nếu vẫn lỗi → hiển thị thông báo: "Không thể tạo báo cáo lúc này. Vui lòng thử lại sau hoặc liên hệ IT Support"
* E1.4 Ghi log lỗi chi tiết cho admin để troubleshoot

E2 — Lỗi tạo file (không đủ bộ nhớ, lỗi template)

* E2.1 Hệ thống không thể tạo file PDF/Excel do lỗi kỹ thuật
* E2.2 Ghi log lỗi chi tiết
* E2.3 Thông báo: "Lỗi khi tạo file báo cáo. Vui lòng thử định dạng khác hoặc giảm khoảng thời gian"
* E2.4 Use case kết thúc

E3 — Người dùng không có quyền xem dữ liệu của công ty khác

* E3.1 Company Admin cố gắng tạo báo cáo cho công ty khác
* E3.2 Hệ thống phát hiện violation → từ chối
* E3.3 Hiển thị: "Bạn không có quyền xem dữ liệu này"
* E3.4 Ghi log security event
* E3.5 Use case kết thúc

| Priority: | Cao | | Frequency of Use | Hàng ngày (Company Admin), hàng tuần (Sales), cuối tháng (Accounting) |

**Business Rules:**

Quyền truy cập:

* Company Admin chỉ xem được báo cáo của công ty mình
* Sales Staff xem được tất cả công ty nhưng không thấy chi tiết tài chính nhạy cảm
* Accounting Staff xem được toàn bộ dữ liệu tài chính
* System Admin có quyền xem mọi báo cáo

Giới hạn dữ liệu:

* Khoảng thời gian tối đa cho 1 báo cáo: 12 tháng
* Nếu >10,000 records → xử lý batch và gửi email khi hoàn thành
* Dữ liệu real-time có độ trễ tối đa 5 phút (cache invalidation)

Loại báo cáo:

* Báo cáo sử dụng (Usage Report): Tổng giờ tập, phân bố theo hạng phòng, nhân viên active, peak hours
* Báo cáo doanh số (Sales Report): Subscription mới, renewal, churn rate, revenue by package
* Báo cáo tài chính (Financial Report): Total revenue, overage charges, unpaid invoices, credit balance

Định dạng:

* PDF: Có header/footer, logo công ty, biểu đồ, format in được
* Excel: Dữ liệu thô, có thể pivot, filter
* CSV: Raw data export cho hệ thống khác

Audit:

* Mọi báo cáo được tạo đều ghi log: user, report type, parameters, timestamp
* Lưu lịch sử 90 ngày cho audit compliance

## 3. Biểu đồ lớp UC ở pha phân tích

**Bước-by-step analysis:**

Bước 1: Giao diện chính → đề xuất lớp GDChinh, có nút "Báo cáo"

Bước 2: Giao diện chọn loại báo cáo → đề xuất lớp GDChonLoaiBaoCao, cần:

* Danh sách loại báo cáo theo vai trò: kiểu output + submit
* Nút chọn: kiểu submit

Bước 3: Giao diện thiết lập tham số → đề xuất lớp GDThietLapBaoCao, cần:

* Ô chọn khoảng thời gian (date picker): kiểu input
* Dropdown chọn công ty/phạm vi: kiểu input
* Radio button chọn mức độ chi tiết: kiểu input
* Dropdown chọn định dạng: kiểu input
* Nút "Tạo báo cáo": kiểu submit

Bước 4: Để validate tham số đầu vào, cần xử lý bên dưới:

* Input: reportType, dateRange, scope, format
* Output: ValidationResult (boolean + error messages)
* → Đề xuất phương thức kiemTraThamSo(), gán cho lớp BaoCao

Bước 5: Để thu thập dữ liệu, cần xử lý:

* Input: reportType, parameters
* Output: List<DataRecord>
* → Đề xuất phương thức layDuLieuBaoCao(), gán cho lớp BaoCao

Bước 6: Để tính toán và tổng hợp, cần:

* Input: raw data, aggregation rules
* Output: processed data + metrics
* → Đề xuất phương thức xuLyDuLieu(), gán cho lớp BaoCao

Bước 7: Để tạo file báo cáo, cần:

* Input: processed data, format (PDF/Excel)
* Output: File object
* → Đề xuất phương thức taoBaoCao(), gán cho lớp BaoCao

Bước 8: Giao diện xem preview → đề xuất lớp GDXemBaoCao, cần:

* Hiển thị preview: kiểu output
* Nút "Tải xuống": kiểu submit
* Nút "Tạo lại": kiểu submit

Bước 9: Để ghi log audit, cần:

* Input: userId, reportType, parameters, timestamp
* Output: boolean (success/fail)
* → Đề xuất phương thức ghiLogBaoCao(), gán cho lớp LichSuBaoCao

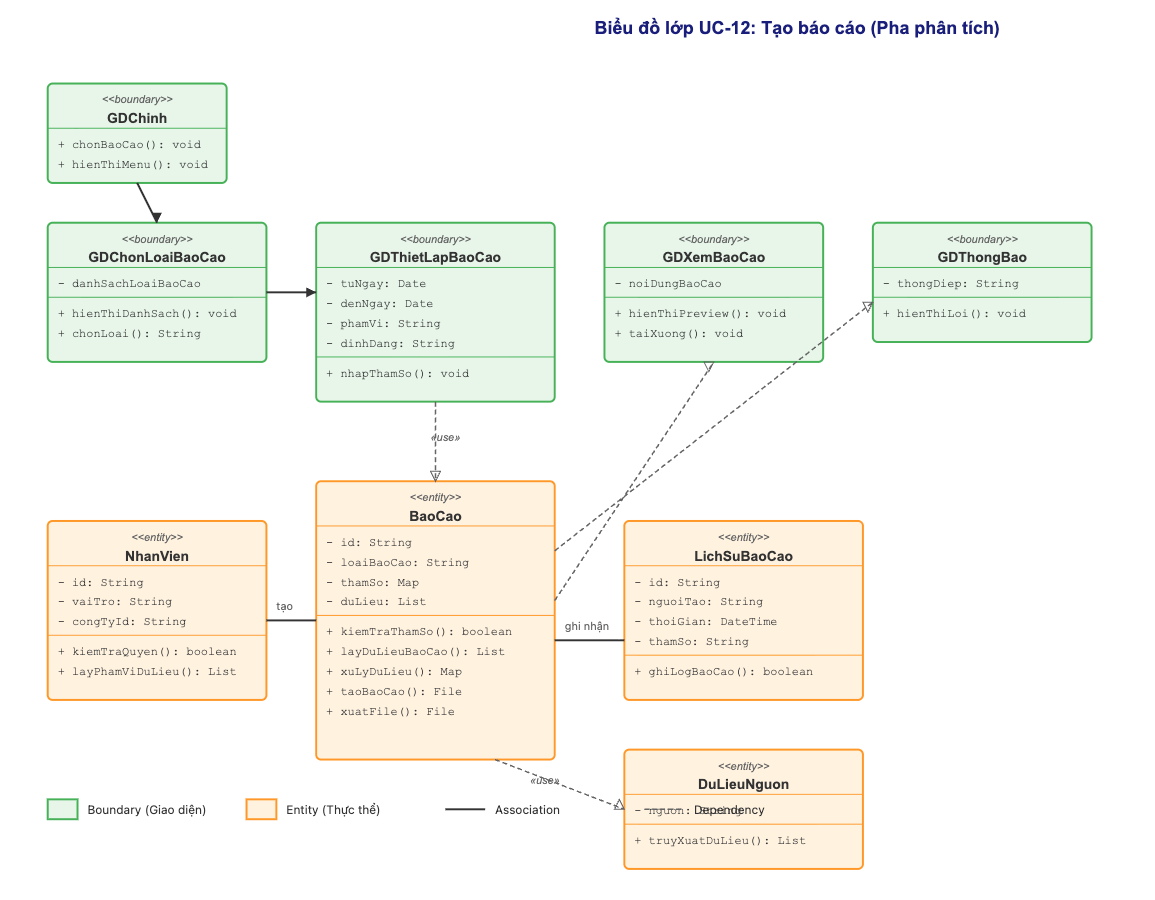
Tóm tắt các lớp:

Boundary Classes:

* GDChinh
* GDChonLoaiBaoCao
* GDThietLapBaoCao
* GDXemBaoCao

Entity Classes với phương thức:

* BaoCao: kiemTraThamSo(), layDuLieuBaoCao(), xuLyDuLieu(), taoBaoCao()
* LichSuBaoCao: ghiLogBaoCao()
* NguoiDung: kiemTraQuyen()



## 4. Biểu đồ lớp UC ở pha thiết kế