

BỘ KHOA HỌC VÀ CÔNG NGHỆ  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO  
THỰC TẬP TỐT NGHIỆP  
ĐẠI HỌC

*Đề tài: “Xây dựng hệ thống quản lý hợp đồng dịch vụ  
tại chung cư bằng kiến trúc Microservices”*

**Người hướng dẫn :** **NGUYỄN ANH HÀO**

**Nhóm sinh viên thực hiện :** **NHÓM 06**

**Sinh viên :** **NGUYỄN THÀNH LONG** - N21DCCN143

**VŨ ĐỨC TRỌNG** - N21DCCN159

**THẠCH THỊ NHANH** - N21DCCN190

**Lớp :** **D21CQCNHT01-N**

**Khoa :** **2021-2026**

**Ngành :** **CÔNG NGHỆ THÔNG TIN**

**Hệ :** **CHÍNH QUY**

**TP.HCM, tháng 08 /2025**

BỘ KHOA HỌC VÀ CÔNG NGHỆ  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO  
THỰC TẬP TỐT NGHIỆP  
ĐẠI HỌC

*Đề tài: “Xây dựng hệ thống quản lý hợp đồng dịch vụ  
tại chung cư bằng kiến trúc Microservices”*

Người hướng dẫn : NGUYỄN ANH HÀO

Nhóm sinh viên thực hiện : Nhóm 06

Sinh viên	:	NGUYỄN THÀNH LONG	- N21DCCN143
		VŨ ĐỨC TRỌNG	- N21DCCN159
		THẠCH THỊ NHANH	- N21DCCN190

Lớp : D21CQCNHT01-N

Khoa : 2021-2026

Ngành : CÔNG NGHỆ THÔNG TIN

Hệ : CHÍNH QUY

TP.HCM, tháng 8/2025

## LỜI CẢM ƠN

Trong quá trình thực hiện và hoàn thành đề tài "Xây dựng hệ thống quản lý hợp đồng và dịch vụ cho cư dân và ban quản lý", chúng em xin gửi lời cảm ơn sâu sắc tới các cá nhân và tổ chức đã hỗ trợ và đồng hành cùng chúng em trong suốt quá trình nghiên cứu và phát triển hệ thống này.

Đầu tiên, chúng em xin cảm ơn giảng viên hướng dẫn – Thầy Nguyễn Anh Hào đã luôn tận tình chỉ bảo, hướng dẫn và cung cấp những kiến thức quý báu giúp chúng em hoàn thiện luận văn này. Sự nhiệt tình và kinh nghiệm của thầy/cô đã giúp chúng em vượt qua những khó khăn và thách thức trong suốt quá trình thực hiện.

Chúng em cũng xin chân thành cảm ơn ban quản lý dự án và các đồng nghiệp trong nhóm phát triển hệ thống đã đóng góp ý tưởng, hỗ trợ kỹ thuật và làm việc chăm chỉ để hệ thống được triển khai thành công. Sự hợp tác chặt chẽ và tinh thần đồng đội của các bạn là yếu tố quan trọng giúp chúng em hoàn thành đề tài này.

Cảm ơn gia đình đã luôn bên cạnh, động viên và tạo điều kiện thuận lợi cho chúng em trong suốt quá trình nghiên cứu và phát triển hệ thống. Sự động viên của gia đình là nguồn động lực lớn giúp chúng em không ngừng cố gắng.

Cuối cùng, chúng em xin gửi lời cảm ơn tới tất cả những người đã giúp đỡ, đóng góp ý kiến và hỗ trợ chúng em trong suốt quá trình hoàn thiện luận văn này. Chúng em hy vọng rằng kết quả nghiên cứu sẽ có ích cho cộng đồng và đóng góp vào sự phát triển của công nghệ quản lý trong lĩnh vực dịch vụ.

Xin trân trọng cảm ơn!

## MỤC LỤC

LỜI CẢM ƠN.....	3
<b>MỤC LỤC .....</b>	<b>4</b>
DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH.....	9
Hình.....	9
Bảng .....	11
<b>MỞ ĐẦU.....</b>	<b>13</b>
Chương 1: Khởi đầu dự án .....	1
1. Giới thiệu đề tài .....	1
2. Lý do chọn đề tài.....	1
3. Ý nghĩa và vai trò của hệ thống.....	1
4. Tính quan trọng và cấp thiết.....	2
5. Mục tiêu .....	2
6. Phạm vi.....	2
7. Hướng giải quyết.....	2
8. Đối tượng và phạm vi áp dụng.....	3
9. Phương pháp nghiên cứu .....	4
Chương 2: Cơ sở lý thuyết.....	5
1. Kiến trúc Microservices.....	5
1.1. Khả năng triển khai độc lập: .....	5
1.2. Tính kết nối lồng lèo: .....	7
2. Spring Boot, RESTful API và MySQL.....	10
2.1. Spring Boot: .....	10
2.2. RESTful API:.....	10
2.3. MySQL:.....	11
Ưu điểm của MySQL.....	11
Nhược điểm của MySQL.....	11
3. RabbitMQ hoặc Kafka.....	12
4. Eureka – Service Discovery.....	12
5. Spring Cloud Gateway .....	12
Chương 3: Phân tích và thiết kế hệ thống .....	13
1. <b>Phân tích nghiệp vụ và thiết kế .....</b>	<b>13</b>
1.1 <b>Khảo sát hiện trạng .....</b>	<b>13</b>
Tính phí cố định hàng tháng .....	15
1.2 <b>Yêu cầu hệ thống.....</b>	<b>16</b>
Yêu cầu chức năng.....	16
<b>Chức năng của Ban Quản Lý (BQL).....</b>	<b>16</b>

# Báo cáo Thực tập Tốt nghiệp Đại học

<b>Chức năng của Cư dân</b> .....	17
<b>Chức năng hệ thống</b> .....	17
Yêu cầu phi chức năng .....	17
<b>1.3 Đổi tượng người dùng &amp; Chức năng chi tiết</b> .....	18
a. Đổi tượng người dùng .....	18
<b>Ban quản lý chung cư (BQL)</b> .....	18
<b>Ngữ cảnh và vai trò</b> .....	18
<b>Cách đại diện</b> .....	18
<b>Lý do giữ 1 role ban đầu</b> .....	18
<b>Cư dân</b> .....	18
<b>Cách đại diện</b> .....	18
<b>Vai trò và quyền</b> .....	18
b. Lợi ích của phương án "1 role ban đầu" .....	19
c. Chức năng chi tiết .....	19
<b>Chức năng dành cho ban quản lý</b> .....	19
<b>Chức năng dành cho cư dân</b> .....	19
<b>2 Mô hình Use Case</b> .....	19
<b>Nhóm 0: Đặc biệt:</b> .....	19
Nhóm 1: Quản lý tài khoản cư dân .....	22
Nhóm 2: Quản lý dịch vụ .....	26
Nhóm 3: Quản lý mẫu hợp đồng .....	32
Nhóm 4: Quản lý hợp đồng cư dân .....	39
Nhóm 5: Quản lý hóa đơn và chỉ số điện/nước .....	49
Nhóm 6: Quản lý tài khoản cá nhân (Cư dân) .....	53
Nhóm 7: Đăng ký dịch vụ hợp đồng .....	55
Nhóm 8: Tra cứu hợp đồng .....	58
Nhóm 9: Thanh toán .....	62
Nhóm 10: Nhận thông báo .....	64
<b>3 Sơ đồ hoạt động (Activity diagram)</b> .....	67
<b>Nhóm 0: Đặc biệt:</b> .....	67
<b>Nhóm 1: Quản lý tài khoản cư dân</b> .....	68
<b>Nhóm 2: Quản lý dịch vụ</b> .....	69
<b>Nhóm 3: Quản lý mẫu hợp đồng</b> .....	71
<b>Nhóm 4: Quản lý hợp đồng cư dân</b> .....	73
<b>Nhóm 5: Quản lý hóa đơn và chỉ số điện/nước</b> .....	76
<b>Nhóm 6: Quản lý tài khoản cá nhân (Cư dân)</b> .....	78
<b>Nhóm 7: Đăng ký dịch vụ hợp đồng</b> .....	80

# Báo cáo Thực tập Tốt nghiệp Đại học

<b>Nhóm 8: Tra cứu hợp đồng</b> .....	82
<b>Nhóm 9: Thanh toán online</b> .....	83
<b>Nhóm 10: Nhận thông báo</b> .....	85
<b>4 Sơ đồ tuần tự (Sequence diagram)</b> .....	88
<b>Nhóm 0: Đặc biệt:</b> .....	88
<b>Nhóm 1: Quản lý tài khoản cư dân</b> .....	89
Nhóm 2. Quản lý dịch vụ.....	91
Nhóm 3. Quản lý mẫu hợp đồng.....	93
Nhóm 4. Quản lý hợp đồng cư dân .....	96
Nhóm 5. Quản lý hóa đơn & chỉ số.....	100
<b>Nhóm 6: Quản lý tài khoản cá nhân (Cư dân)</b> .....	102
<b>Nhóm 7: Đăng ký dịch vụ hợp đồng</b> .....	103
<b>Nhóm 8: Tra cứu hợp đồng</b> .....	105
<b>Nhóm 9: Thanh toán online</b> .....	106
<b>Nhóm 10: Nhận thông báo</b> .....	108
<b>5 Thiết kế kiến trúc hệ thống tổng thể (Diagram Microservices)</b> .....	111
<b>5.1 Sơ đồ thực thể (ERD)</b> .....	112
Dịch vụ Quản lý Người dùng (User Management Service).....	112
Dịch vụ Quản lý Phụ lục Dịch vụ (Service Addendum Service) .....	112
Dịch vụ Quản lý Hợp đồng (Contract Service) .....	113
Dịch vụ Lập hóa đơn & Thanh toán (Billing & Payment Service) .....	113
Dịch vụ Thông báo (Notification Service) .....	114
<b>5.2 Sơ đồ chuyển đổi trạng thái</b> .....	114
. Dịch vụ Quản lý Người dùng (User Management Service) .....	114
Bảng TaiKhoan .....	114
Bảng NhanVienBQL .....	116
Bảng CuDan.....	117
2. Dịch vụ Quản lý Hợp đồng (Contract Service).....	118
Bảng HopDong .....	118
Bảng YeuCauThayDoiHopDong .....	120
3. Dịch vụ Quản lý Phụ lục Dịch vụ (Service Addendum Service) .....	121
Bảng DichVu .....	121
Bảng GoiCuocDichVu.....	122
Bảng PhuLucDichVu .....	123
4. Dịch vụ Lập hóa đơn & Thanh toán (Billing & Payment Service) .....	125
Bảng HoaDon .....	125
Bảng LichSuThanhToan .....	126

<b>5.3 Thiết kế cơ sở dữ liệu .....</b>	126
Dịch vụ Quản lý Người dùng (User Management Service).....	126
Dịch vụ Quản lý Hợp đồng (Contract Service) .....	130
Dịch vụ Quản lý Phụ lục Dịch vụ (Service Addendum Service).....	131
Dịch vụ Lập hóa đơn & Thanh toán (Billing & Payment Service) .....	136
Dịch vụ Thông báo (Notification Service).....	139
<b>Chương 4: Cài đặt và triển khai hệ thống .....</b>	141
<b>1 Mô tả chi tiết các Microservices.....</b>	141
<b>1.1. Tổng quan về kiến trúc Microservices.....</b>	141
<b>1.2. Các thành phần trong hệ thống Microservices .....</b>	141
<b>1.3. Spring Boot và RESTful APIs cho mỗi vi dịch vụ.....</b>	141
<b>1.4. MySQL cho mỗi vi dịch vụ.....</b>	142
<b>1.5. Eureka cho Discovery Service .....</b>	142
<b>1.6. Spring Cloud Gateway cho điểm truy cập duy nhất.....</b>	142
<b>2. Triển khai Backend (Spring Boot, Cấu trúc Code, Các API Chính).....</b>	142
<b>2.1. Môi trường phát triển và công cụ sử dụng.....</b>	142
<b>2.2. Cấu trúc dự án Spring Boot .....</b>	142
<b>2.3. Cấu hình cơ sở dữ liệu MySQL trong Spring Boot.....</b>	143
• <b>Cấu hình MySQL trong application.properties:.....</b>	143
• <b>Entity Classes:</b> Trong Spring Boot, các bảng cơ sở dữ liệu sẽ được đại diện bởi các lớp Entity. Ví dụ, lớp User sẽ tương ứng với bảng người dùng trong cơ sở dữ liệu.....	143
• <b>Repository Layer:</b> Để thực hiện các thao tác CRUD với cơ sở dữ liệu, bạn sẽ sử dụng Spring Data JPA để tạo các repository.....	144
<b>2.4. Các API chính.....</b>	144
API của User Service .....	144
<b>1. Quản lý Xác thực (AuthController) .....</b>	144
<b>1.1. Đăng nhập.....</b>	144
<b>2. Quản lý Cư dân (CuDanController) .....</b>	145
<b>2.1. Lấy thông tin cư dân và căn hộ.....</b>	145
<b>2.2. Cập nhật thông tin cư dân .....</b>	145
<b>2.3. Lấy danh sách cư dân .....</b>	146
<b>3. Đăng ký (DangKyController).....</b>	146
<b>3.1. Đăng ký tài khoản cư dân .....</b>	146
<b>3.2. Lấy danh sách căn hộ.....</b>	147
<b>4. Quản lý Tài khoản (TaiKhoanController) .....</b>	147
<b>4.1. Duyệt/Hủy duyệt tài khoản.....</b>	147
<b>4.2. Khóa/Mở khóa tài khoản.....</b>	148

# Báo cáo Thực tập Tốt nghiệp Đại học

API của Service Addendum Service .....	148
<b>1. MauHopDongDichVuController</b> .....	148
<b>2. GoiCuocDichVuController</b> .....	149
<b>3. CauHinhDichVuController</b> .....	150
<b>4. MauHopDongController</b> .....	151
<b>5. PhuLucDichVuController</b> .....	152
<b>6. ThamSoPhiController</b> .....	152
Chương 5: Kết Luận .....	154
<b>5.1 Kết quả đạt được</b> .....	154
<b>5.2 Lợi ích đối với ban quản lý và cư dân</b> .....	154
<b>5.3 Hạn chế còn tồn tại</b> .....	154
<b>5.4 Hướng phát triển, nâng cấp trong tương lai</b> .....	155
Tài liệu tham khảo.....	156
Bảng phân công nhiệm vụ.....	157

## **DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH**

### **Hình**

Figure 1: Ảnh mô tả kiến trúc Microservice .....	5
Figure 2: Ảnh định nghĩa cách 1 microservice được coi là độc lập .....	6
Figure 3: Ảnh minh họa cách liên kết thời gian chạy làm giảm tính sẵn sàng của service .....	7
Figure 4: Ảnh minh họa về Liên kết thời gian chạy (Runtime Coupling) trong quá trình tạo đơn hàng	8
Figure 5: Usecase Đăng nhập.....	20
Figure 6: Usecase Quản lý tài khoản cư dân .....	22
Figure 7: UC Quản lý dịch vụ .....	26
Figure 8: UC Quản lý mẫu hợp đồng .....	32
Figure 9: UC Quản lý hợp đồng cư dân .....	39
Figure 10: Quản lý hóa đơn và chỉ số điện/nước .....	49
Figure 11: Xác nhận thanh toán .....	52
Figure 12: UC đăng ký tài khoản .....	53
Figure 13: UC Cập nhật thông tin cá nhân.....	54
Figure 14: Use Case: Lập hợp đồng.....	55
Figure 15: Xem điều khoản dịch vụ .....	56
Figure 16: Thay đổi dịch vụ hợp đồng .....	57
Figure 17: Tra cứu hợp đồng.....	59
Figure 18: UC Thanh toán.....	62
Figure 19: UC Nhận thông báo .....	64
Figure 20: AD Ban quản lý đăng nhập Hệ thống .....	68
Figure 21: Duyệt tài khoản.....	68
Figure 22: AD Khóa / Mở khóa .....	69
Figure 23: AD Khai báo dịch vụ mới.....	69
Figure 24: AD Quản lý Cấu hình Chi tiết Dịch vụ .....	70
Figure 25: AD Chính sửa Thông tin & Trạng thái Dịch vụ đã khai báo .....	71
Figure 26: AD Tạo mẫu hợp đồng dịch vụ.....	72
Figure 27: AD Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ .....	72
Figure 28: AD Chính sửa mẫu .....	73
Figure 29 AD Tự động chấm dứt do vi phạm .....	74
Figure 30 AD BQL chấm dứt do vi phạm nội quy.....	75
Figure 31 AD BQL đơn phương chấm dứt một dịch vụ riêng lẻ .....	75
Figure 32 AD BQL Duyệt Hợp đồng Dịch vụ mới .....	76
Figure 33 AD BQL Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng .....	76
Figure 34 AD Tự động lập hóa đơn .....	77
Figure 35 AD Ghi chỉ số điện/nước .....	78
Figure 36 AD Xác nhận thanh toán.....	78
Figure 37 AD Đăng ký tài khoản .....	79
Figure 38 AD Cập nhật thông tin cá nhân.....	80
Figure 39 AD Lập hợp đồng .....	81
Figure 40 AD Xem điều khoản dịch vụ .....	82
Figure 41 AD Thay đổi dịch vụ hợp đồng .....	82
Figure 42 AD Xem danh sách hợp đồng đã ký .....	83
Figure 43 AD Xem trạng thái thanh toán và thời hạn hợp đồng .....	83
Figure 44 AD Thanh toán thủ công (chuyển khoản, tiền mặt).....	84
Figure 45 AD Thanh toán online qua VNPay .....	85
Figure 46 AD Cư dân nhận nhắc nhở thời hạn hợp đồng.....	86
Figure 47 AD Cư dân nhận thông báo khi có hóa đơn mới .....	87

# Báo cáo Thực tập Tốt nghiệp Đại học

Figure 48 AD Cư dân nhận thông báo khi sắp hết hạn dịch vụ.....	88
Figure 49 AD Cư dân nhận và xem thông báo qua email & trên web .....	88
Figure 50 SD Ban quản lý đăng nhập Hệ thống.....	89
Figure 51 SD Xem danh sách cư dân đã đăng ký .....	89
Figure 52 SD Duyệt tài khoản.....	90
Figure 53 SD Khóa / Mở khóa .....	91
Figure 54 SD Khai báo dịch vụ mới.....	91
Figure 55 SD Quản lý Cấu hình Chi tiết Dịch vụ .....	92
Figure 56 SD Chính sửa Thông tin & Trạng thái Dịch vụ đã khai báo.....	93
Figure 57 SD Xem danh sách dịch vụ đã khai báo .....	93
Figure 58 SD Tạo mẫu hợp đồng dịch vụ .....	94
Figure 59 SD Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ.....	95
Figure 60 SD Xem danh sách mẫu hợp đồng.....	96
Figure 61 SD Chính sửa mẫu .....	96
Figure 62 SD Xem danh sách hợp đồng cá nhân .....	97
Figure 63 SD Xem chi tiết hợp đồng cụ thể .....	97
Figure 64 SD Tự động chấm dứt do vi phạm.....	98
Figure 65 SD BQL chấm dứt do vi phạm nội quy .....	98
Figure 66 SD BQL đơn phương chấm dứt một dịch vụ riêng lẻ .....	99
Figure 67 SD BQL Duyệt Hợp đồng Dịch vụ mới.....	99
Figure 68 SD BQL Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng .....	100
Figure 69 SD Tự động lập hóa đơn .....	100
Figure 70 SD Ghi chỉ số điện/nước .....	101
Figure 71 SD Xác nhận thanh toán .....	102
Figure 72 SD Đăng ký tài khoản .....	103
Figure 73 SD Cập nhật thông tin cá nhân .....	103
Figure 74 SD Lập hợp đồng .....	104
Figure 75 SD Xem điều khoản dịch vụ .....	104
Figure 76 SD Thay đổi dịch vụ hợp đồng .....	105
Figure 77 SD Tra cứu hợp đồng.....	106
Figure 78 SD Thanh toán thủ công (chuyển khoản, tiền mặt).....	107
Figure 79 SD Thanh toán online qua VNPAY .....	108
Figure 80 SD Cư dân nhận nhắc nhở thời hạn hợp đồng .....	109
Figure 81 SD Cư dân nhận thông báo khi có hóa đơn mới .....	110
Figure 82 SD Cư dân nhận thông báo khi sắp hết hạn dịch vụ .....	110
Figure 83 SD Cư dân nhận và xem thông báo qua email & trên web .....	111
Figure 84 kiến trúc hệ thống tổng thể .....	112
Figure 85 Dịch vụ Quản lý Người dùng.....	112
Figure 86 Dịch vụ Quản lý Phụ lục Dịch vụ .....	113
Figure 87 Dịch vụ Quản lý Hợp đồng .....	113
Figure 88 Dịch vụ Lập hóa đơn & Thanh toán .....	114
Figure 89 Dịch vụ Thông báo .....	114
Figure 90 Sơ đồ chuyển đổi trạng thái tài khoản.....	115
Figure 91 chuyển đổi trạng thái BQL.....	116
Figure 92 chuyển đổi trạng thái cư dân .....	117
Figure 93 chuyển đổi trạng thái hợp đồng .....	119
Figure 94 chuyển đổi trạng thái yêu cầu thay đổi .....	120
Figure 95 chuyển đổi trạng thái dịch vụ .....	121
Figure 96 chuyển đổi trạng thái gói cước.....	122
Figure 97 chuyển đổi trạng thái phụ lục dịch vụ .....	124

# Báo cáo Thực tập Tốt nghiệp Đại học

Figure 98 chuyển đổi trạng thái hóa đơn.....	125
Figure 99 chuyển đổi trạng thái lịch sử thanh toán .....	126

## Bảng

Table 1: Đặc tả usecase Đăng nhập.....	20
Table 2: Đặc tả usecase Xem danh sách tài khoản cư dân .....	22
Table 3: Đặc tả UC Khóa / Mở khóa tài khoản cư dân .....	25
Table 4: Đặc tả UC Khai báo dịch vụ mới .....	26
Table 5: Đặc tả Cấu hình Chi tiết Dịch vụ .....	28
Table 6: Đặc tả Chính sửa Thông tin & Trạng thái Dịch vụ đã khai báo.....	29
Table 7: Đặc tả Xem danh sách dịch vụ đã khai báo .....	31
Table 8: Đặc tả Tạo mẫu hợp đồng dịch vụ .....	33
Table 9: Đặc tả Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ.....	34
Table 10: Đặc tả Xem danh sách mẫu hợp đồng đã tạo .....	36
Table 11: Chính sửa mẫu hợp đồng .....	37
Table 12: Đặc tả Xem danh sách hợp đồng cá nhân .....	39
Table 13: Đặc tả Xem chi tiết hợp đồng cụ thể.....	40
Table 14: Đặc tả Tự động chấm dứt hợp đồng do vi phạm thanh toán .....	41
Table 15: BQL chấm dứt hợp đồng do vi phạm nội quy.....	42
Table 16: Đặc tả BQL đơn phương chấm dứt một dịch vụ riêng lẻ .....	43
Table 17: Đặc tả Duyệt Hợp đồng Dịch vụ mới.....	45
Table 18: Đặc tả Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng .....	47
Table 19: Đặc tả Theo dõi lập hóa đơn tự động hàng tháng .....	49
Table 20: Đặc tả Ghi chỉ số điện/nước thủ công .....	51
Table 21: Đặc tả Xác nhận thanh toán .....	52
Table 22: Đặc tả UC đăng ký tài khoản .....	53
Table 23: Đặc tả UC cập nhật thông tin cá nhân.....	54
Table 24: Đặc tả UC lập hợp đồng.....	55
Table 25: Đặc tả UC Xem điều khoản dịch vụ.....	56
Table 26: Thay đổi dịch vụ hợp đồng .....	57
Table 27: Đặc tả Xem danh sách hợp đồng đã ký .....	59
Table 28: Đặc tả Xem trạng thái thanh toán và thời hạn hợp đồng.....	60
Table 29 Đặc tả Xem trạng thái thanh toán và thời hạn hợp đồng .....	60
Table 30: Đặc tả Thanh toán thủ công (chuyển khoản, tiền mặt).....	62
Table 31: Đặc tả Thanh toán online qua VNPAY .....	63
Table 32: Đặc tả Cư dân nhận nhắc nhở thời hạn hợp đồng .....	64
Table 33: Đặc tả Cư dân nhận thông báo khi có hóa đơn mới .....	65
Table 34: Đặc tả Cư dân nhận thông báo khi sắp hết hạn dịch vụ .....	66
Table 35: Đặc tả Cư dân nhận và xem thông báo qua email & trên web .....	67
Table 36 Chuyển đổi trạng thái tài khoản .....	115
Table 37: chuyển đổi trạng thái BQL.....	116
Table 38: chuyển đổi trạng thái cư dân .....	118
Table 39 chuyển đổi trạng thái hợp đồng .....	119
Table 40 : chuyển đổi trạng thái yêu cầu thay đổi.....	120
Table 41 chuyển đổi trạng thái dịch vụ .....	122
Table 42 chuyển đổi trạng thái gói cước .....	123
Table 43 chuyển đổi trạng thái phục lục dịch vụ .....	124
Table 44 chuyển đổi trạng thái hóa đơn .....	125
Table 45 chuyển đổi trạng thái lịch sử thanh toán .....	126

## Báo cáo Thực tập Tốt nghiệp Đại học

Table 46 Bảng: TaiKhoan .....	127
Table 47 Bảng: CanHo .....	127
Table 48 Bảng: CuDan .....	128
Table 49 Bảng: NhanVienBQL .....	129
Table 50 Bảng: HopDong .....	130
Table 51 Bảng: YeuCauThayDoiHopDong .....	131
Table 52 Bảng: MauHopDong .....	131
Table 53 Bảng: MauHopDongDIchVu .....	132
Table 54 Bảng: DichVu .....	132
Table 55 Bảng: CauHinhDichVu .....	133
Table 56 Bảng: ThamSoPhi .....	134
Table 57 Bảng: GoiCuocDichVu .....	134
Table 58 Bảng: PhuLucDichVu .....	135
Table 59 Bảng: HoaDon .....	136
Table 60 Bảng: ChiTietHoaDon .....	137
Table 61 Bảng: ChiSoDichVu .....	138
Table 62 Bảng: LichSuThanhToan .....	139
Table 63 Bảng: ThongBao .....	139

## MỞ ĐẦU

Trong bối cảnh công nghệ thông tin ngày càng phát triển mạnh mẽ, các giải pháp phần mềm ứng dụng vào quản lý dịch vụ và hợp đồng trở thành một phần không thể thiếu trong việc nâng cao hiệu quả hoạt động của các tổ chức và doanh nghiệp. Đặc biệt, trong các môi trường quản lý cư dân và các dịch vụ tiện ích, việc xây dựng một hệ thống quản lý hợp đồng và dịch vụ không chỉ giúp tiết kiệm thời gian mà còn đảm bảo tính chính xác, minh bạch trong việc theo dõi và xử lý thông tin.

Với mục tiêu đáp ứng nhu cầu quản lý hiệu quả các hợp đồng dịch vụ cho cư dân và ban quản lý, đề tài "Xây dựng hệ thống quản lý hợp đồng và dịch vụ cho cư dân và ban quản lý" được thực hiện. Hệ thống được thiết kế theo kiến trúc microservices, sử dụng công nghệ Spring Boot, MySQL và React.js, nhằm đảm bảo tính linh hoạt, mở rộng và dễ dàng bảo trì. Hệ thống không chỉ cung cấp các chức năng quản lý hợp đồng và dịch vụ cho cư dân mà còn giúp ban quản lý thực hiện các công việc như cấp tài khoản, duyệt hợp đồng, tạo hóa đơn, và thanh toán một cách tự động.

Báo cáo này trình bày quá trình thiết kế, triển khai và đánh giá hiệu quả của hệ thống. Mục tiêu của nghiên cứu là cung cấp một giải pháp toàn diện giúp tối ưu hóa công tác quản lý và nâng cao sự hài lòng của cư dân thông qua việc ứng dụng công nghệ vào các hoạt động quản lý.

Trong quá trình thực hiện đề tài, chúng em đã nhận được sự hỗ trợ quý báu từ các giảng viên, đồng nghiệp và gia đình. Chúng em hy vọng rằng, với những kết quả đạt được, hệ thống này sẽ đóng góp vào việc cải tiến các giải pháp quản lý trong các môi trường tương tự và mang lại lợi ích cho cộng đồng sử dụng.

## Chương 1: Khởi đầu dự án

### 1. Giới thiệu đề tài

Trong bối cảnh đô thị hóa diễn ra mạnh mẽ, các khu chung cư cao tầng ngày càng xuất hiện nhiều tại các thành phố lớn nhằm đáp ứng nhu cầu nhà ở ngày càng tăng của người dân. Song song với đó, việc quản lý vận hành và cung cấp các dịch vụ cho cư dân trở thành một thách thức lớn đối với ban quản lý.

Quản lý cư dân, hợp đồng dịch vụ, hóa đơn thanh toán, bảo trì, vệ sinh, bảo vệ,... là những nghiệp vụ phức tạp, đòi hỏi tính chính xác, kịp thời và minh bạch cao. Tuy nhiên, nhiều chung cư hiện nay vẫn đang sử dụng các phương pháp quản lý thủ công hoặc phần mềm tập trung với kiến trúc monolithic, dẫn đến các hạn chế như:

- + Dễ xảy ra sai sót, thất thoát thông tin.
- + Khó mở rộng, khó tích hợp thêm dịch vụ mới.
- + Thiếu tính tự động hóa, ảnh hưởng đến trải nghiệm của cư dân.

Để giải quyết những hạn chế đó, đề tài "Xây dựng hệ thống quản lý hợp đồng dịch vụ tại chung cư bằng kiến trúc Microservices" được lựa chọn. Đề tài hướng tới việc xây dựng một hệ thống phần mềm hiện đại, linh hoạt, giúp nâng cao hiệu quả quản lý, tăng tính minh bạch và mang lại sự tiện lợi cho cư dân.

### 2. Lý do chọn đề tài

Việc áp dụng kiến trúc Microservices cho hệ thống quản lý chung cư giúp giải quyết triệt để những vấn đề tồn tại trong các mô hình cũ, đồng thời bắt kịp xu hướng chuyển đổi số. Các lý do chính để lựa chọn đề tài này gồm:

- + Tính cấp thiết và thực tế cao: Đáp ứng nhu cầu quản lý ngày càng phức tạp tại các khu chung cư lớn.
- + Nâng cao chất lượng dịch vụ: Giúp ban quản lý dễ dàng kiểm soát thông tin cư dân, hợp đồng, dịch vụ, hóa đơn và các cảnh báo liên quan.
- + Tối ưu trải nghiệm cư dân: Cho phép cư dân tự quản lý hợp đồng, nhận thông báo tự động và thực hiện thanh toán online.
- + Khả năng mở rộng và bảo trì dễ dàng: Nhờ kiến trúc Microservices, từng chức năng có thể phát triển, triển khai và mở rộng độc lập mà không ảnh hưởng đến toàn hệ thống.

### 3. Ý nghĩa và vai trò của hệ thống

Hệ thống đóng vai trò then chốt trong hoạt động nghiệp vụ của ban quản lý chung cư:

- + Giảm tải công việc thủ công: Các quy trình như lập hợp đồng, xuất hóa đơn, gửi cảnh báo và quản lý dịch vụ đều được tự động hóa.
- + Đảm bảo minh bạch: Thông tin dịch vụ, chi phí và hợp đồng được lưu trữ đầy đủ, rõ ràng và dễ tra cứu.
- + Tăng tính kết nối và tương tác: Ban quản lý và cư dân có thể tương tác trực tuyến qua hệ thống, giảm thiểu khâu trung gian và giấy tờ.
- + Cung cấp dữ liệu hỗ trợ ra quyết định: Hệ thống lưu trữ và phân tích dữ liệu giúp ban quản lý đánh giá hiệu quả vận hành và cải thiện dịch vụ.

## 4. Tính quan trọng và cấp thiết

Trong thực tế, nếu không có một hệ thống quản lý hợp đồng dịch vụ chuyên nghiệp:

- + Ban quản lý sẽ gặp khó khăn trong việc kiểm soát số lượng lớn cư dân, dịch vụ và hóa đơn, dễ dẫn đến thất thoát, tranh chấp và giảm uy tín.
- + Cư dân sẽ thiếu thông tin minh bạch, khó nắm bắt tình trạng hợp đồng, không được nhắc nhở kịp thời khi đến hạn thanh toán hoặc hết hạn dịch vụ, gây ra bất tiện và phiền hà.
- + Hệ thống quản lý lỗi thời không thể đáp ứng kịp nhu cầu phát triển, mở rộng hoặc tích hợp các công nghệ, dịch vụ mới trong tương lai.

Do đó, việc triển khai một hệ thống quản lý hợp đồng dịch vụ tại chung cư dựa trên kiến trúc Microservices là giải pháp tối ưu, cấp thiết và mang tính chiến lược nhằm nâng cao hiệu quả quản lý, đảm bảo quyền lợi cư dân và đáp ứng yêu cầu phát triển bền vững.

## 5. Mục tiêu

Mục tiêu chính của đề tài là xây dựng một hệ thống quản lý hợp đồng dịch vụ tại chung cư dựa trên kiến trúc Microservices, với các yêu cầu cụ thể như sau:

- + Tự động hóa quy trình quản lý hợp đồng: Từ khâu định nghĩa dịch vụ, đăng ký, phê duyệt, đến lập hóa đơn và thanh toán.
- + Tạo điều kiện cho cư dân tự quản lý thông tin: Cho phép cư dân đăng ký, tra cứu, cập nhật hợp đồng, nhận cảnh báo và thực hiện thanh toán online một cách thuận tiện.
- + Đảm bảo khả năng mở rộng và tích hợp: Thiết kế hệ thống linh hoạt, có thể dễ dàng bổ sung hoặc thay đổi dịch vụ, tích hợp thêm các module mới hoặc các hệ thống thanh toán điện tử.
- + Nâng cao hiệu quả quản lý: Hỗ trợ ban quản lý giảm tải công việc thủ công, kiểm soát chính xác thông tin, giảm rủi ro sai sót và tranh chấp.

## 6. Phạm vi

Phạm vi triển khai của đề tài tập trung vào các chức năng nghiệp vụ chính và không đi quá sâu vào các yếu tố bên ngoài (ví dụ như hạ tầng phần cứng, chi tiết bảo mật nâng cao, hoặc các công thanh toán thực tế). Cụ thể:

- + Đối với ban quản lý: Quản lý cư dân, định nghĩa dịch vụ, duyệt/hủy tài khoản, lập hóa đơn và quản lý thanh toán.
- + Đối với cư dân: Đăng ký, quản lý tài khoản, lập và tra cứu hợp đồng, cập nhật dịch vụ, nhận thông báo tự động.
- + Về hình thức thanh toán: Hệ thống hỗ trợ hai hình thức thanh toán gồm thủ công (cư dân nộp tiền mặt hoặc chuyển khoản và được ban quản lý xác nhận trên hệ thống) và thanh toán online thông qua công điện tử. Trong khuôn khổ đề tài thực tập, thanh toán online sẽ được mô phỏng bằng cách tích hợp môi trường sandbox của VNPay, giúp kiểm tra luồng thanh toán thật nhưng không phát sinh giao dịch thực tế.
- + Về mặt kỹ thuật: Xây dựng backend Microservices bằng Spring Boot, RESTful APIs, MySQL; giao tiếp bắt đồng bộ bằng RabbitMQ hoặc Kafka; sử dụng Eureka và Spring Cloud Gateway; phát triển giao diện web cho ban quản lý và cư dân.

## 7. Hướng giải quyết

Để đạt được các mục tiêu trên, đề tài sẽ triển khai các hướng giải quyết cụ thể:

- + Áp dụng kiến trúc Microservices: Tách hệ thống thành nhiều dịch vụ nhỏ, độc lập, dễ triển khai và bảo trì.

## Báo cáo Thực tập Tốt nghiệp Đại học

- + Sử dụng công nghệ Spring Boot: Tận dụng khả năng phát triển nhanh, dễ tích hợp và cộng đồng hỗ trợ mạnh mẽ.
- + Giao tiếp thông qua RESTful APIs: Đảm bảo chuẩn hóa, dễ kết nối giữa frontend và backend.
- + Triển khai giao tiếp bất đồng bộ: Sử dụng RabbitMQ hoặc Kafka giúp hệ thống xử lý luồng dữ liệu lớn, giảm độ trễ.
- + Đảm bảo khả năng mở rộng: Có thể thêm mới dịch vụ mà không ảnh hưởng đến các chức năng khác.
- + Phát triển giao diện web thân thiện: Hỗ trợ truy cập mọi lúc mọi nơi, tối ưu trải nghiệm người dùng cho cả ban quản lý và cư dân.

### 8. Đối tượng và phạm vi áp dụng

Hệ thống quản lý hợp đồng dịch vụ tại chung cư bằng kiến trúc Microservices được thiết kế để phục vụ hai nhóm đối tượng chính:

#### + Ban quản lý chung cư (BQL):

Là đơn vị trực tiếp điều hành, chịu trách nhiệm kiểm soát, phê duyệt và giám sát tất cả các hoạt động liên quan đến hợp đồng dịch vụ của cư dân.

Quyền và chức năng chính của BQL:

- + Duyệt, cấp hoặc hủy tài khoản cư dân.
- + Định nghĩa, cập nhật và quản lý danh mục dịch vụ (điện, nước, bảo trì, vệ sinh, gửi xe...).
- + Lập và xuất hóa đơn dịch vụ hàng tháng.
- + Quản lý tình trạng thanh toán, xử lý các vấn đề phát sinh.
- + Gửi thông báo tự động cho cư dân khi hợp đồng sắp hết hạn hoặc đến kỳ thanh toán.

Ràng buộc hoạt động của BQL:

- + Chỉ được duyệt tài khoản khi cư dân cung cấp thông tin hợp lệ.
- + Không được thay đổi nội dung hợp đồng của cư dân nếu không có yêu cầu chính thức.
- + Không can thiệp trực tiếp vào quá trình thanh toán online.

#### + Cư dân:

Là những người đang sinh sống và sử dụng dịch vụ tại chung cư.

Quyền và chức năng chính của cư dân:

- + Đăng ký tài khoản và chờ BQL phê duyệt.
- + Lập hợp đồng dịch vụ mới hoặc điều chỉnh hợp đồng đang sử dụng.
- + Tra cứu thông tin hợp đồng, hóa đơn, lịch sử thanh toán.
- + Nhận thông báo tự động từ hệ thống.
- + Thanh toán online các khoản phí dịch vụ.

Ràng buộc của cư dân:

- + Chỉ được quản lý hợp đồng gắn với phòng mà mình đang thuê hoặc sở hữu.
- + Khi đăng ký tài khoản, phải khai báo đầy đủ thông tin xác thực (CCCD, số điện thoại, email).
- + Chỉ được điều chỉnh dịch vụ trong phạm vi và thời gian quy định.

Về phạm vi áp dụng, hệ thống được tập trung cho các khu chung cư cao tầng tại các thành phố lớn, nơi có số lượng cư dân đông và nhu cầu dịch vụ đa dạng. Trong mô hình quản lý, mỗi phòng (căn hộ) được xem là đơn vị đại diện chính cho hợp đồng dịch vụ; cư dân (chủ hộ) là người đứng tên thực hiện giao dịch, thanh toán và nhận thông báo.

Các dịch vụ, hợp đồng, hóa đơn đều được gắn với phòng, nhưng trách nhiệm thanh toán và quyền điều chỉnh thuộc về cư dân đang cư trú tại phòng đó. Điều này giúp đảm bảo tính minh

# Báo cáo Thực tập Tốt nghiệp Đại học bách, dễ kiểm soát và phù hợp với nghiệp vụ thực tế.

Mối quan hệ ràng buộc giữa hai đối tượng:

- + BQL là bên cung cấp dịch vụ và kiểm soát thông tin, cư dân là bên sử dụng dịch vụ và thực hiện nghĩa vụ thanh toán.
  - + Cư dân chỉ được đăng ký, lập hoặc điều chỉnh hợp đồng sau khi BQL phê duyệt tài khoản.
  - + Mọi hóa đơn được BQL phát hành dựa trên dữ liệu hợp đồng mà cư dân đã xác nhận.
  - + BQL gửi thông báo, cư dân nhận và thực hiện các nghĩa vụ tương ứng.
  - + Cư dân không thể tự ý thay đổi điều khoản hợp đồng nếu chưa có sự đồng thuận từ BQL.
- Mối quan hệ này giúp đảm bảo minh bạch, hạn chế tranh chấp, đồng thời tăng cường tính chuyên nghiệp và trách nhiệm của cả hai bên.

## 9. Phương pháp nghiên cứu

Đề tài được thực hiện thông qua phương pháp nghiên cứu kết hợp, bao gồm:

- Nghiên cứu lý thuyết: Tìm hiểu các kiến trúc phần mềm hiện đại, đặc biệt là kiến trúc Microservices, cùng các công nghệ như Spring Boot, RESTful APIs, MySQL, RabbitMQ/Kafka, Eureka và Spring Cloud Gateway. Ngoài ra, nghiên cứu các giải pháp giao tiếp bất đồng bộ và phương pháp tích hợp công thanh toán điện tử (VNPay sandbox).
- Khảo sát thực tế: Tìm hiểu nghiệp vụ quản lý hợp đồng dịch vụ tại một số chung cư lớn, các vấn đề thường gặp trong quy trình lập hợp đồng, quản lý hóa đơn và thanh toán. Thu thập các yêu cầu thực tế từ ban quản lý và nhu cầu của cư dân.
- Phân tích và mô hình hóa: Phân tích nghiệp vụ, xác định các chức năng chính, thiết kế sơ đồ kiến trúc tổng thể và mô hình dữ liệu, từ đó chia nhỏ hệ thống thành các vi dịch vụ độc lập.
- Thủ nghiệm và phát triển: Xây dựng hệ thống demo, triển khai các chức năng chính, thử nghiệm quy trình lập hợp đồng, tạo hóa đơn và thanh toán (bao gồm mô phỏng thanh toán online). Đánh giá tính ổn định, khả năng mở rộng và mức độ đáp ứng yêu cầu nghiệp vụ.

Phương pháp tiếp cận này giúp đảm bảo hệ thống vừa đáp ứng được yêu cầu thực tiễn, vừa có tính khoa học, hiện đại và dễ dàng phát triển, mở rộng trong tương lai.

## Chương 2: Cơ sở lý thuyết

### 1. Kiến trúc Microservices

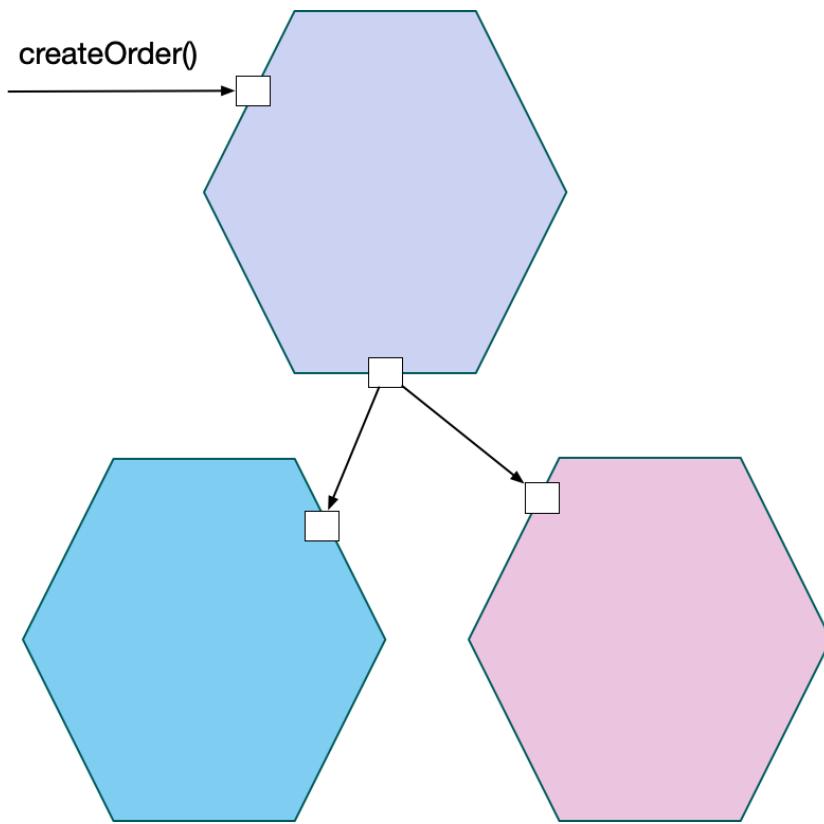


Figure 1: Ảnh mô tả kiến trúc Microservice

Microservices — còn được gọi là kiến trúc microservice — là một kiểu kiến trúc phần mềm cấu trúc một ứng dụng thành tập hợp gồm hai hoặc nhiều dịch vụ mà:

- + “Có thể triển khai độc lập”.
- + “Kết nối lồng léo”.

Các dịch vụ thường được tổ chức xoay quanh các năng lực nghiệp vụ. Mỗi dịch vụ thường được sở hữu bởi một nhóm nhỏ, riêng biệt.

#### 1.1. **Khả năng triển khai độc lập:**

##### 1.1.1. **Định nghĩa đơn giản về khả năng triển khai độc lập:**

Định nghĩa đơn giản của ‘independently deployable’ (khả năng triển khai độc lập) là một dịch vụ được đóng gói thành một đơn vị có thể triển khai hoặc có thể thực thi. Ví dụ về một đơn vị có thể triển khai hoặc thực thi bao gồm: tệp JAR có thể thực thi, tệp WAR, tệp thực thi của hệ điều hành, một image của Docker container, hoặc một tệp Zip định nghĩa một hàm AWS Lambda. Ngược lại, một tệp JAR thông thường cần được đóng gói cùng với các tệp JAR khác thì không phải là một đơn vị có thể triển khai hoặc thực thi. Mặc dù điều này đáp ứng định nghĩa theo đúng nghĩa đen của ‘independently deployable’, nó lại hoàn toàn không đủ.

##### 1.1.2. **Định nghĩa hoàn chỉnh về triển khai độc lập :**

# Báo cáo Thực tập Tốt nghiệp Đại học

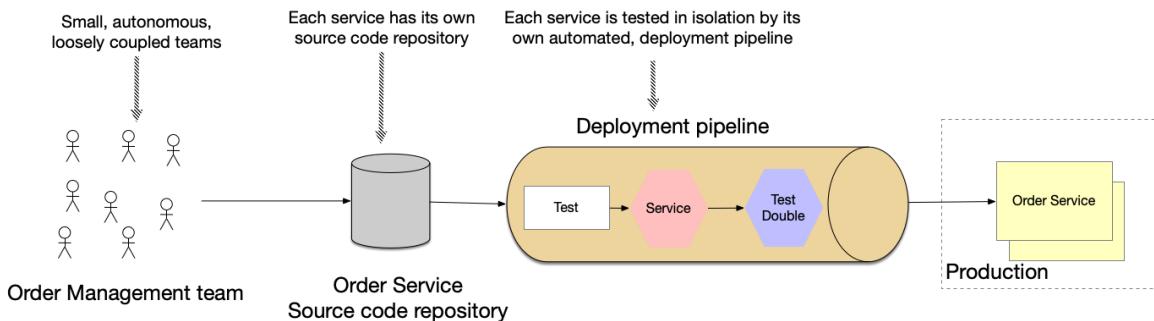


Figure 2: *Ảnh định nghĩa cách 1 microservice được coi là độc lập*

**Dịch vụ triển khai độc lập** là một dịch vụ được đóng gói dưới dạng một đơn vị có thể triển khai hoặc thực thi được, và sẵn sàng cho môi trường sản xuất sau khi đã được kiểm thử độc lập. Một dịch vụ như vậy sẽ có kho mã nguồn và quy trình triển khai riêng. Quy trình triển khai này sẽ kiểm thử dịch vụ một cách độc lập bằng cách sử dụng **test doubles** (đối tượng kiểm thử) cho các cộng tác viên của nó, cùng với kiểm thử hợp đồng hướng đến người tiêu dùng (consumer+driven contract testing). Kết quả cuối cùng từ quy trình triển khai là một dịch vụ có thể và nên được triển khai vào môi trường sản xuất.

Nếu chúng ta cần kiểm thử dịch vụ của mình với các dịch vụ khác để xác minh rằng nó đã sẵn sàng cho sản xuất, thì nó không phải là dịch vụ triển khai độc lập. Hơn nữa, chúng ta có thể muốn cân nhắc đặt các dịch vụ đó vào một kho lưu trữ duy nhất. Điều này đảm bảo rằng đầu ra của một quy trình triển khai duy nhất thực sự sẵn sàng cho sản xuất và cũng loại bỏ sự phức tạp của việc phát triển trên nhiều kho lưu trữ.

Một lợi ích quan trọng của dịch vụ triển khai độc lập là nó giúp đẩy nhanh quy trình triển khai. Nó loại bỏ sự cần thiết của các bài kiểm thử end-to-end (kiểm thử đầu cuối) chậm chạp, dễ hỏng và phức tạp của nhiều dịch vụ. Nó cũng loại bỏ sự cần thiết của việc các nhóm phải phối hợp và có khả năng cản trở lẫn nhau.

### 1.1.3. Vấn đề phát sinh:

Một trở ngại đối với việc triển khai các dịch vụ một cách độc lập chính là các bài kiểm thử chấp nhận người dùng (User Acceptance Tests + UAT) ở cấp độ hệ thống. Các bài kiểm thử chấp nhận này thường được viết từ góc nhìn của người dùng và thường bao trùm nhiều dịch vụ. Việc triển khai trực tiếp các bài kiểm thử như vậy sẽ đòi hỏi phải kiểm thử nhiều dịch vụ cùng lúc.

Để các dịch vụ có thể được triển khai độc lập, chúng ta phải thay thế các bài kiểm thử chấp nhận người dùng cấp hệ thống bằng các bài kiểm thử chấp nhận người dùng cấp độ dịch vụ.

### 1.1.4. Yêu cầu về đặc tả rõ ràng cho dịch vụ triển khai độc lập:

Để một dịch vụ có thể được kiểm thử độc lập, nó cần có hai yếu tố chính:

- + Một đặc tả (specification) được định nghĩa rõ ràng.
- + Một bộ kiểm thử (test suite) xác minh rằng hành vi của nó tuân thủ đặc tả đó.

Nếu chúng ta không thể đặc tả chính xác hành vi của một dịch vụ, có lẽ chúng ta nên xem xét lại liệu nó có hợp lý để tồn tại như một dịch vụ riêng biệt hay không.

Các dịch vụ cũng phải hợp tác thông qua các API ổn định để tránh việc phải thay đổi đồng loạt một cách thường xuyên. Đây là một khía cạnh của đặc điểm khác của kiến trúc microservice – kết nối lỏng lẻo (loosely coupled) – mà chúng em sẽ mô tả chi tiết hơn trong phần tiếp theo.

## 1.2. Tính kết nối lỏng lẻo:

Thực tế có hai loại liên kết (coupling) khác nhau:

+ Liên kết thời gian chạy (runtime coupling) + ảnh hưởng đến tính khả dụng (availability).

+ Liên kết thời gian thiết kế (design+time coupling) + ảnh hưởng đến tốc độ phát triển (development velocity).

Giảm thiểu liên kết thời gian thiết kế và giảm thiểu liên kết thời gian chạy là hai trong số năm "lực lượng vật chất tối" (dark matter forces) định hình kiến trúc microservice.

### 1.2.1. Liên kết thời gian chạy và tính khả dụng

Liên kết thời gian chạy (runtime coupling) giữa các dịch vụ là mức độ mà tính khả dụng (availability) của một dịch vụ bị ảnh hưởng bởi tính khả dụng của một dịch vụ khác. Hay nói chính xác hơn, đó là mức độ mà tính khả dụng của một hoạt động được triển khai bởi một dịch vụ bị ảnh hưởng bởi tính khả dụng của một dịch vụ khác.

### 1.2.2. Liên kết thời gian chạy làm giảm tính khả dụng

Ví dụ, hãy tưởng tượng rằng thao tác hệ thống createOrder() được triển khai bởi điểm cuối HTTP POST /orders trong Dịch vụ Đặt hàng (Order Service). Dịch vụ Đặt hàng xử lý yêu cầu HTTP POST bằng cách gọi các dịch vụ khác, chờ chúng phản hồi, rồi sau đó gửi phản hồi cho client của nó.

Trong thiết kế này, Dịch vụ Đặt hàng không thể phản hồi yêu cầu POST cho đến khi các dịch vụ khác phản hồi lại nó. Dịch vụ Đặt hàng (hoặc thao tác createOrder()) được gọi là có liên kết thời gian chạy với các dịch vụ khác đó. Kết quả là, tính khả dụng của thao tác createOrder() bị giảm đi vì tất cả các dịch vụ liên quan đều phải khả dụng.

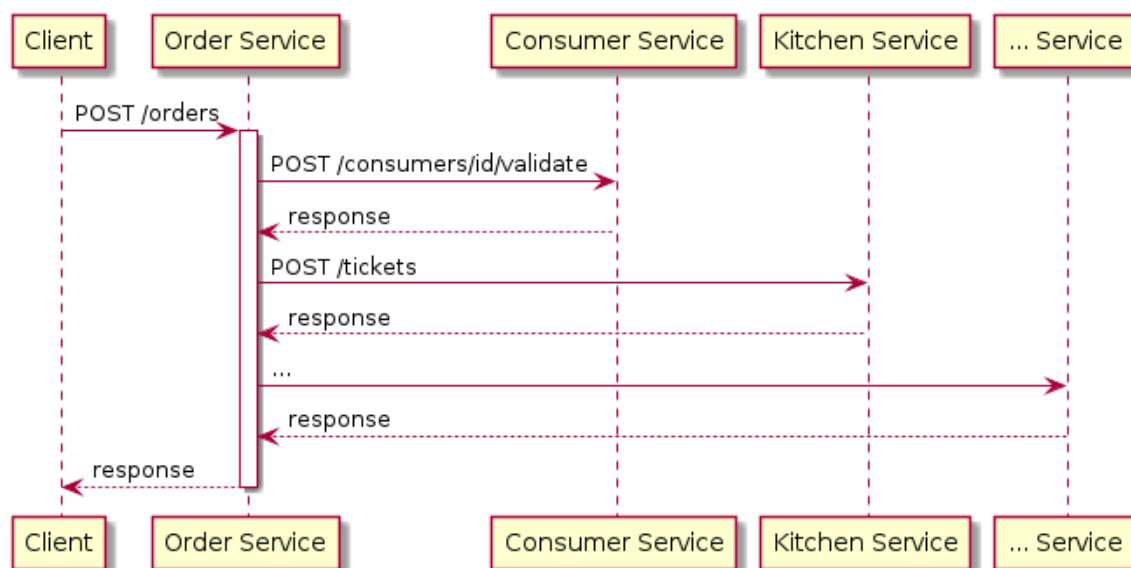


Figure 3: Ảnh minh họa cách liên kết thời gian chạy làm giảm tính sẵn sàng của service

### 1.2.3. Giảm thiểu liên kết thời gian chạy:

**Giảm thiểu liên kết thời gian chạy** là một trong những "lực hút của vật chất tối" chống lại việc phân tách. Một cách để giảm liên kết thời gian chạy của một thao tác là **giảm số lượng dịch vụ thực hiện nó**. Trên thực tế, chúng ta có thể loại bỏ hoàn toàn liên kết thời gian chạy bằng cách làm cho một thao tác trở thành cục bộ (local) đối với một dịch vụ duy nhất. Tuy nhiên, không phải lúc nào cũng có thể tạo ra một kiến trúc microservice mà tất cả các thao tác đều là cục bộ. Điều đó có thể vi phạm các "lực đẩy của năng lượng tối", vốn khuyến khích việc phân tách.

Cách khác để giảm liên kết thời gian chạy trong khi vẫn thỏa mãn các lực của năng lượng tối là **thiết kế các dịch vụ tự chứa (self-contained services)**. Một dịch vụ tự chứa sẽ phản hồi một yêu cầu đồng bộ với một kết quả cục bộ (partial outcome) và sau đó hoàn thành thao tác một cách bất đồng bộ. Ví dụ, Dịch vụ Đặt hàng (Order Service) có thể phản hồi yêu cầu HTTP POST /orders bằng một phản hồi 202 Accepted và sau đó bắt đầu một **"Create Order Saga"** để hoàn tất thao tác. Cách tiếp cận này cải thiện tính khả dụng của Dịch vụ Đặt hàng. Nhược điểm là nó làm cho client trở nên phức tạp hơn vì nó phải có khả năng xử lý các kết quả cục bộ và bằng cách nào đó xác định kết quả cuối cùng của thao tác.

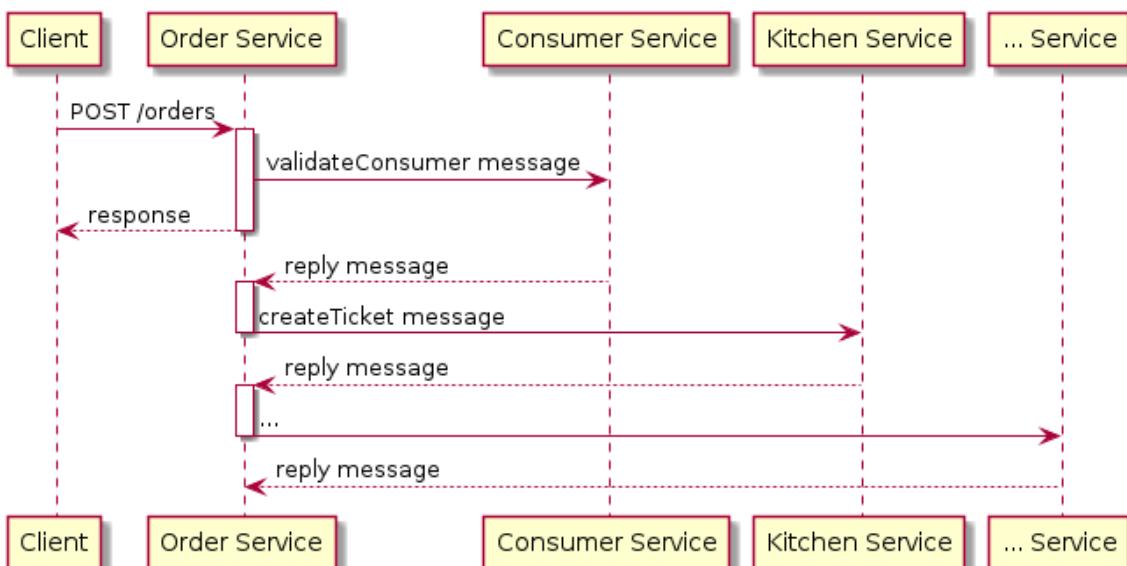


Figure 4: Ảnh minh họa về Liên kết thời gian chạy (Runtime Coupling) trong quá trình tạo đơn hàng

#### 1.2.4. Liên kết thời gian thiết kế và tốc độ phát triển:

Mức độ liên kết thời gian thiết kế (design+time coupling) giữa một cặp các phần tử phần mềm – từ các lớp (classes) cho đến các dịch vụ (services) – là khả năng chúng cần phải thay đổi cùng nhau vì cùng một lý do. Liên kết thời gian thiết kế giữa các dịch vụ trong kiến trúc microservice đặc biệt có vấn đề.

##### 1.2.4.1. Liên kết thời gian thiết kế làm giảm tốc độ phát triển

Nếu hai dịch vụ có liên kết lỏng lẻo (loosely coupled), thì một thay đổi ở dịch vụ này hiếm khi đòi hỏi một thay đổi ở dịch vụ kia. Tuy nhiên, nếu hai dịch vụ có liên kết chặt chẽ (tightly coupled), thì một thay đổi ở dịch vụ này thường đòi hỏi một thay đổi ở dịch vụ kia. Những loại thay đổi đồng bộ (lockstep changes) như vậy rất tốn kém vì chúng thường liên quan đến các thay đổi API gây hỏng (breaking API changes).

## Báo cáo Thực tập Tốt nghiệp Đại học

Ví dụ, hãy tưởng tượng rằng Dịch vụ Đặt hàng (Order Service) và Dịch vụ Khách hàng (Customer Service) có liên kết chặt chẽ. Mỗi khi cần thực hiện một thay đổi gây hỏng đổi với Dịch vụ Khách hàng, chuỗi các bước sẽ như sau:

- + Thay đổi Dịch vụ Khách hàng để thêm một phiên bản chính mới của API. Dịch vụ này phải triển khai cả phiên bản API cũ và mới cho đến khi tất cả các client đã được di chuyển sang.
- + Di chuyển Dịch vụ Đặt hàng sang phiên bản API mới.
- + Loại bỏ phiên bản API cũ khỏi Dịch vụ Khách hàng.

Điều tồi tệ hơn nữa là, khá thường xuyên các dịch vụ này thuộc sở hữu của các đội khác nhau, điều này đòi hỏi các đội đó phải phối hợp các thay đổi. Nói cách khác, liên kết thời gian thiết kế giữa các dịch vụ làm suy yếu tính tự chủ của đội (team autonomy).

### 1.2.4.2. Giảm thiểu liên kết thời gian thiết kế

Giảm thiểu liên kết thời gian thiết kế là một trong những "lực hút của vật chất tối" chống lại việc phân tách. Có một vài cách khác nhau để giảm thiểu liên kết thời gian thiết kế giữa các dịch vụ:

- + Thiết kế các phân miền (subdomain) có liên kết lỏng lẻo: Các phân miền có liên kết lỏng lẻo có thể được đóng gói thành các dịch vụ khác nhau. Liên kết thời gian thiết kế lỏng lẻo thường đạt được bằng cách mỗi phân miền có một API ổn định giúp đóng gói việc triển khai bên trong của nó.
- + Đóng gói các phân miền có liên kết chặt chẽ vào cùng một dịch vụ: Nếu hai phân miền có liên kết chặt chẽ với nhau, thì việc đóng gói chúng lại cùng trong một dịch vụ sẽ tránh được liên kết thời gian thiết kế giữa các dịch vụ.

### 1.2.5. Ưu điểm:

- Tăng khả năng mở rộng và linh hoạt cho hệ thống.
- Dễ dàng phát triển, bảo trì và triển khai từng phần riêng biệt.
- Cho phép sử dụng công nghệ khác nhau cho từng dịch vụ nếu cần.
- Cải thiện tính chịu lỗi, tránh ảnh hưởng toàn hệ thống khi một dịch vụ gặp sự cố.

### 1.2.6. Nhược điểm:

- Quản lý phức tạp hơn do nhiều service, nhiều cơ sở dữ liệu.
- Yêu cầu hệ thống giám sát, logging và cơ chế giao tiếp giữa các dịch vụ phức tạp hơn.
- Đòi hỏi thiết kế hạ tầng mạng, bảo mật, quản lý phiên bản tốt.

### 1.2.7. Mẫu thiết kế phổ biến:

- API Gateway Pattern: Là điểm truy cập duy nhất, hỗ trợ routing, authentication và load balancing.
- Service Discovery Pattern: Giúp các dịch vụ tự động tìm thấy nhau mà không cần cấu hình thủ công.
- Circuit Breaker Pattern: Ngăn chặn lỗi lan rộng khi dịch vụ downstream bị lỗi.
- Database per Service: Mỗi dịch vụ quản lý database riêng, tránh phụ thuộc lẫn nhau.

- Event-driven Pattern: Các dịch vụ giao tiếp bằng sự kiện (event), giúp tách biệt, giảm kết dính.

## 2. Spring Boot, RESTful API và MySQL

### 2.1. Spring Boot:

**Spring Boot** là một framework mã nguồn mở dựa trên nền tảng Spring Framework, được phát triển nhằm đơn giản hóa và tăng tốc quá trình phát triển ứng dụng Java. Nếu như Spring truyền thống yêu cầu nhiều bước cấu hình phức tạp, Spring Boot ra đời với mục tiêu giúp lập trình viên có thể tạo ứng dụng "**production-ready**" nhanh chóng, ít cấu hình nhất có thể.

Spring Boot nổi bật nhờ khả năng **tự động cấu hình (auto-configuration)**, nghĩa là framework tự nhận biết các thư viện và thành phần đã được thêm vào (dependencies) để cấu hình phù hợp. Bên cạnh đó, Spring Boot cung cấp các **starter dependencies** — những gói thư viện tổng hợp, được cấu hình sẵn, giúp lập trình viên chỉ cần khai báo ngắn gọn mà vẫn tích hợp đầy đủ các tính năng như Web, JPA, Security, v.v.

Một ưu điểm lớn của Spring Boot là hỗ trợ **embedded server** (Tomcat, Jetty hoặc Undertow), cho phép ứng dụng chạy độc lập dưới dạng file JAR mà không cần cài đặt hoặc triển khai trên server bên ngoài. Điều này giúp việc đóng gói, triển khai, và quản lý ứng dụng trở nên đơn giản, nhanh chóng, phù hợp với kiến trúc microservices hiện đại.

Ngoài ra, Spring Boot còn tích hợp sẵn nhiều công cụ hỗ trợ vận hành (Actuator) giúp kiểm tra sức khỏe hệ thống, giám sát, và cung cấp các thông tin runtime (metrics, health check, environment, v.v.). Framework này cũng tuân thủ triết lý "**Convention over Configuration**", đưa ra các cấu hình mặc định hợp lý, giúp giảm thiểu số lượng cấu hình phải viết thủ công.

Nhờ những ưu điểm vượt trội, Spring Boot ngày nay được sử dụng rộng rãi trong phát triển ứng dụng web, API backend, hệ thống microservices, và các ứng dụng doanh nghiệp yêu cầu khả năng mở rộng và bảo trì cao.

### 2.2. RESTful API:

**RESTful API** (Representational State Transfer Application Programming Interface) là một kiểu thiết kế API phổ biến, dựa trên nguyên tắc của kiến trúc REST, được giới thiệu lần đầu bởi Roy Fielding vào năm 2000.

REST mô tả một tập hợp các nguyên tắc giúp xây dựng các dịch vụ web nhẹ, dễ mở rộng và dễ bảo trì. Một API được coi là RESTful khi tuân thủ các nguyên tắc cốt lõi sau:

- **Client–Server (Khách–Chủ):** Tách biệt phần giao diện (client) và phần xử lý dữ liệu (server), giúp phát triển và quản lý độc lập.
- **Stateless (Không trạng thái):** Mỗi request từ client đến server phải chứa đủ thông tin để server hiểu và xử lý; server không lưu trạng thái của client giữa các request.
- **Cacheable (Có thể cache):** Các response có thể được cache để giảm tải và cải thiện hiệu suất.
- **Uniform Interface (Giao diện thống nhất):** Các API sử dụng tập hợp phương thức HTTP chuẩn như GET (lấy dữ liệu), POST (tạo mới), PUT (cập nhật), DELETE (xóa).
- **Layered System (Hệ thống phân lớp):** Client không cần biết trực tiếp server nào xử lý, có thể có thêm các lớp proxy hoặc load balancer.

RESTful API thường sử dụng **HTTP/HTTPS** làm giao thức truyền tải và trả về dữ liệu dưới dạng **JSON** (phổ biến), đôi khi là XML hoặc định dạng khác.

Nhờ sự đơn giản, dễ sử dụng, và khả năng tương thích cao, RESTful API đã trở thành lựa chọn hàng đầu để xây dựng các dịch vụ backend cho ứng dụng web, mobile, IoT và các hệ thống phân tán.

### 2.3. MySQL:

**MySQL** là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS — Relational Database Management System) mã nguồn mở, được phát triển ban đầu bởi công ty MySQL AB (Thụy Điển) và hiện nay được Oracle Corporation duy trì.

MySQL sử dụng **SQL (Structured Query Language)** để quản lý và thao tác dữ liệu. Đây là một trong những hệ quản trị cơ sở dữ liệu được sử dụng rộng rãi nhất trên thế giới, đặc biệt phổ biến trong các ứng dụng web, dịch vụ SaaS, thương mại điện tử và các nền tảng quản lý nội dung (CMS) như WordPress, Joomla.

#### *Ưu điểm của MySQL*

- **Miễn phí và mã nguồn mở:** Cho phép sử dụng tự do, dễ dàng tùy chỉnh và tích hợp, có cộng đồng hỗ trợ lớn.
- **Hiệu suất cao và ổn định:** Hoạt động nhanh, hỗ trợ khối lượng lớn truy vấn và dữ liệu, đáp ứng tốt nhu cầu thực tế của các website hoặc hệ thống doanh nghiệp.
- **Khả năng mở rộng:** Dễ dàng mở rộng từ ứng dụng nhỏ đến các hệ thống lớn, hỗ trợ replication (nhân bản dữ liệu) và clustering.
- **Tương thích đa nền tảng:** Chạy trên hầu hết các hệ điều hành (Linux, Windows, macOS).
- **Dễ sử dụng và triển khai:** Giao diện quản lý thân thiện, cú pháp SQL chuẩn, dễ học đối với người mới.
- **Bảo mật tốt:** Hỗ trợ phân quyền người dùng chi tiết, cơ chế mã hóa kết nối, xác thực mạnh mẽ.

#### *Nhược điểm của MySQL*

- **Hạn chế về tính năng cao cấp:**  
Một số tính năng nâng cao (như kiểm soát giao dịch phức tạp, xử lý song song ở mức cao) vẫn kém hơn so với các hệ quản trị thương mại (ví dụ: Oracle Database, SQL Server).
- **Giới hạn về khả năng xử lý dữ liệu lớn rất phức tạp:**  
Khi khối lượng dữ liệu cực lớn (hàng chục TB), việc tối ưu cần nhiều kỹ thuật bổ sung.
- **Thiếu sự hỗ trợ toàn diện cho tiêu chuẩn SQL:**  
Một số chuẩn SQL không được hỗ trợ đầy đủ, hoặc cách triển khai khác biệt so với các RDBMS khác.

### 3. RabbitMQ hoặc Kafka

- **RabbitMQ:**
  - Dễ cài đặt và tích hợp với Spring Boot (qua Spring AMQP).
  - Dùng mô hình Producer – Exchange – Queue – Consumer.
  - Phù hợp cho các tác vụ như gửi thông báo, xử lý nền.
- **Apache Kafka:**
  - Hiệu năng cao, xử lý lượng lớn dữ liệu theo thời gian thực.
  - Mô hình publish-subscribe, dùng topic để phân phối sự kiện đến nhiều consumer.
  - Phù hợp nếu bạn hướng đến quy mô lớn, nhiều sự kiện phức tạp.

### 4. Eureka – Service Discovery

- Là một phần của hệ sinh thái Spring Cloud, đóng vai trò như **trung tâm đăng ký các dịch vụ**.
- Mỗi microservice sẽ **tự động đăng ký** vào Eureka khi khởi động.
- Khi một dịch vụ cần gọi một dịch vụ khác, nó sẽ tra cứu thông tin (IP/port) từ Eureka thay vì hard-code.
- Tăng tính linh hoạt và khả năng tự phục hồi khi dịch vụ thay đổi hoặc chết tạm thời.

### 5. Spring Cloud Gateway

- Là **API Gateway** giúp định tuyến các request từ người dùng đến đúng microservice.
- Tích hợp dễ dàng với Eureka để biết dịch vụ nào đang sống.
- Có thể thêm các chức năng:
  - Xác thực và phân quyền.
  - Giới hạn tốc độ request.
- Log và giám sát luồng dữ liệu ra/vào hệ thống.

## Chương 3: Phân tích và thiết kế hệ thống

### 1. Phân tích nghiệp vụ và thiết kế

#### 1.1 Khảo sát hiện trạng

Qua khảo sát một vài chung cư thì quy trình quản lý hợp đồng dịch vụ tại các chung cư không có phần mềm hỗ trợ chủ yếu thực hiện thủ công và bằng giấy tờ. Các công việc này đòi hỏi sự tham gia của nhiều bộ phận và dễ dàng gặp phải sai sót, gây bất tiện cho cả cư dân và ban quản lý. Quy trình này bao gồm các bước sau:

- Đăng ký thông tin và lựa chọn dịch vụ:** Khi cư dân mới chuyển đến chung cư, họ phải trực tiếp đến văn phòng ban quản lý để đăng ký tài khoản. Cư dân sẽ được cung cấp thông tin về các dịch vụ chung cư cung cấp như điện, nước, bảo trì, vệ sinh, bảo vệ, và gửi xe, .... Sau đó, cư dân sẽ lựa chọn các dịch vụ cần sử dụng và điền vào mẫu đăng ký dịch vụ trên giấy. Quy trình này thường diễn ra thủ công, không có hệ thống tự động để hỗ trợ việc lựa chọn dịch vụ.
- Lập hợp đồng dịch vụ:** Sau khi cư dân hoàn thành thủ tục đăng ký dịch vụ, ban quản lý sẽ tiến hành lập hợp đồng dịch vụ bằng tay, liệt kê các dịch vụ cư dân đã đăng ký và các điều khoản kèm theo như giá trị dịch vụ, thời gian cung cấp, các điều kiện thanh toán, v.v. Hợp đồng này thường được in ra giấy và yêu cầu cư dân ký trực tiếp. Việc lập hợp đồng thủ công tiềm ẩn nguy cơ sai sót trong quá trình nhập dữ liệu hoặc ghi nhầm thông tin.
- Quản lý hợp đồng:** Mỗi hợp đồng dịch vụ sẽ được ban quản lý lưu trữ theo dạng hồ sơ giấy và được lưu trữ vào bảng tính Excel. Việc theo dõi hợp đồng, kiểm tra các điều khoản và cập nhật thông tin cư dân rất khó khăn khi dữ liệu được phân tán và không được tổ chức một cách khoa học. Khi có yêu cầu từ cư dân về việc thay đổi thông tin hợp đồng hoặc yêu cầu sửa chữa, ban quản lý cần phải tìm kiếm và đổi chiếu thủ công, rất mất thời gian.
- Tạo hóa đơn và thanh toán:** Vào mỗi kỳ thanh toán, ban quản lý sẽ tính toán thủ công các khoản phí dịch vụ dựa trên hợp đồng đã ký. Các khoản phí này sẽ được ghi vào hóa đơn giấy và phát cho cư dân. Cư dân có thể thanh toán bằng tiền mặt trực tiếp tại văn phòng hoặc qua chuyển khoản ngân hàng. Tuy nhiên, quy trình này có thể dẫn đến sai sót trong việc ghi hóa đơn hoặc mất thời gian chờ đợi cho cả cư dân và nhân viên quản lý.
- Theo dõi thanh toán và hợp đồng hết hạn:** Sau khi hóa đơn được phát hành, ban quản lý phải theo dõi thủ công các khoản thanh toán từ cư dân. Điều này bao gồm việc kiểm tra xem cư dân đã thanh toán đầy đủ hay chưa, và ghi nhận trạng thái thanh toán trong hệ thống giấy tờ. Khi hợp đồng hết hạn, cư dân không nhận được cảnh báo tự động, dẫn đến việc có thể quên gia hạn hợp đồng hoặc thanh toán phí dịch vụ. Ban quản lý cũng phải nhắc nhở cư dân qua gọi điện thoại hoặc email, nhưng việc này thường xuyên bị bỏ qua hoặc không thực hiện kịp thời.
- Vấn đề liên quan đến thông tin cư dân:** Việc thay đổi thông tin cư dân như chuyển đổi căn hộ, thay đổi số điện thoại, email hay thông tin thanh toán không được cập nhật đồng bộ trong hệ thống. Cư dân sẽ phải liên hệ với ban quản lý để yêu cầu chỉnh sửa thông tin trong hợp đồng hoặc hóa đơn, gây mất thời gian và khó khăn trong việc theo dõi.

7. **Khó khăn trong việc quản lý dịch vụ:** Với hệ thống quản lý thủ công, việc quản lý số lượng lớn dịch vụ và hợp đồng trở nên khó khăn. Ban quản lý phải kiểm soát thủ công từng dịch vụ đã đăng ký của cư dân, có thể dẫn đến việc quên gia hạn dịch vụ, hoặc không kịp thời cung cấp dịch vụ khi cư dân yêu cầu.

Mỗi cư dân (hoặc hộ gia đình) sẽ ký một **hợp đồng tổng** với BQL, trong đó bao gồm nhiều dịch vụ có thể đăng ký hoặc hủy bất kỳ lúc nào. Một số dịch vụ tiêu biểu:

- Dịch vụ thuê nhà/căn hộ.
- Điện, nước.
- Dịch vụ vệ sinh căn hộ hoặc khu vực chung.
- Dịch vụ bảo vệ, an ninh, lắp đặt camera.
- Dịch vụ giữ xe (ô tô, xe máy, xe đạp, ...).
- Dịch vụ Internet, truyền hình, tiện ích số.
- Các dịch vụ tiện ích khác (phòng gym, hồ bơi, kho chứa đồ, ...).

Trong đó:

#### **Dịch vụ không thể hủy riêng lẻ**

- **Thuê căn hộ** (chỉ kết thúc khi kết thúc hợp đồng tổng).
- **Điện, nước** (gắn liền với sinh hoạt, không thể dừng riêng).
- **Bảo vệ, an ninh** (tính chung toàn cư dân, không thể cá nhân hủy).
- **Đặc điểm:**
  - + Áp dụng mặc định khi ký hợp đồng tổng.
  - + Không thể hủy hoặc tạm dừng riêng từng dịch vụ (chỉ chấm dứt khi thanh lý hợp đồng tổng)
  - + Phí dịch vụ bắt buộc, thu hàng tháng.

#### **Dịch vụ có thể hủy**

- **Giữ xe.**
- **Vệ sinh căn hộ riêng** (nếu có gói riêng).
- **Internet/truyền hình.**
- **Dịch vụ tiện ích khác** (gym, hồ bơi, kho chứa, sân thể thao...).
- **Đặc điểm:**
  - + Cư dân có thể đăng ký thêm, hủy, hoặc điều chỉnh bất kỳ lúc nào.
  - + Khi hủy, phí dịch vụ dừng tính từ kỳ tiếp theo (trừ khi có cam kết tối thiểu).
  - + Quản lý bằng các phụ lục dịch vụ riêng lẻ, liên kết với hợp đồng tổng.

#### **Điều khoản thông thường của một phụ lục dịch vụ**

**Điều 1: Nội dung dịch vụ**

Mô tả chi tiết dịch vụ cung cấp (ví dụ: gói Internet 100Mbps, thuê 1 chỗ ô tô, ...).

**Điều 2: Thời gian cung cấp**

Ngày bắt đầu, ngày kết thúc (nếu có kỳ hạn), hoặc "vô thời hạn" đến khi có yêu cầu hủy.

**Điều 3: Phí dịch vụ và phương thức thanh toán**

- Mức phí cụ thể (cố định hoặc biến đổi).
- Hình thức thanh toán (chuyển khoản, ví điện tử, trực tiếp).

**Điều 4: Quyền và nghĩa vụ của cư dân**

- Quyền sử dụng, quyền yêu cầu hỗ trợ.
- Nghĩa vụ bảo quản, tuân thủ quy định.

**Điều 5: Quyền và nghĩa vụ của BQL**

Đảm bảo chất lượng, bảo trì, hỗ trợ kỹ thuật.

Thông báo thay đổi chính sách hoặc gián đoạn.

**Điều 6: Điều kiện hủy, chấm dứt**

Điều kiện để cư dân yêu cầu hủy.

Quy định vi phạm và quyền BQL đơn phương chấm dứt.

**Điều 7: Giải quyết tranh chấp**

Nguyên tắc xử lý khi phát sinh khiếu nại.

**Phân loại dịch vụ theo cách tính và thu tiền**

*Tính phí cố định hàng tháng*

- Thuê căn hộ.
- An ninh, bảo vệ.
- Vệ sinh khu vực chung.
- Giữ xe (phí cố định theo loại xe).
- Internet/truyền hình (theo gói).

**- Đặc điểm:**

- + Giá không đổi mỗi kỳ, thu đều hàng tháng.
- + Tính vào hóa đơn tổng, dễ dự toán.

**Tính phí biến đổi theo mức tiêu thụ**

- Điện (kWh).
- Nước (m<sup>3</sup>).

**- Đặc điểm:**

- + Phí phụ thuộc chỉ số thực tế hàng tháng.
- + Có đồng hồ hoặc thiết bị đo.

**Tính phí theo lượt, theo nhu cầu**

- Dịch vụ vệ sinh riêng (tính theo lần hoặc gói tuần/tháng).
- Tiện ích khác: hồ bơi, phòng gym, phòng sinh hoạt, thuê kho (tính theo lượt hoặc thuê kỳ hạn).

**- Đặc điểm:**

- + Cư dân chọn khi cần, không bắt buộc.
- + Có thể phát sinh hóa đơn riêng hoặc gộp vào hóa đơn tổng

**Mô hình hợp đồng**

**Hợp đồng tổng (hợp đồng khung)**

- Là văn bản pháp lý xác lập quan hệ giữa BQL và cư dân.
- Quy định quyền, nghĩa vụ chung, phương thức thanh toán, xử lý tranh chấp.
- Có hiệu lực dài hạn, đến khi cư dân chấm dứt cư trú hoặc bị hủy.

**Phụ lục dịch vụ (dịch vụ con)**

- Được gắn kèm hợp đồng tổng, thể hiện từng dịch vụ riêng biệt.
- Ghi chi tiết: loại dịch vụ, gói dịch vụ, mức phí, điều kiện điều chỉnh/hủy.
- Có thể thêm, xóa, hoặc điều chỉnh linh hoạt mà không cần thay đổi hợp đồng tổng.

**1.2 Yêu cầu hệ thống**

Dựa trên khảo sát hiện trạng và phân tích nghiệp vụ, hệ thống được thiết kế bám sát mô hình hợp đồng tổng + phụ lục dịch vụ, phục vụ hai nhóm đối tượng chính: Ban Quản Lý (BQL) và Cư dân. Các yêu cầu được chia thành hai nhóm: chức năng và phi chức năng.

**Yêu cầu chức năng**

**Chức năng của Ban Quản Lý (BQL)**

- **Quản lý cư dân & tài khoản:** duyệt, cấp, khóa/mở khóa hoặc hủy tài khoản; cập nhật thông tin cư dân & căn hộ.
- **Quản lý dịch vụ:** định nghĩa & cấu hình dịch vụ bắt buộc (thuê căn hộ, điện, nước, bảo vệ, vệ sinh chung, bảo trì) và tùy chọn (giữ xe, Internet, vệ sinh riêng, tiện ích khác); thiết lập điều khoản, phí & chu kỳ thanh toán; quản lý trạng thái dịch vụ.
- **Quản lý hợp đồng & phụ lục:** tra cứu, cập nhật lịch sử thay đổi, xử lý đăng ký/hủy/sửa dịch vụ theo đúng ràng buộc hợp đồng.
- **Quản lý hóa đơn & thanh toán:** lập & gửi hóa đơn tổng hợp hàng tháng; theo dõi thanh toán, công nợ; xử lý vi phạm.

## Báo cáo Thực tập Tốt nghiệp Đại học

- **Gửi thông báo:** thông báo tự động về hóa đơn đến hạn, hợp đồng sắp hết hạn, thay đổi phí dịch vụ; gửi thông báo thủ công khi cần.

### Chức năng của Cư dân

- **Quản lý tài khoản:** đăng ký, cập nhật thông tin cá nhân, theo dõi trạng thái phê duyệt.
- **Quản lý hợp đồng:** ký hợp đồng tổng; đăng ký, hủy, thay đổi dịch vụ; tra cứu hợp đồng & phụ lục; xem lịch sử thay đổi.
- **Thanh toán & tra cứu hóa đơn:** thanh toán online hoặc thủ công; xem lịch sử thanh toán.
- **Nhận thông báo:** xem thông báo về hợp đồng, dịch vụ, hóa đơn; nhận cảnh báo tự động khi hợp đồng sắp đến hạn hoặc nợ phí dịch vụ.

### Chức năng hệ thống

- Quản lý quyền & phân quyền giữa BQL và cư dân, sẵn sàng mở rộng thêm nhân viên hỗ trợ với quyền hạn riêng.
- Hỗ trợ xử lý vi phạm: tạm ngưng dịch vụ, hủy hợp đồng hoặc phụ lục khi cư dân vi phạm.

### Yêu cầu phi chức năng

- **Tính sẵn sàng và ổn định:**
  - Hệ thống hoạt động liên tục, phục vụ nhu cầu truy cập 24/7.
  - Khả năng tự phục hồi khi một dịch vụ gặp sự cố nhờ kiến trúc Microservices.
- **Hiệu suất:**
  - Đảm bảo phản hồi nhanh các thao tác CRUD của người dùng.
  - Hỗ trợ đồng thời nhiều người dùng truy cập và thao tác.
- **Khả năng mở rộng:**
  - Dễ dàng thêm/bớt dịch vụ, chức năng, hoặc triển khai cho nhiều chung cư khác.
  - Có thể tích hợp thêm các module hoặc công nghệ mới mà không ảnh hưởng đến hệ thống đang chạy.
- **Bảo mật:**
  - Đảm bảo an toàn dữ liệu cư dân, hợp đồng, hóa đơn.
  - Có xác thực, phân quyền, mã hóa dữ liệu.
- **Khả năng khôi phục:**

Dữ liệu được sao lưu định kỳ và có thể khôi phục khi xảy ra sự cố.
- **Thân thiện với người dùng:**
  - Giao diện web trực quan, dễ sử dụng cho cả BQL và cư dân.

- o Hỗ trợ đa nền tảng: máy tính, tablet, smartphone.
- **Khả năng tích hợp:**  
Để dàng tích hợp thêm công thanh toán điện tử hoặc các dịch vụ bên thứ ba.
- **Ghi log và giám sát:**  
Ghi nhận hoạt động hệ thống, hỗ trợ giám sát & xử lý lỗi.

### 1.3 Đối tượng người dùng & Chức năng chi tiết

#### a. Đối tượng người dùng

##### **Ban quản lý chung cư (BQL)**

###### *Người cảnh và vai trò*

Ban quản lý chung cư (BQL) là đơn vị đại diện pháp lý, được giao trách nhiệm vận hành, quản lý và cung cấp các dịch vụ nội khu cho cư dân theo đúng quy định pháp luật. Trong hệ thống, BQL được xem là người sử dụng nội bộ, thực hiện các nghiệp vụ như: duyệt tài khoản cư dân, định nghĩa dịch vụ, quản lý hợp đồng, lập hóa đơn, xác nhận thanh toán, gửi thông báo.

###### *Cách đại diện*

Mỗi nghiệp vụ trên hệ thống do một cá nhân cụ thể thuộc BQL thực hiện (ví dụ: trưởng ban, kế toán, nhân viên CSKH). Tuy nhiên, trong giai đoạn đầu, để đảm bảo đơn giản hóa thiết kế và kịp tiến độ, hệ thống chỉ sử dụng một role duy nhất cho BQL, gọi là ADMIN\_BQL. Role ADMIN\_BQL có toàn quyền thực hiện tất cả chức năng liên quan đến quản trị và vận hành.

###### *Lý do giữ 1 role ban đầu*

- Giúp rút gọn thời gian triển khai, phù hợp thời gian nộp đế tài (2 tuần).
- Giảm phức tạp trong thiết kế giao diện và logic kiểm tra quyền.
- Tuy nhiên, hệ thống vẫn chuẩn bị sẵn hạ tầng hỗ trợ phân quyền (Role-based Access Control) để sau này dễ dàng mở rộng thêm các role chuyên biệt như:
  - Kế toán : chỉ xử lý hóa đơn, thanh toán.
  - CSKH : chỉ duyệt tài khoản cư dân, hỗ trợ thông tin.
  - Quản lý vận hành: chỉ quản lý dịch vụ, hợp đồng.

##### **Cư dân**

###### *Cách đại diện*

Mỗi cư dân được đại diện bởi thông tin pháp lý chính thức, sử dụng số căn cước công dân (CCCD) hoặc hộ chiếu. Một cư dân có thể sở hữu hoặc đăng ký nhiều căn hộ. Trong hệ thống, CCCD được dùng để định danh duy nhất cư dân (primary identity).

###### *Vai trò và quyền*

- Ký kết hợp đồng dịch vụ với BQL.
- Quản lý hợp đồng, tra cứu thông tin.
- Thanh toán hóa đơn, nhận thông báo.

**b. Lợi ích của phương án "1 role ban đầu"**

- Đơn giản hóa quy trình phát triển (chỉ cần tập trung 1 view quản lý).
- Dễ kiểm thử, dễ triển khai demo.
- Giảm tải khói lượng công việc khi chỉ có 2 tuần để hoàn thiện đề tài.
- Giúp tập trung vào nghiệp vụ cốt lõi: quản lý cư dân, hợp đồng, dịch vụ, hóa đơn.

**c. Chức năng chi tiết**

**Chức năng dành cho ban quản lý**

- Quản lý tài khoản cư dân:
  - Xem danh sách cư dân đã đăng ký.
  - Duyệt hoặc hủy tài khoản.
  - Khóa/mở khóa tài khoản khi cần.
- Quản lý dịch vụ và hợp đồng:
  - Khai báo danh mục dịch vụ: thuê nhà, điện, nước, bảo trì, vệ sinh, bảo vệ, gửi xe...
  - Định nghĩa điều khoản áp dụng cho từng dịch vụ.
  - Tạo mẫu hợp đồng để cư dân chọn.
- Quản lý hóa đơn:
  - Lập hóa đơn tự động hàng tháng dựa trên hợp đồng.
  - Gửi thông báo nhắc nhở đến cư dân.
  - Xác nhận thanh toán (thủ công hoặc online).

**Chức năng dành cho cư dân**

- Quản lý tài khoản cá nhân:
  - Đăng ký tài khoản.
  - Đăng nhập, cập nhật thông tin cá nhân.
- Đăng ký hợp đồng dịch vụ:
  - Tạo hợp đồng mới, chọn dịch vụ cần dùng.
  - Xem điều khoản dịch vụ.
  - Chính sửa dịch vụ trong hợp đồng.
- Tra cứu hợp đồng:
  - Xem danh sách hợp đồng đã ký.
  - Xem trạng thái thanh toán, thời hạn hợp đồng.
- Thanh toán:
  - Thanh toán thủ công (chuyển khoản, tiền mặt).
  - Thanh toán online qua VNPay.
- Nhận thông báo:
  - Nhắc nhở thời hạn hợp đồng.
  - Thông báo khi có hóa đơn mới hoặc sắp hết hạn dịch vụ.
  - Gửi qua email và hiển thị trên giao diện web.

**2 Mô hình Use Case**

**Nhóm 0: Đặc biệt:**

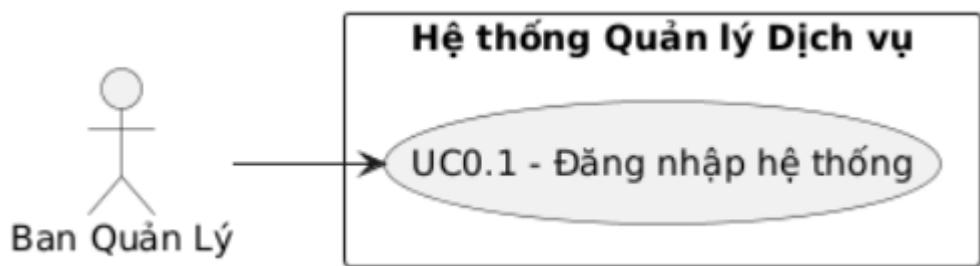


Figure 5: Usecase Đăng nhập

UC0.1 – Ban quản lý đăng nhập Hệ thống

Table 1: Đặc tả usecase Đăng nhập

Thành phần	Nội dung
Tên Use Case	UC0.1 – Ban quản lý đăng nhập Hệ thống
Tác nhân chính	Ban Quản Lý
Mục tiêu	Tạm ngưng hoặc khôi phục quyền truy cập hệ thống của cư dân
Tiền điều kiện	<ul style="list-style-type: none"> <li>- BQL đã có tài khoản hợp lệ trong hệ thống.</li> <li>- Thiết bị của BQL có kết nối internet và trình duyệt web.</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>BQL truy cập vào địa chỉ URL của hệ thống.</li> <li>Hệ thống hiển thị giao diện trang đăng nhập, bao gồm các trường nhập "Tên đăng nhập" và "Mật khẩu" cùng với nút "Đăng nhập".</li> <li>BQL nhập "Tên đăng nhập" và "Mật khẩu" vào các trường tương ứng.</li> <li>BQL nhấn nút "Đăng nhập".</li> <li>Hệ thống kiểm tra thông tin đăng nhập:             <ol style="list-style-type: none"> <li>Xác thực tên đăng nhập và mật khẩu đã nhập với dữ liệu đã được băm (hash) lưu trữ trong cơ sở dữ liệu.</li> <li>Kiểm tra trạng thái tài khoản (đang hoạt động, bị khóa, v.v.).</li> </ol> </li> </ol>

Báo cáo Thực tập Tốt nghiệp Đại học

	<p>6. Nếu thông tin hợp lệ và tài khoản đang hoạt động, hệ thống tạo phiên làm việc (session) cho BQL và chuyển hướng BQL đến trang quản trị chính (Dashboard) với đầy đủ quyền quản trị.</p>
Luồng thay thế	<p>AF1 – Thông tin đăng nhập không hợp lệ:</p> <ul style="list-style-type: none"> <li>- Tại Bước 5 của Luồng chính: Nếu tên đăng nhập hoặc mật khẩu không khớp với bất kỳ tài khoản nào trong hệ thống.</li> <li>- Hệ thống hiển thị thông báo lỗi: "Tên đăng nhập hoặc mật khẩu không đúng. Vui lòng thử lại."</li> <li>- Luồng quay lại Bước 3 để BQL có thể nhập lại.</li> </ul> <p>AF4 – Lỗi hệ thống:</p> <ul style="list-style-type: none"> <li>- Tại Bước 5 hoặc 6 của Luồng chính: Nếu xảy ra lỗi không mong muốn từ hệ thống (ví dụ: lỗi kết nối cơ sở dữ liệu, lỗi server).</li> <li>- Hệ thống hiển thị thông báo lỗi chung: "Đã xảy ra lỗi hệ thống. Vui lòng thử lại sau."</li> <li>- Luồng kết thúc.</li> </ul>
Hậu điều kiện	BQL được đăng nhập vào hệ thống và có quyền truy cập vào các chức năng quản trị.

## Nhóm 1: Quản lý tài khoản cư dân

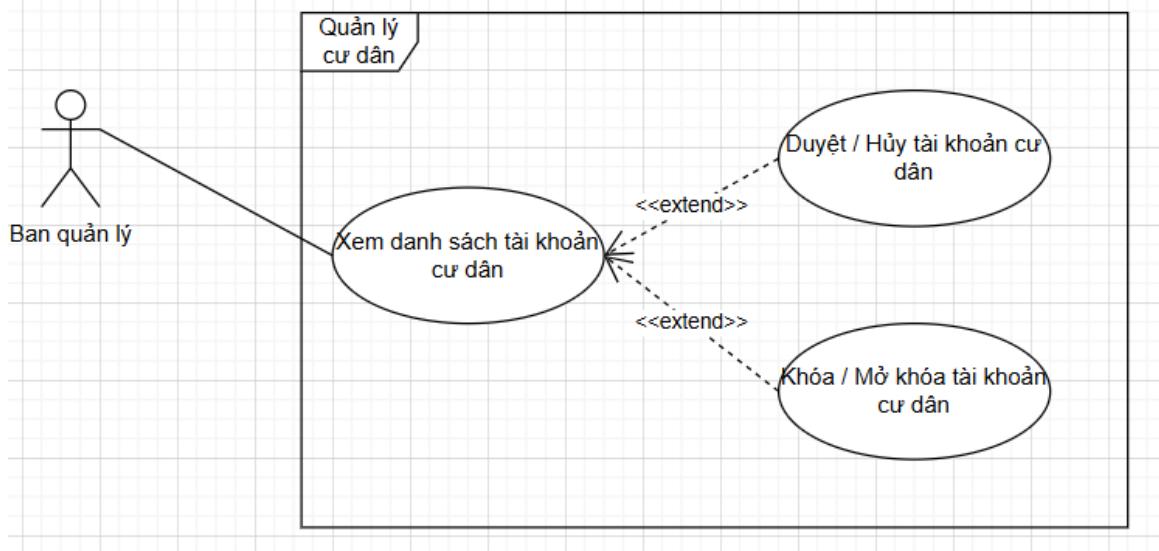


Figure 6: Usecase Quản lý tài khoản cư dân

UC1.1 – Xem danh sách cư dân đã đăng ký

Table 2: ĐẶC TẢ USECASE XEM DANH SÁCH TÀI KHOẢN CƯ DÂN

Thành phần	Nội dung
Tên Use Case	UC1.1 – Xem danh sách cư dân đã đăng ký
Tác nhân chính	Ban Quản Lý
Mục tiêu	Xem thông tin tài khoản cư dân đã đăng ký để duyệt hoặc xử lý tiếp theo
Tiền điều kiện	BQL đã đăng nhập vào hệ thống với quyền quản trị
Luồng chính	<p>BQL đăng nhập vào hệ thống</p> <p>BQL chọn menu “<b>Quản lý tài khoản cư dân</b>”, sau đó chọn tab “Danh sách đã đăng ký”</p> <p>Hệ thống hiển thị danh sách tất cả tài khoản cư dân</p> <p>Có ô <b>chọn trạng thái</b> (Tất cả, Chờ duyệt, Đang hoạt động, Đã khóa) để lọc danh sách</p> <p>Hệ thống hiển thị bảng thông tin gồm:</p> <ul style="list-style-type: none"> <li>– Họ tên cư dân</li> </ul>

## Báo cáo Thực tập Tốt nghiệp Đại học

	<ul style="list-style-type: none"> <li>– Căn hộ (VD: A3-1201)</li> <li>– Email đăng nhập</li> <li>– Số điện thoại</li> <li>– Trạng thái tài khoản</li> <li>– Ngày đăng ký</li> </ul> <p>BQL có thể nhấn vào từng dòng để xem chi tiết thông tin cá nhân (trừ thông tin bảo mật)</p>
Luồng thay thế	<p>EF1 – Không có dữ liệu:</p> <p>Tại bước 3 của Luồng chính (Hệ thống hiển thị danh sách tất cả tài khoản cư dân), nếu hệ thống không tìm thấy bất kỳ cư dân nào đã đăng ký.</p> <p>Hệ thống hiển thị thông báo: "Không có cư dân nào đã đăng ký trong hệ thống."</p> <p>Ban Quản Lý (BQL) có thể quay lại màn hình chính hoặc thực hiện các thao tác khác.</p> <p>EF2 – Lỗi kết nối cơ sở dữ liệu:</p> <p>Tại bất kỳ bước nào yêu cầu truy xuất dữ liệu từ hệ thống.</p> <p>Hệ thống hiển thị thông báo lỗi: "Lỗi kết nối cơ sở dữ liệu. Vui lòng thử lại sau."</p> <p>Luồng kết thúc, BQL không thể xem danh sách cư dân.</p>
Hậu điều kiện	BQL xem được thông tin tài khoản cư dân
Ngoại lệ	Lỗi kết nối cơ sở dữ liệu, phiên đăng nhập hết hạn

### UC1.2 – Duyệt tài khoản cư dân

Thành phần	Nội dung
Tên Use Case	UC1.2 – Duyệt tài khoản cư dân
Tác nhân chính	Ban Quản Lý
Mục tiêu	BQL xem xét thông tin đăng ký tài khoản của cư dân và phê duyệt để kích hoạt tài

Báo cáo Thực tập Tốt nghiệp Đại học

	<p>khoản thành trạng thái "Đang hoạt động". Việc duyệt này chỉ liên quan đến thông tin tài khoản cơ bản và căn hộ, không liên quan đến hợp đồng dịch vụ.</p>
Tiền điều kiện	<ul style="list-style-type: none"> <li>- BQL đã đăng nhập vào hệ thống với quyền quản trị.</li> <li>- Có ít nhất một tài khoản cư dân đang ở trạng thái "Chờ duyệt".</li> </ul>
Luồng chính	<p>BQL truy cập chức năng "Quản lý tài khoản cư dân", sau đó chọn trạng thái "Chờ duyệt".</p> <p>Hệ thống hiển thị danh sách các tài khoản cư dân đang ở trạng thái "Chờ duyệt".</p> <p>BQL chọn một tài khoản cụ thể để xem chi tiết thông tin đăng ký (Họ tên, Email, Số điện thoại, Căn hộ đã được chọn...).</p> <p>BQL xem xét thông tin tài khoản để đảm bảo tính hợp lệ và đầy đủ.</p> <p>BQL quyết định Duyệt hoặc Từ chối tài khoản.</p> <p>Nếu Duyệt:</p> <ul style="list-style-type: none"> <li>- BQL xác nhận phê duyệt.</li> <li>- Hệ thống cập nhật trạng thái của tài khoản cư dân từ "Chờ duyệt" thành "Đang hoạt động".</li> <li>- Hệ thống gửi thông báo xác nhận tài khoản đã được kích hoạt cho Cư dân liên quan.</li> <li>- Hệ thống ghi lại lịch sử duyệt (người duyệt, thời gian, trạng thái).</li> </ul>
Luồng thay thế	<p>AF1 – BQL Từ chối tài khoản:</p> <ul style="list-style-type: none"> <li>- Tại bước 5 của Luồng chính, thay vì Duyệt, BQL chọn Từ chối.</li> <li>- BQL nhập lý do từ chối.</li> </ul>

## Báo cáo Thực tập Tốt nghiệp Đại học

	<ul style="list-style-type: none"> <li>- Hệ thống cập nhật trạng thái của tài khoản cư dân thành "Đã từ chối" hoặc "Đã khóa".</li> <li>- Hệ thống gửi thông báo kèm lý do từ chối cho Cư dân liên quan.</li> <li>- Hệ thống ghi lại lịch sử từ chối.</li> </ul>
Hậu điều kiện	Tài khoản được kích hoạt và có thể sử dụng hệ thống
Ngoại lệ	Cư dân đã bị xóa, lỗi cập nhật trạng thái

### UC1.3 – Khóa / Mở khóa tài khoản cư dân

Table 3: **Đặc tả UC Khóa / Mở khóa tài khoản cư dân**

Thành phần	Nội dung
Tên Use Case	UC1.3 – Khóa / Mở khóa tài khoản cư dân
Tác nhân chính	Ban Quản Lý
Mục tiêu	Tạm ngưng hoặc khôi phục quyền truy cập hệ thống của cư dân
Tiền điều kiện	Tài khoản cư dân đang ở trạng thái "Đang hoạt động" hoặc "Đã khóa"
Luồng chính	<p>Khóa tài khoản:</p> <ol style="list-style-type: none"> <li>BQL chọn cư dân đang "Đang hoạt động"</li> <li>Nhấn "Khóa tài khoản"</li> <li>Xác nhận thao tác</li> <li>Hệ thống chuyển trạng thái thành "Đã khóa"</li> </ol> <p>Mở khóa tài khoản:</p> <ol style="list-style-type: none"> <li>BQL chọn cư dân đang "Đã khóa"</li> <li>Nhấn "Mở khóa tài khoản"</li> <li>Xác nhận thao tác</li> <li>Hệ thống chuyển trạng thái thành Đang hoạt động</li> </ol>
Luồng thay thế	<p>AF1 – Hủy thao tác:</p> <ol style="list-style-type: none"> <li>BQL nhấn "Hủy" hoặc đóng hộp thoại</li> <li>Hệ thống không thay đổi trạng thái</li> </ol>

Hậu điều kiện	Trạng thái tài khoản được cập nhật tương ứng
---------------	--

## Nhóm 2. Quản lý dịch vụ

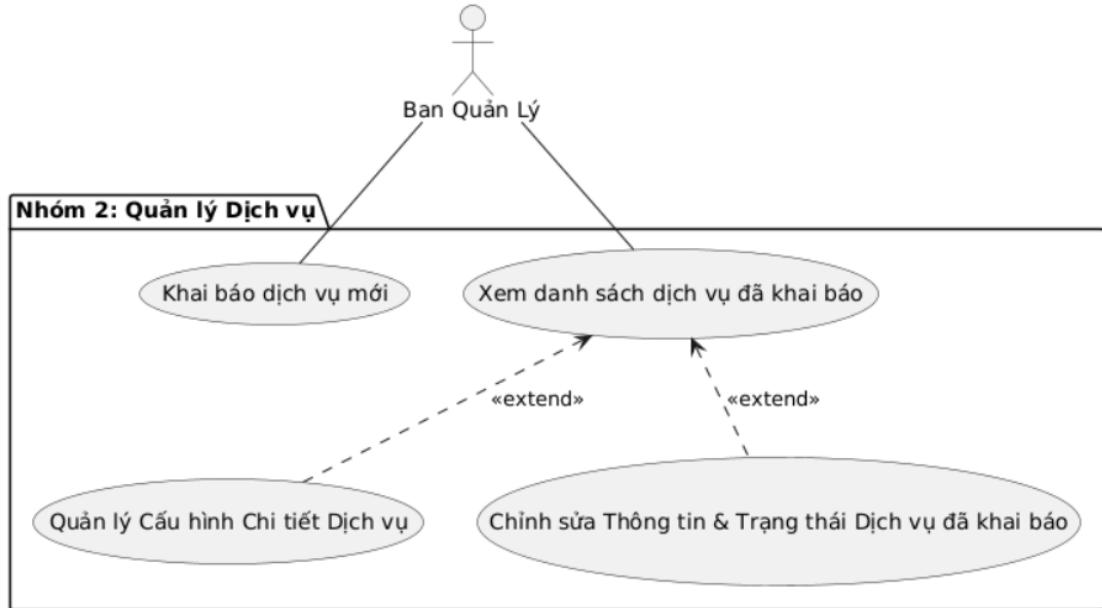


Figure 7: UC Quản lý dịch vụ

Use Case UC2.1 – Khai báo dịch vụ mới

Table 4: ĐẶC TẢ UC KHAI BÁO DỊCH VỤ MỚI

Thành phần	Nội dung
Tên Use Case	UC2.1 – Khai báo dịch vụ mới
Tác nhân chính	Ban Quản Lý
Mục tiêu	BQL định nghĩa và thêm một loại dịch vụ mới vào hệ thống, bao gồm các thuộc tính cơ bản và phương pháp tính phí tổng quát, và đánh dấu xem dịch vụ này có phải là bắt buộc đối với hợp đồng hay không.
Tiền điều kiện	BQL đã đăng nhập hệ thống và có quyền thao tác chức năng Quản lý dịch vụ
Luồng chính	BQL truy cập chức năng "Quản lý Dịch vụ", sau đó chọn "Khai báo dịch vụ mới". Hệ thống hiển thị một biểu mẫu để khai báo thông tin dịch vụ.

	<p>BQL nhập các thông tin cần thiết:</p> <ul style="list-style-type: none"> <li>- Tên Dịch vụ (ví dụ: Điện sinh hoạt, Nước sinh hoạt, Phí quản lý căn hộ).</li> <li>- Mã Dịch vụ (mã định danh duy nhất).</li> <li>- Đơn vị tính (ví dụ: kWh, m<sup>3</sup>, VNĐ/tháng, VNĐ/m<sup>2</sup>).</li> <li>- Loại hình tính phí tổng quát (ví dụ: Theo bậc thang, Theo định mức, Cố định, Theo m<sup>2</sup>).</li> <li>- Thuộc tính "Dịch vụ bắt buộc": Chọn Có/Không để đánh dấu dịch vụ này có phải là bắt buộc đối với hợp đồng dịch vụ của cư dân hay không.</li> <li>- Mô tả dịch vụ.</li> </ul> <p>BQL xác nhận lưu thông tin dịch vụ mới.</p> <p>Hệ thống lưu thông tin dịch vụ vào cơ sở dữ liệu và hiển thị thông báo "Khai báo dịch vụ thành công".</p>
Luồng thay thế	<p><b>AF1 – Thông tin nhập vào không hợp lệ:</b></p> <p>Tại bước "Nhấn 'Lưu dịch vụ'" của Luồng chính.</p> <p>Nếu một hoặc nhiều trường thông tin (tên dịch vụ, đơn giá, cách tính, v.v.) không tuân thủ quy tắc dữ liệu (ví dụ: tên bị trùng, giá trị không phải số, trường bắt buộc bị bỏ trống).</p> <p>Hệ thống hiển thị cảnh báo/lỗi bên cạnh các trường không hợp lệ và yêu cầu BQL nhập lại.</p> <p>Luồng quay lại bước "Nhập thông tin chung của dịch vụ" hoặc điểm nhập liệu cuối cùng.</p> <p><b>EF1 – Lỗi hệ thống:</b> Xảy ra lỗi trong quá trình lưu dữ liệu dịch vụ</p>

Báo cáo Thực tập Tốt nghiệp Đại học

Hậu điều kiện	Một loại dịch vụ mới đã được thêm vào hệ thống với các thuộc tính đã khai báo, sẵn sàng để cấu hình chi tiết và sử dụng trong hợp đồng.
---------------	---

Use Case UC2.2 – Cấu hình Chi tiết Dịch vụ

Table 5: **Đặc tả Cấu hình Chi tiết Dịch vụ**

Thành phần	Nội dung
Tên Use Case	UC2.2 – Cấu hình Chi tiết Dịch vụ
Tác nhân chính	Ban Quản Lý
Mục tiêu	BQL định nghĩa tạo mới hoặc thay đổi/chỉnh sửa các tham số cụ thể cho phương pháp tính phí của một loại dịch vụ đã khai báo, quản lý các "mẫu tính phí" chi tiết có thể được áp dụng cho hợp đồng.
Tiền điều kiện	<ul style="list-style-type: none"> <li>- BQL đã đăng nhập vào hệ thống với quyền quản trị.</li> <li>- Có ít nhất một loại dịch vụ đã được khai báo trong hệ thống (từ UC2.1).</li> </ul>
Luồng chính	<p>BQL đang ở màn hình "Xem danh sách dịch vụ đã khai báo" (từ UC2.4).</p> <p>BQL chọn một dịch vụ cụ thể từ danh sách (ví dụ: nhấn vào một nút "Cấu hình" hoặc "Xem chi tiết" tương ứng với dịch vụ đó).</p> <p>Hệ thống chuyển sang màn hình "Quản lý Cấu hình Chi tiết Dịch vụ" cho dịch vụ đã chọn.</p> <p>Tại màn hình này, BQL có thể:</p> <p>Tạo Cấu hình Mới cho Dịch vụ đã chọn:</p> <ul style="list-style-type: none"> <li>- Hệ thống hiển thị giao diện để tạo cấu hình mới (tương ứng với "Loại hình tính phí tổng quát" của dịch vụ đã chọn).</li> <li>- BQL nhập các tham số chi tiết (ví dụ: bậc thang và giá tiền, phí cố định,</li> </ul>

## Báo cáo Thực tập Tốt nghiệp Đại học

	<p>thành phần gói, ngày hiệu lực, tên cấu hình).</p> <ul style="list-style-type: none"> <li>- BQL xác nhận lưu cấu hình mới.</li> <li>- Hệ thống lưu mẫu tính phí chi tiết và hiển thị thông báo "Tạo cấu hình chi tiết dịch vụ thành công".</li> </ul> <p>Chỉnh sửa Cấu hình Hiện có của Dịch vụ đã chọn:</p> <ul style="list-style-type: none"> <li>- Hệ thống hiển thị danh sách các cấu hình chi tiết hiện có cho dịch vụ đã chọn.</li> <li>- BQL chọn một cấu hình từ danh sách để chỉnh sửa.</li> <li>- Hệ thống hiển thị các thông tin chi tiết của cấu hình đó dưới dạng có thể chỉnh sửa.</li> <li>- BQL thực hiện các thay đổi cần thiết đối với các tham số tính phí.</li> <li>- BQL xác nhận lưu các thay đổi.</li> <li>- Hệ thống cập nhật cấu hình chi tiết dịch vụ và hiển thị thông báo "Chỉnh sửa cấu hình chi tiết dịch vụ thành công".</li> </ul>
Luồng thay thế	AF1 – Tham số không hợp lệ: Nếu BQL nhập các tham số không hợp lệ (ví dụ: bậc thang chòng chéo, giá tiền âm), hệ thống hiển thị cảnh báo và yêu cầu nhập lại.
Hậu điều kiện	Một mẫu tính phí chi tiết mới đã được tạo hoặc một cấu hình hiện có đã được cập nhật, sẵn sàng để được áp dụng vào hợp đồng.

### Use Case UC2.3 – Chỉnh sửa Thông tin & Trạng thái Dịch vụ đã khai báo

Table 6: **Đặc tả Chỉnh sửa Thông tin & Trạng thái Dịch vụ đã khai báo**

Thành phần	Nội dung
Tên Use Case	UC2.3 – Chỉnh sửa Thông tin & Trạng thái Dịch vụ đã khai báo

Báo cáo Thực tập Tốt nghiệp Đại học

Tác nhân chính	Ban Quản Lý
Mục tiêu	BQL thay đổi các thông tin chung (cấp cao) của một loại dịch vụ đã được khai báo (từ UC2.1), bao gồm mô tả, và quản lý trạng thái hoạt động (cung cấp/dừng cung cấp) cũng như thuộc tính bắt buộc của dịch vụ đó.
Tiền điều kiện	<ul style="list-style-type: none"> <li>- BQL đã đăng nhập vào hệ thống với quyền quản trị.</li> <li>- Có ít nhất một loại dịch vụ đã được khai báo trong hệ thống (từ UC2.1).</li> </ul>
Luồng chính	<p>BQL truy cập chức năng "Quản lý Dịch vụ", sau đó chọn "Xem danh sách dịch vụ đã khai báo" (hoặc chức năng tương tự để quản lý dịch vụ cấp cao).</p> <p>Hệ thống hiển thị danh sách các dịch vụ đã được khai báo.</p> <p>BQL chọn một dịch vụ cụ thể để chỉnh sửa.</p> <p>Hệ thống hiển thị các thông tin chung của dịch vụ đó dưới dạng có thể chỉnh sửa.</p> <p>BQL thực hiện các thay đổi cần thiết:</p> <ul style="list-style-type: none"> <li>- Chỉnh sửa mô tả dịch vụ.</li> <li>- Thay đổi trạng thái cung cấp dịch vụ (ví dụ: từ "Đang cung cấp" sang "Dừng cung cấp" hoặc ngược lại).</li> <li>- Chỉnh sửa thuộc tính "Dịch vụ bắt buộc" (Có/Không).</li> <li>- (Các thông tin chung khác như tên/mã dịch vụ nếu được phép thay đổi sau khi khai báo).</li> </ul> <p>BQL xác nhận lưu các thay đổi.</p> <p>Hệ thống cập nhật thông tin và trạng thái của dịch vụ trong cơ sở dữ liệu và hiển thị thông báo "Cập nhật dịch vụ thành công".</p>

Báo cáo Thực tập Tốt nghiệp Đại học

Luồng thay thế	<p>AF1 – Thông tin không hợp lệ: Nếu BQL nhập thông tin không hợp lệ, hệ thống hiển thị cảnh báo.</p> <p>AF2 – Không thể thay đổi trạng thái: Nếu dịch vụ đang có các hợp đồng/hóa đơn chưa kết thúc, hệ thống có thể cảnh báo và không cho phép thay đổi trạng thái dùng cung cấp, hoặc yêu cầu xác nhận đặc biệt.</p>
Hậu điều kiện	Thông tin chung và/hoặc trạng thái của dịch vụ đã được cập nhật theo thay đổi của BQL.

Use Case UC2.4 – Xem danh sách dịch vụ đã khai báo

Table 7: **Đặc tả Xem danh sách dịch vụ đã khai báo**

Thành phần	Nội dung
Tên Use Case	UC2.4 – Xem danh sách dịch vụ đã khai báo
Tác nhân chính	Ban Quản Lý
Mục tiêu	Cho BQL xem danh sách dịch vụ hiện có trong hệ thống, cùng chức năng chỉnh sửa dịch vụ và phụ lục
Tiền điều kiện	BQL đã tạo dịch vụ
Luồng chính	<p>BQL truy cập menu "Quản lý dịch vụ"</p> <p>Hệ thống hiển thị danh sách dịch vụ gồm: Tên, đơn giá, cách tính, phụ lục đi kèm</p> <p>BQL có thể nhấn "Chỉnh sửa dịch vụ" hoặc "Chỉnh sửa phụ lục" ở từng dòng</p> <p>Hệ thống chuyển đến giao diện chỉnh sửa tương ứng</p>
Luồng thay thế	<p>AF1 – Không có dữ liệu:</p> <p>Tại bước 3 của Luồng chính (Hệ thống hiển thị danh sách tất cả dịch vụ đã khai báo).</p>

	<p>Nếu hệ thống không tìm thấy bất kỳ dịch vụ nào đã được khai báo.</p> <p>Hệ thống hiển thị thông báo: "Chưa có dịch vụ nào được khai báo trong hệ thống."</p> <p>BQL có thể quay lại màn hình chính hoặc thực hiện các thao tác khác.</p>
Hậu điều kiện	<p>AF2 – Lỗi kết nối cơ sở dữ liệu:</p> <p>Tại bất kỳ bước nào yêu cầu truy xuất dữ liệu từ hệ thống.</p> <p>Hệ thống hiển thị thông báo lỗi: "Lỗi kết nối cơ sở dữ liệu. Vui lòng thử lại sau."</p> <p>Luồng kết thúc, BQL không thể xem danh sách dịch vụ.</p>

### Nhóm 3. Quản lý mẫu hợp đồng

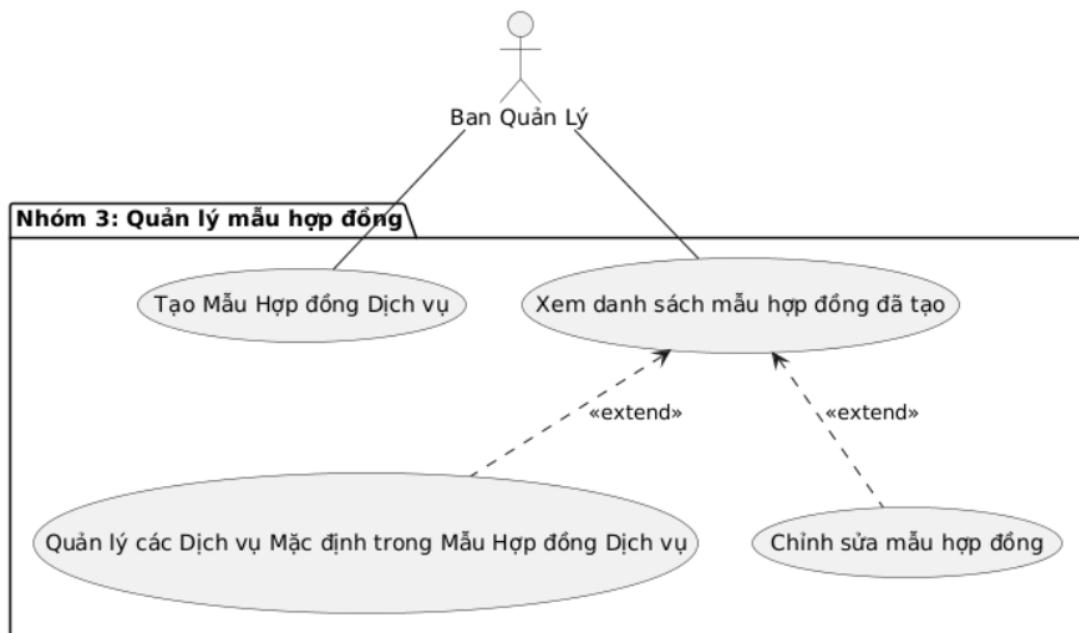


Figure 8: UC Quản lý mẫu hợp đồng

Use Case UC3.1 – Tạo mẫu hợp đồng dịch vụ

# Báo cáo Thực tập Tốt nghiệp Đại học

Table 8: **Đặc tả Tạo mẫu hợp đồng dịch vụ**

Thành phần	Nội dung
Tên Use Case	UC3.1 – Tạo mẫu hợp đồng dịch vụ
Tác nhân chính	Ban Quản Lý
Mục tiêu	Cho phép BQL tạo một khuôn mẫu chuẩn cho các Hợp đồng Dịch vụ cư dân, bao gồm việc định nghĩa các điều khoản chung, cấu trúc nội dung, và thiết lập các dịch vụ sẽ mặc định đi kèm với mẫu hợp đồng này.
Tiền điều kiện	BQL có quyền tạo mẫu hợp đồng mới
Luồng chính	<p>BQL chọn chức năng "Quản lý mẫu hợp đồng".</p> <p>BQL nhấn "Tạo mẫu hợp đồng mới".</p> <p>Hệ thống hiển thị giao diện để nhập thông tin mẫu hợp đồng.</p> <p>BQL nhập các thông tin cần thiết:</p> <ul style="list-style-type: none"> <li>- Tên mẫu hợp đồng.</li> <li>- Mô tả mẫu hợp đồng.</li> <li>- Nội dung các điều khoản chính của hợp đồng.</li> <li>- BQL tùy chọn cấu hình các dịch vụ mặc định sẽ được tự động thêm vào hợp đồng khi sử dụng mẫu này (thực hiện thông qua chức năng của UC3.2).</li> </ul> <p>BQL xác nhận "Lưu".</p> <p>Hệ thống kiểm tra tính hợp lệ của thông tin.</p> <p>Hệ thống lưu mẫu hợp đồng mới vào cơ sở dữ liệu.</p> <p>Hệ thống hiển thị thông báo "Tạo mẫu hợp đồng thành công".</p>
Luồng thay thế	AF1 – Thông tin nhập vào không hợp lệ:

	<p>Nếu một hoặc nhiều trường thông tin của mẫu hợp đồng không tuân thủ quy tắc dữ liệu (ví dụ: tên mẫu bị trống, nội dung quá dài).</p> <p>Hệ thống hiển thị cảnh báo/lỗi bên cạnh các trường không hợp lệ và yêu cầu BQL nhập lại.</p> <p>Luồng quay lại bước "BQL nhập các thông tin cần thiết".</p>
Hậu điều kiện	Một mẫu hợp đồng dịch vụ mới đã được tạo và lưu trữ trong hệ thống, sẵn sàng để sử dụng.

#### Use Case UC3.2 – Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ

Table 9: Đặc tả Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ

Thành phần	Nội dung
Tên Use Case	UC3.2 – Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ
Tác nhân chính	Ban Quản Lý
Mục tiêu	Cho phép BQL định nghĩa và chỉnh sửa danh sách các dịch vụ sẽ tự động được thêm vào (mặc định) khi tạo một Hợp đồng Dịch vụ mới dựa trên một Mẫu Hợp đồng cụ thể.
Tiền điều kiện	<ul style="list-style-type: none"> <li>- BQL đã đăng nhập vào hệ thống với quyền quản lý mẫu hợp đồng.</li> <li>- Đã có ít nhất một mẫu hợp đồng dịch vụ được tạo (từ UC3.1).</li> <li>- Đã có các dịch vụ được khai báo trong hệ thống (từ UC2.1).</li> </ul>
Luồng chính	BQL truy cập chức năng "Quản lý mẫu hợp đồng" và chọn một mẫu hợp đồng cần chỉnh sửa (có thể từ UC3.1 hoặc UC3.3 - Xem danh sách mẫu hợp đồng đã tạo).

	<p>Hệ thống hiển thị thông tin chi tiết của mẫu hợp đồng, bao gồm mục quản lý các dịch vụ mặc định.</p> <p>BQL có thể:</p> <p>Thêm dịch vụ mặc định:</p> <ul style="list-style-type: none"><li>- BQL chọn "Thêm dịch vụ mặc định" từ danh sách các dịch vụ đã khai báo.</li><li>- Hệ thống hiển thị danh sách các dịch vụ hiện có (từ UC2.4).</li><li>- BQL chọn một hoặc nhiều dịch vụ để thêm vào danh sách mặc định của mẫu hợp đồng này.</li><li>- BQL xác nhận thêm.</li><li>- Hệ thống hiển thị dịch vụ đã được thêm vào danh sách dịch vụ mặc định của mẫu hợp đồng.</li></ul> <p>Xóa dịch vụ mặc định:</p> <ul style="list-style-type: none"><li>- BQL chọn một dịch vụ từ danh sách dịch vụ mặc định hiện có của mẫu hợp đồng.</li><li>- BQL chọn "Xóa".</li><li>- Hệ thống hiển thị hộp thoại xác nhận.</li><li>- BQL xác nhận xóa.</li><li>- Hệ thống loại bỏ dịch vụ đó khỏi danh sách mặc định của mẫu hợp đồng.</li></ul> <p>BQL xác nhận lưu các thay đổi đối với danh sách dịch vụ mặc định của mẫu hợp đồng.</p> <p>Hệ thống cập nhật mẫu hợp đồng với danh sách dịch vụ mặc định mới.</p> <p>Hệ thống hiển thị thông báo "Cập nhật dịch vụ mặc định thành công".</p>
--	--

Luồng thay thế	<p>AF1 – Dịch vụ đã tồn tại:</p> <p>Tại bước "Thêm dịch vụ mặc định", nếu BQL cố gắng thêm một dịch vụ đã có trong danh sách mặc định.</p> <p>Hệ thống hiển thị cảnh báo "Dịch vụ đã có trong danh sách mặc định".</p> <p>Luồng tiếp tục.</p> <p>AF2 – Không có dịch vụ nào để thêm:</p> <p>Tại bước "Hệ thống hiển thị danh sách các dịch vụ hiện có", nếu không có dịch vụ nào để chọn.</p> <p>Hệ thống hiển thị thông báo "Chưa có dịch vụ nào được khai báo để thêm".</p>
Hậu điều kiện	Danh sách các dịch vụ mặc định của mẫu hợp đồng đã được cập nhật.

Use Case UC3.3 – Xem danh sách mẫu hợp đồng đã tạo

Table 10: Đặc tả Xem danh sách mẫu hợp đồng đã tạo

Thành phần	Nội dung
Tên Use Case	UC3.3 – Xem danh sách mẫu hợp đồng đã tạo
Tác nhân chính	Ban Quản Lý
Mục tiêu	Cho phép BQL xem danh sách các mẫu hợp đồng hiện có, kèm điều khoản và phụ lục dịch vụ đã đính kèm
Tiền điều kiện	Đã có ít nhất một mẫu hợp đồng trong hệ thống
Luồng chính	<p>BQL chọn chức năng "Quản lý mẫu hợp đồng"</p> <p>Hệ thống hiển thị danh sách mẫu hợp đồng với các thông tin: tên, mô tả, trạng thái, số lượng phụ lục đính kèm</p>

## Báo cáo Thực tập Tốt nghiệp Đại học

	BQL có thể chọn "Xem chi tiết" mẫu bất kỳ để xem điều khoản và phụ lục cụ thể
Luồng thay thế	<p>AF1 – Không có dữ liệu:</p> <p>Tại bước 3 của Luồng chính (Hệ thống hiển thị danh sách tất cả mẫu hợp đồng).</p> <p>Nếu hệ thống không tìm thấy bất kỳ mẫu hợp đồng nào.</p> <p>Hệ thống hiển thị thông báo: "Chưa có mẫu hợp đồng nào được tạo trong hệ thống."</p> <p>BQL có thể quay lại màn hình chính hoặc thực hiện các thao tác khác.</p> <p>AF2 – Lỗi kết nối cơ sở dữ liệu:</p> <p>Tại bất kỳ bước nào yêu cầu truy xuất dữ liệu từ hệ thống.</p> <p>Hệ thống hiển thị thông báo lỗi: "Lỗi kết nối cơ sở dữ liệu. Vui lòng thử lại sau."</p> <p>Luồng kết thúc, BQL không thể xem danh sách mẫu hợp đồng.</p>
Hậu điều kiện	BQL xem được thông tin chi tiết từng mẫu hợp đồng

### Use Case UC3.4 – Chính sửa mẫu hợp đồng

Table 11: Chính sửa mẫu hợp đồng

Thành phần	Nội dung
Tên Use Case	UC3.4 – Chính sửa mẫu hợp đồng
Tác nhân chính	Ban Quản Lý
Mục tiêu	Cho phép chỉnh sửa nội dung điều khoản
Tiền điều kiện	Mẫu hợp đồng đã tồn tại và BQL có quyền chỉnh sửa
Luồng chính	BQL chọn mẫu hợp đồng cần chỉnh sửa Hệ thống hiển thị thông tin mẫu hợp đồng

Báo cáo Thực tập Tốt nghiệp Đại học

	<p>BQL sửa nội dung điều khoản chính hoặc điều chỉnh danh sách phụ lục</p> <p>Nhấn "Lưu"</p> <p><u>Hệ thống ghi nhận thay đổi</u></p>
Luồng thay thế	<p>AF1 – Thông tin chỉnh sửa không hợp lệ:</p> <p>Tại bước "Nhấn "Lưu thay đổi"" của Luồng chính.</p> <p>Nếu thông tin BQL nhập vào để chỉnh sửa mẫu hợp đồng không hợp lệ.</p> <p>Hệ thống hiển thị cảnh báo/lỗi bên cạnh các trường không hợp lệ và yêu cầu BQL nhập lại.</p> <p>Luồng quay lại bước "BQL chỉnh sửa thông tin".</p> <p>AF2 – Lỗi hệ thống khi cập nhật:</p> <p>Tại bước "Hệ thống cập nhật thông tin mẫu hợp đồng" của Luồng chính.</p> <p>Nếu xảy ra lỗi hệ thống không thể lưu các thay đổi.</p> <p>Hệ thống hiển thị thông báo lỗi: "Đã xảy ra lỗi khi cập nhật mẫu hợp đồng. Vui lòng thử lại."</p> <p>Luồng kết thúc, thay đổi không được lưu.</p>
Hậu điều kiện	Mẫu hợp đồng được cập nhật nội dung (nếu chưa áp dụng thực tế)

## Nhóm 4. Quản lý hợp đồng cư dân

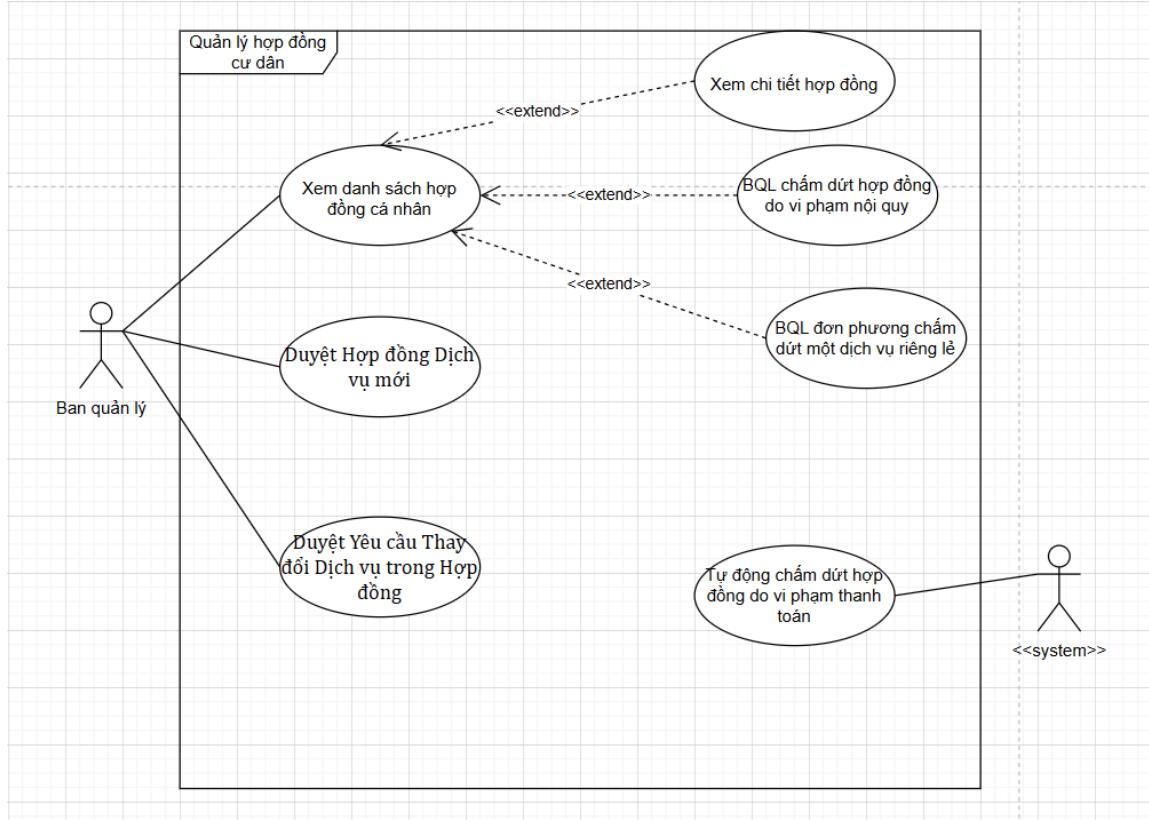


Figure 9: UC Quản lý hợp đồng cư dân

Use Case UC4.1 – Xem danh sách hợp đồng cá nhân

Table 12: Đặc tả Xem danh sách hợp đồng cá nhân

Thành phần	Nội dung
Tên Use Case	UC4.1 – Xem danh sách hợp đồng cá nhân
Tác nhân chính	Ban Quản Lý
Mục tiêu	Cho phép BQL theo dõi danh sách tất cả hợp đồng giữa cư dân và BQL
Tiền điều kiện	Hệ thống đã có các hợp đồng được ký kết
Luồng chính	<p>BQL truy cập chức năng "Quản lý hợp đồng cư dân"</p> <p>Hệ thống hiển thị danh sách hợp đồng gồm: tên cư dân, trạng thái hợp đồng, ngày ký, số dịch vụ đã đăng ký</p> <p>BQL có thể lọc theo trạng thái (đang hoạt động, đã chấm dứt, v.v.) hoặc tìm kiếm theo tên cư dân</p>

## Báo cáo Thực tập Tốt nghiệp Đại học

Luồng thay thế	<p>AF1 – Không có dữ liệu:</p> <p>Tại bước 3 của Luồng chính (Hệ thống hiển thị danh sách tất cả hợp đồng của cư dân).</p> <p>Nếu hệ thống không tìm thấy bất kỳ hợp đồng nào cho cư dân đó.</p> <p>Hệ thống hiển thị thông báo: "Cư dân này chưa có hợp đồng dịch vụ nào."</p> <p>Luồng kết thúc.</p> <p>AF2 – Lỗi kết nối cơ sở dữ liệu:</p> <p>Tại bất kỳ bước nào yêu cầu truy xuất dữ liệu từ hệ thống.</p> <p>Hệ thống hiển thị thông báo lỗi: "Lỗi kết nối cơ sở dữ liệu. Vui lòng thử lại sau."</p> <p>Luồng kết thúc, không thể xem danh sách hợp đồng.</p>
Hậu điều kiện	BQL xem được danh sách và trạng thái hợp đồng

### Use Case UC4.2 – Xem chi tiết hợp đồng cụ thể

Table 13: Đặc tả Xem chi tiết hợp đồng cụ thể

Thành phần	Nội dung
Tên Use Case	UC4.2 – Xem chi tiết hợp đồng cụ thể
Tác nhân chính	Ban Quản Lý
Mục tiêu	Cho phép BQL xem chi tiết điều khoản và phụ lục của một hợp đồng đã ký
Tiền điều kiện	Đã có hợp đồng trong hệ thống
Luồng chính	<p>BQL chọn hợp đồng từ danh sách</p> <p>Hệ thống hiển thị các thông tin: điều khoản chính, danh sách dịch vụ.</p> <p>BQL có thể tải xuống bản PDF nếu cần</p>

Luồng thay thế	<p>AF1 – Lỗi kết nối cơ sở dữ liệu:</p> <p>Tại bất kỳ bước nào yêu cầu truy xuất dữ liệu từ hệ thống.</p> <p>Hệ thống hiển thị thông báo lỗi: "Lỗi kết nối cơ sở dữ liệu. Vui lòng thử lại sau."</p> <p>Luồng kết thúc, không thể xem chi tiết hợp đồng.</p>
Hậu điều kiện	BQL xem được nội dung chi tiết hợp đồng

### Use Case UC4.3 – Tự động chấm dứt hợp đồng do vi phạm thanh toán

Table 14: Đặc tả Tự động chấm dứt hợp đồng do vi phạm thanh toán

Thành phần	Nội dung
Tên Use Case	UC4.3 – Tự động chấm dứt hợp đồng do vi phạm thanh toán
Tác nhân chính	Hệ thống
Mục tiêu	Hệ thống tự động chấm dứt hợp đồng nếu cư dân không thanh toán hóa đơn tổng sau thời hạn quy định
Tiền điều kiện	Hóa đơn tổng của cư dân đã quá hạn thanh toán theo điều khoản
Luồng chính	<p>Scheduler của hệ thống kiểm tra các hóa đơn quá hạn định kỳ</p> <p>Nếu hóa đơn chưa được thanh toán và đã quá số ngày quy định, hệ thống:</p> <ul style="list-style-type: none"> <li>- Đánh dấu hợp đồng chính là "Bị chấm dứt"</li> <li>- Chuyển tất cả phụ lục liên quan sang trạng thái "Ngừng hiệu lực"</li> </ul> <p>Ghi log và thông báo nội bộ cho BQL</p>
Luồng thay thế	EF1 – Lỗi trong quá trình xử lý một hợp đồng cụ thể:

	<p>Trong quá trình hệ thống duyệt và xử lý các hợp đồng (Vòng lặp trong Luồng chính).</p> <p>Nếu có lỗi xảy ra khi xử lý một hợp đồng cụ thể (ví dụ: không thể truy xuất thông tin thanh toán, lỗi khi cập nhật trạng thái).</p> <p>Hệ thống ghi lại chi tiết lỗi vào log hệ thống (error log) cùng với ID hợp đồng bị lỗi.</p> <p>Hệ thống bỏ qua hợp đồng đó và tiếp tục xử lý các hợp đồng tiếp theo trong danh sách.</p> <p>Ban Quản Lý sẽ cần kiểm tra log để xử lý thủ công các trường hợp lỗi này.</p> <p>EF2 – Không có hợp đồng nào đủ điều kiện chấm dứt:</p> <p>Sau bước "Hệ thống truy vấn và lọc danh sách các hợp đồng đủ điều kiện chấm dứt" của Luồng chính.</p> <p>Nếu không tìm thấy bất kỳ hợp đồng nào đáp ứng tiêu chí vi phạm thanh toán.</p> <p>Hệ thống ghi một thông báo vào log: "Không có hợp đồng nào đủ điều kiện tự động chấm dứt do vi phạm thanh toán trong kỳ này."</p> <p>Luồng kết thúc.</p>
Hậu điều kiện	Hợp đồng chính và toàn bộ phụ lục bị chấm dứt tự động

Use Case UC4.4 – BQL chấm dứt hợp đồng do vi phạm nội quy

Table 15: BQL chấm dứt hợp đồng do vi phạm nội quy

Thành phần	Nội dung
Tên Use Case	UC4.4 – BQL chấm dứt hợp đồng do vi phạm nội quy

Báo cáo Thực tập Tốt nghiệp Đại học

Tác nhân chính	Ban quản lý
Mục tiêu	Cho phép BQL chấm dứt hợp đồng nếu cư dân vi phạm quy định nội bộ nghiêm trọng
Tiền điều kiện	Cư dân đang có hợp đồng hoạt động và có hành vi vi phạm nội quy
Luồng chính	<p>BQL truy cập danh sách hợp đồng đang hoạt động</p> <p>Chọn hợp đồng vi phạm cần xử lý</p> <p>Nhập lý do chấm dứt (vi phạm nội quy, phá hoại tài sản, lạm dụng dịch vụ...)</p> <p>Xác nhận "Chấm dứt hợp đồng"</p> <p>Hệ thống cập nhật trạng thái hợp đồng và toàn bộ phụ lục sang "Ngừng hiệu lực"</p>
Luồng thay thế	<p>AF1 – Lỗi hệ thống khi cập nhật trạng thái:</p> <p>Tại bước 5 của Luồng chính (Hệ thống cập nhật trạng thái hợp đồng thành "Đã chấm dứt").</p> <p>Nếu xảy ra lỗi hệ thống không thể cập nhật trạng thái.</p> <p>Hệ thống hiển thị thông báo lỗi: "Đã xảy ra lỗi khi chấm dứt hợp đồng. Vui lòng thử lại."</p> <p>Trạng thái hợp đồng không được thay đổi.</p> <p>Luồng kết thúc.</p>
Hậu điều kiện	Hợp đồng và tất cả dịch vụ bị chấm dứt, log xử lý được lưu lại

Use Case UC4.5 – BQL đơn phương chấm dứt một dịch vụ riêng lẻ

Table 16: Đặc tả BQL đơn phương chấm dứt một dịch vụ riêng lẻ

Thành phần	Nội dung
Tên Use Case	UC4.5 – BQL đơn phương chấm dứt một dịch vụ riêng lẻ
Tác nhân chính	Ban quản lý

Báo cáo Thực tập Tốt nghiệp Đại học

Mục tiêu	Cho phép BQL ngừng cung cấp một dịch vụ cụ thể cho cư dân nếu phát hiện vi phạm trong phạm vi dịch vụ đó
Tiền điều kiện	Cư dân đang có phụ lục dịch vụ đang hoạt động và có hành vi vi phạm quy định dịch vụ
Luồng chính	<p>BQL đang xem chi tiết một Hợp đồng Dịch vụ ( thông qua UC4.2 - Xem chi tiết Hợp đồng Dịch vụ).</p> <p>Trên giao diện chi tiết hợp đồng, BQL xác định Dịch vụ cụ thể mà họ muốn chấm dứt.</p> <p>BQL chọn biểu tượng "Chấm dứt" bên cạnh Dịch vụ đó.</p> <p>Hệ thống hiển thị hộp thoại xác nhận (có thể yêu cầu BQL nhập lý do chấm dứt).</p> <p>BQL nhập lý do chấm dứt dịch vụ này.</p> <p>BQL xác nhận hành động.</p> <p>Hệ thống cập nhật trạng thái của Dịch vụ đó trong bản ghi Hợp đồng Dịch vụ thành "Đã chấm dứt" (hoặc "Không hoạt động").</p> <p>Hệ thống ghi lại lịch sử thay đổi này (người thực hiện, thời gian, lý do).</p> <p>Hệ thống thông báo cho cư dân liên quan về việc chấm dứt dịch vụ này.</p>
Luồng thay thế	<p>AF1 – Dịch vụ không thể chấm dứt:</p> <ul style="list-style-type: none"> <li>- Nếu dịch vụ được chọn là dịch vụ bắt buộc của hợp đồng.</li> <li>- Hệ thống hiển thị thông báo lỗi: “Dịch vụ này là dịch vụ bắt buộc và không thể chấm dứt riêng lẻ.” và không cho phép chấm dứt.</li> </ul>
Hậu điều kiện	Phụ lục dịch vụ bị chấm dứt, các dịch vụ khác vẫn duy trì bình thường

Use Case: UC4.6 – Duyệt Hợp đồng Dịch vụ mới

Báo cáo Thực tập Tốt nghiệp Đại học

Table 17: **Đặc tả Duyệt Hợp đồng Dịch vụ mới**

Thành phần	Nội dung
Tên Use Case	UC4.6 –Duyệt Hợp đồng Dịch vụ mới
Tác nhân chính	Ban quản lý
Mục tiêu	BQL xem xét, kiểm tra và phê duyệt hoặc từ chối một Hợp đồng Dịch vụ mới do cư dân đã khởi tạo.
Tiền điều kiện	<ul style="list-style-type: none"> <li>- BQL đã đăng nhập vào hệ thống với quyền quản lý.</li> <li>- Có ít nhất một Yêu cầu Thay đổi Hợp đồng với LoạiThayDoi là "Tạo mới Hợp đồng" đang ở trạng thái "Chờ duyệt".</li> </ul>
Luồng chính	<p>BQL truy cập chức năng "Quản lý Yêu cầu Chờ Duyệt" hoặc "Duyệt Hợp đồng mới".</p> <p>Hệ thống hiển thị danh sách các Yêu cầu Thay đổi Hợp đồng đang chờ duyệt (có thể lọc theo loại "Tạo mới Hợp đồng").</p> <p>BQL chọn một yêu cầu cụ thể để xem thông tin chi tiết của Hợp đồng Dịch vụ mới được đề xuất.</p> <p>Hệ thống hiển thị các thông tin cần thiết để duyệt: thông tin cư dân liên quan, thông tin căn hộ, các Dịch vụ được đăng ký trong hợp đồng mới cùng các thông số chi tiết của chúng, và các điều khoản hợp đồng.</p> <p>BQL xem xét kỹ lưỡng thông tin, kiểm tra tính hợp lệ và phù hợp.</p> <ul style="list-style-type: none"> <li>- (Nếu cần xem toàn bộ ngũ cảnh hợp đồng hiện tại hoặc các thông tin khác, BQL có thể điều hướng đến UC4.2 – Xem chi tiết Hợp đồng Dịch vụ thông qua một liên kết được cung cấp).</li> </ul> <p>BQL quyết định Duyệt hoặc Từ chối hợp đồng.</p> <p>Nếu Duyệt:</p>

Báo cáo Thực tập Tốt nghiệp Đại học

	<ul style="list-style-type: none"> <li>+ BQL xác nhận phê duyệt.</li> <li>+ Hệ thống cập nhật trạng thái của bản ghi Hợp đồng Dịch vụ từ "Chờ BQL duyệt" thành "Đang hoạt động".</li> <li>+ Hệ thống cập nhật trạng thái của Yêu cầu Thay đổi Hợp đồng tương ứng thành "Đã duyệt".</li> <li>+ Hệ thống gửi thông báo xác nhận Hợp đồng Dịch vụ đã được kích hoạt cho Cư dân liên quan.</li> <li>+ Hệ thống ghi lại lịch sử duyệt (người duyệt, thời gian, trạng thái).</li> </ul>
Luồng thay thế	<p>AF1 – BQL Từ chối yêu cầu:</p> <ul style="list-style-type: none"> <li>- Tại bước 6 của Luồng chính, thay vì Duyệt, BQL chọn Từ chối.</li> <li>- BQL nhập lý do từ chối.</li> <li>- Hệ thống cập nhật trạng thái của bản ghi Hợp đồng Dịch vụ thành "Bị từ chối" hoặc "Đã hủy".</li> <li>- Hệ thống cập nhật trạng thái của Yêu cầu Thay đổi Hợp đồng tương ứng thành "Đã từ chối".</li> <li>- Hệ thống gửi thông báo kèm lý do từ chối cho Cư dân liên quan.</li> <li>- Hệ thống ghi lại lịch sử từ chối.</li> </ul> <p>AF2 – Thông tin không hợp lệ: Nếu BQL phát hiện thông tin không đủ hoặc không chính xác, họ có thể từ chối yêu cầu.</p>
Hậu điều kiện	<ul style="list-style-type: none"> <li>- Hợp đồng Dịch vụ mới được tạo có trạng thái "Đang hoạt động" hoặc "Bị từ chối", và Yêu cầu Thay đổi Hợp đồng được cập nhật trạng thái tương ứng.</li> <li>- Cư dân nhận được thông báo về kết quả duyệt.</li> </ul>

Báo cáo Thực tập Tốt nghiệp Đại học

Use Case: UC4.7 – Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng

Table 18: **Đặc tả Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng**

Thành phần	Nội dung
Tên Use Case	UC4.7 – Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng
Tác nhân chính	Ban quản lý
Mục tiêu	BQL xem xét và phê duyệt hoặc từ chối yêu cầu của cư dân về việc thêm, sửa đổi, hoặc hủy bỏ một hoặc nhiều Dịch vụ trong một Hợp đồng Dịch vụ hiện có.
Tiền điều kiện	<ul style="list-style-type: none"> <li>- BQL đã đăng nhập vào hệ thống với quyền quản lý.</li> <li>- Có ít nhất một Yêu cầu Thay đổi Hợp đồng với LoạiThayDoi là "Thêm Dịch vụ", "Sửa đổi Dịch vụ", hoặc "Hủy Dịch vụ" đang ở trạng thái "Chờ duyệt".</li> </ul>
Luồng chính	<p>BQL truy cập chức năng "Quản lý Yêu cầu Chờ Duyệt" hoặc "Duyệt Yêu cầu Thay đổi Dịch vụ".</p> <p>Hệ thống hiển thị danh sách các Yêu cầu Thay đổi Hợp đồng đang chờ duyệt (có thể lọc theo loại thay đổi).</p> <p>BQL chọn một yêu cầu cụ thể để xem chi tiết yêu cầu thay đổi.</p> <p>Hệ thống hiển thị các thông tin cần thiết để duyệt yêu cầu thay đổi: Hợp đồng Dịch vụ bị ảnh hưởng, cư dân gửi yêu cầu, loại thay đổi, và chi tiết cụ thể các Dịch vụ được đề xuất thêm/sửa/hủy (bao gồm thông số, giá...).</p> <p>- (Nếu cần xem toàn bộ ngữ cảnh hợp đồng hiện tại hoặc các thông tin khác, BQL có thể điều hướng đến UC4.2 – Xem chi tiết Hợp đồng Dịch vụ thông qua một liên kết được cung cấp).</p> <p>BQL xem xét kỹ lưỡng yêu cầu, kiểm tra tính hợp lệ và phù hợp.</p>

	<p>BQL quyết định Duyệt hoặc Từ chối yêu cầu.</p> <p>Nếu Duyệt:</p> <ul style="list-style-type: none"> <li>+ BQL xác nhận phê duyệt.</li> <li>+ Hệ thống thực hiện cập nhật tương ứng trên bản ghi Hợp đồng Dịch vụ (ví dụ: thêm các dòng dịch vụ mới, cập nhật thông số dịch vụ, hoặc đánh dấu dịch vụ là "Đã chấm dứt").</li> <li>+ Hệ thống cập nhật trạng thái của Yêu cầu Thay đổi Hợp đồng thành "Đã duyệt".</li> <li>+ Hệ thống gửi thông báo xác nhận thay đổi đã được áp dụng cho Cư dân liên quan.</li> <li>+ Hệ thống ghi lại lịch sử duyệt.</li> </ul>
Luồng thay thế	<p>AF1 – BQL Từ chối yêu cầu:</p> <ul style="list-style-type: none"> <li>- Tại bước 6 của Luồng chính, thay vì Duyệt, BQL chọn Từ chối.</li> <li>- BQL nhập lý do từ chối.</li> <li>- Hệ thống cập nhật trạng thái của Yêu cầu Thay đổi Hợp đồng thành "Đã từ chối".</li> <li>- Hệ thống gửi thông báo kèm lý do từ chối cho Cư dân liên quan.</li> <li>- Hệ thống ghi lại lịch sử từ chối.</li> </ul> <p>AF2 – Yêu cầu không hợp lệ: Yêu cầu thêm dịch vụ không tồn tại, hoặc hủy dịch vụ không có trong hợp đồng.</p>
Hậu điều kiện	<ul style="list-style-type: none"> <li>- Hợp đồng Dịch vụ mới được tạo có trạng thái "Đang hoạt động" hoặc "Bị từ chối", và Yêu cầu Thay đổi Hợp đồng được cập nhật trạng thái tương ứng.</li> <li>- Cư dân nhận được thông báo về kết quả duyệt.</li> </ul>

## Nhóm 5. Quản lý hóa đơn và chỉ số điện/nước

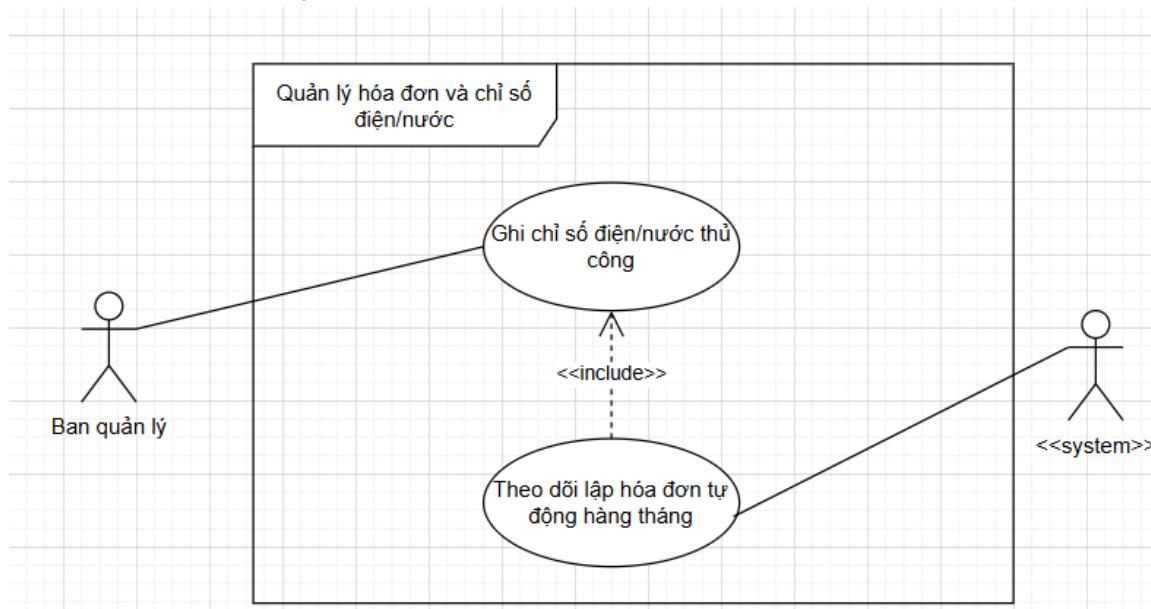


Figure 10: Quản lý hóa đơn và chỉ số điện/nước

Use Case UC5.1 – Theo dõi lập hóa đơn tự động hàng tháng

Table 19: Đặc tả Theo dõi lập hóa đơn tự động hàng tháng

Thành phần	Nội dung
Tên Use Case	UC5.1 – Theo dõi lập hóa đơn tự động hàng tháng
Tác nhân chính	Hệ thống
Mục tiêu	Hệ thống tự động tạo hóa đơn hàng tháng cho các hợp đồng đang hoạt động, gộp tất cả dịch vụ cư dân đã đăng ký
Tiền điều kiện	Cư dân có hợp đồng và dịch vụ còn hiệu lực, hệ thống có thông tin chỉ số điện nước nếu cần
Luồng chính	<p>Vào thời điểm định kỳ (ví dụ ngày 1 hàng tháng), scheduler được kích hoạt</p> <p>Hệ thống lấy danh sách hợp đồng đang hoạt động</p> <p>Với mỗi hợp đồng:</p> <ul style="list-style-type: none"> <li>- Tính toán chi phí cho từng dịch vụ</li> </ul>

Báo cáo Thực tập Tốt nghiệp Đại học

	<ul style="list-style-type: none"> <li>- Lấy chỉ số điện, nước nếu có (từ UC5.3 hoặc thiết bị tự động)</li> <li>- Gộp lại thành một hóa đơn tổng</li> </ul> <p>Hệ thống lưu hóa đơn, cập nhật trạng thái "Chưa thanh toán"</p> <p>Gửi thông tin hóa đơn đến hệ thống thông báo (do nhóm khác phụ trách)</p> <p>Log quá trình lập hóa đơn để BQL kiểm tra nếu cần</p>
Luồng thay thế	<p>AF1 – Thiếu dữ liệu chỉ số điện/nước: (Đã có, chỉ làm rõ hơn)</p> <p>Trong quá trình hệ thống tính toán chi phí cho từng dịch vụ và lấy chỉ số điện/nước (bước "Lấy chỉ số điện, nước (Nếu có)" của Luồng chính).</p> <p>Nếu hệ thống không thể lấy được chỉ số điện/nước cần thiết cho một dịch vụ cụ thể của một hợp đồng (do chưa nhập, lỗi thiết bị, v.v.).</p> <p>Hệ thống sẽ ghi nhận chi phí của dịch vụ đó là 0 hoặc đánh dấu là thiếu dữ liệu.</p> <p>Hệ thống vẫn tiếp tục gộp các dịch vụ khác vào hóa đơn tổng.</p> <p>Hóa đơn được tạo sẽ được đánh dấu trạng thái "Chờ chỉ số" hoặc "Thiếu thông tin" (thay vì "Chưa thanh toán").</p> <p>Hệ thống gửi cảnh báo nội bộ đến BQL hoặc module quản lý chỉ số để yêu cầu bổ sung dữ liệu thủ công.</p> <p>Hệ thống tiếp tục xử lý các hợp đồng khác.</p> <p>AF2 – Không có hợp đồng nào đang hoạt động:</p> <p>Sau bước "Lấy danh sách hợp đồng đang hoạt động" của Luồng chính.</p>

	<p>Nếu hệ thống không tìm thấy bất kỳ hợp đồng nào ở trạng thái "Đang hoạt động".</p> <p>Hệ thống ghi một thông báo vào log: "Không có hợp đồng nào đang hoạt động để lập hóa đơn trong kỳ này."</p> <p>Luồng kết thúc.</p>
Hậu điều kiện	Hóa đơn tổng cho mỗi cư dân được tạo mỗi tháng, sẵn sàng để theo dõi thanh toán

### Use Case UC5.2 – Ghi chỉ số điện/nước thủ công

**Table 20: Đặc tả Ghi chỉ số điện/nước thủ công**

Thành phần	Nội dung
Tên Use Case	UC5.2 – Ghi chỉ số điện/nước thủ công
Tác nhân chính	Ban quản lý
Mục tiêu	BQL nhập chỉ số điện/nước hàng tháng thủ công cho cư dân để phục vụ tính hóa đơn
Tiền điều kiện	Thiết bị đo đã hoạt động, cư dân đang sử dụng dịch vụ điện/nước
Luồng chính	<p>BQL truy cập chức năng "Ghi chỉ số điện/nước"</p> <p>Chọn tháng và chọn cư dân cần nhập</p> <p>Hệ thống hiển thị chỉ số tháng trước (nếu có) và ô nhập cho tháng hiện tại</p> <p>BQL nhập chỉ số mới, xác nhận lưu</p> <p>Hệ thống tính sản lượng sử dụng = chỉ số mới - cũ và lưu vào hệ thống</p>
Luồng thay thế	<p>AF1 – Chỉ số không hợp lệ: (Đã có, chỉ làm rõ hơn)</p> <p>Tại bước "BQL nhập chỉ số mới, xác nhận lưu" của Luồng chính (hoặc sau khi BQL xác nhận lưu).</p>

	<p>Nếu chỉ số mới mà BQL nhập vào nhỏ hơn chỉ số cũ đã ghi nhận trước đó cho cùng một kỳ hoặc một lỗi logic nào đó.</p> <p>Hệ thống hiển thị cảnh báo lỗi: "Chỉ số không hợp lệ. Chỉ số mới phải lớn hơn hoặc bằng chỉ số cũ. Vui lòng kiểm tra lại."</p> <p>Luồng quay lại bước "BQL nhập chỉ số mới" để BQL có thể nhập lại.</p> <p>AF3 – Lỗi hệ thống khi lưu chỉ số:</p> <p>Tại bước "Hệ thống tính sản lượng sử dụng = chỉ số mới - cũ và lưu vào hệ thống" của Luồng chính.</p> <p>Nếu xảy ra lỗi hệ thống (ví dụ: mất kết nối, lỗi cơ sở dữ liệu) không thể lưu chỉ số mới.</p> <p>Hệ thống hiển thị thông báo lỗi: "Đã xảy ra lỗi khi lưu chỉ số. Vui lòng thử lại."</p> <p>Luồng kết thúc, chỉ số không được lưu.</p>
Hậu điều kiện	Chỉ số được lưu đúng cho từng dịch vụ, phục vụ tính hóa đơn tự động

### Use Case UC5.3 – Xác nhận thanh toán

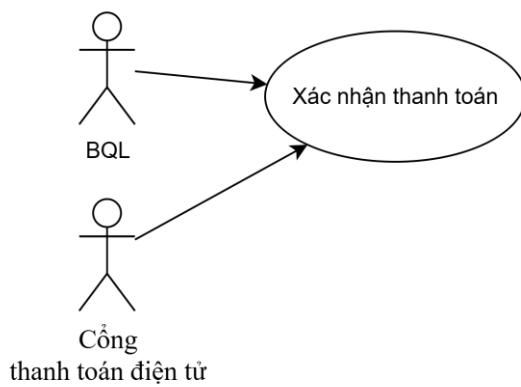


Figure 11: Xác nhận thanh toán

### Use Case UC5.3 – Xác nhận thanh toán

Table 21: Đặc tả Xác nhận thanh toán

Use Case ID	UC5.3

<b>Use Case Name</b>	Xác nhận thanh toán
<b>Description (Tóm tắt mục tiêu nghiệp vụ của Use Case):</b>	Ban quản lý xác nhận thanh toán của cư dân, bao gồm thanh toán thủ công hoặc online.
<b>Actor(s)</b>	Ban quản lý, Hệ thống, Cổng thanh toán điện tử
<b>Priority</b>	Cao
<b>Trigger (Điều kiện khởi phát Use Case)</b>	Ban quản lý nhận được yêu cầu xác nhận thanh toán từ cư dân.
<b>Pre-condition(s) (Điều kiện tiên quyết):</b>	Cư dân đã thực hiện thanh toán và hệ thống nhận được yêu cầu xác nhận thanh toán. Ban quản lý truy cập giao diện quản lý hợp đồng
<b>Post-condition(s) (Kết quả sau khi hoàn thành):</b>	Thanh toán của cư dân được xác nhận và hóa đơn được cập nhật.
<b>Basic Flow (Luồng chính):</b>	<ul style="list-style-type: none"> <li>o Ban quản lý chọn hợp đồng cần xác nhận thanh toán.</li> <li>o Hệ thống kiểm tra tình trạng thanh toán của cư dân.</li> <li>o Ban quản lý kiểm tra biên lai thanh toán và xác nhận.</li> <li>o Hệ thống cập nhật trạng thái hóa đơn và gửi thông báo xác nhận thanh toán đến cư dân.</li> </ul>
<b>Alternative Flow(s) (Luồng thay thế nếu có):</b>	Nếu thanh toán online, hệ thống xác nhận thông qua cổng thanh toán điện tử.
<b>Exception Flow(s) (Luồng ngoại lệ nếu có):</b>	Nếu có lỗi trong quá trình xác nhận thanh toán, hệ thống sẽ thông báo lỗi và yêu cầu thử lại.
<b>Business Rules (Quy tắc nghiệp vụ liên quan)</b>	<ul style="list-style-type: none"> <li>• Thanh toán phải được xác nhận trước khi hợp đồng có hiệu lực.</li> <li>• Thanh toán thủ công phải có biên lai hợp lệ hoặc chứng từ thanh toán.</li> </ul>
<b>Non-Functional Requirements (Yêu cầu phi chức năng nếu có)</b>	Quá trình xác nhận thanh toán phải được thực hiện trong thời gian không quá 5 giây đối với thanh toán online và không quá 10 giây đối với thanh toán thủ công.

#### Nhóm 6: Quản lý tài khoản cá nhân (Cư dân)

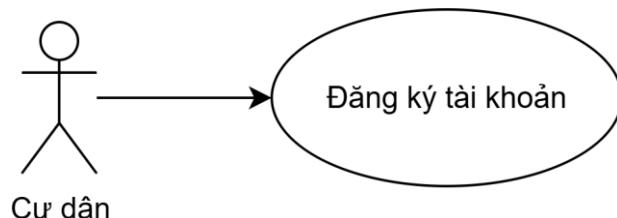


Figure 12: UC đăng ký tài khoản

Table 22: ĐẶC TẢ UC ĐĂNG KÝ TÀI KHOẢN

<b>Use Case ID</b>	UC6.1
<b>Use Case Name</b>	Đăng ký tài khoản
<b>Description (Tóm tắt mục tiêu nghiệp vụ của Use Case):</b>	1. Cư dân tạo tài khoản mới trên hệ thống để sử dụng các dịch vụ.
<b>Actor(s)</b>	Cư dân, Hệ thống
<b>Priority</b>	Cao

<b>Trigger (Điều kiện khởi phát Use Case)</b>	Cư dân muốn tạo tài khoản mới.
<b>Pre-condition(s) (Điều kiện tiên quyết):</b>	Cư dân chưa có tài khoản trong hệ thống. Cư dân đã truy cập vào trang đăng ký tài khoản
<b>Post-condition(s) (Kết quả sau khi hoàn thành):</b>	Tài khoản mới được tạo thành công và chờ ban quản lý duyệt
<b>Basic Flow (Luồng chính):</b>	<ul style="list-style-type: none"> <li>o Cư dân nhập thông tin cần thiết (tên, email, số điện thoại, mật khẩu, v.v.).</li> <li>o Hệ thống xác nhận thông tin và tạo tài khoản.</li> <li>o Cư dân nhận thông báo tài khoản đã được tạo thành công.</li> <li>o Hệ thống quay trở về màn hình đăng nhập</li> <li>o Hệ thống thông báo tài khoản đang chờ BQL duyệt.</li> </ul>
<b>Alternative Flow(s) (Luồng thay thế nếu có):</b>	Nếu thông tin không hợp lệ (email đã tồn tại, mật khẩu yếu), hệ thống sẽ yêu cầu cư dân sửa lại thông tin.
<b>Exception Flow(s) (Luồng ngoại lệ nếu có):</b>	Nếu hệ thống gặp lỗi trong quá trình tạo tài khoản, hệ thống sẽ thông báo lỗi và yêu cầu thử lại.
<b>Business Rules (Quy tắc nghiệp vụ liên quan)</b>	<ul style="list-style-type: none"> <li>• Mật khẩu phải chứa ít nhất 8 ký tự, bao gồm chữ hoa, chữ thường, và số.</li> <li>• Email phải có định dạng hợp lệ và chưa được sử dụng.</li> </ul>
<b>Non-Functional Requirements (Yêu cầu phi chức năng nếu có)</b>	Hệ thống phải xử lý việc đăng ký tài khoản trong vòng 3 giây.

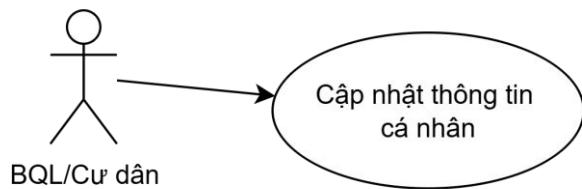


Figure 13: UC Cập nhật thông tin cá nhân

Table 23: ĐẶC TẢ UC CẬP NHẬT THÔNG TIN CÁ NHÂN

<b>Use Case ID</b>	UC6.2
<b>Use Case Name</b>	Cập nhật thông tin cá nhân
<b>Description (Tóm tắt mục tiêu nghiệp vụ của Use Case):</b>	1. Cư dân hoặc Ban quản lý cập nhật thông tin cá nhân của họ trong hệ thống.
<b>Actor(s)</b>	Cư dân, Ban quản lý, Hệ thống
<b>Priority</b>	Cao
<b>Trigger (Điều kiện khởi phát Use Case)</b>	1. Người dùng (cư dân hoặc ban quản lý) đã đăng nhập và muốn chỉnh sửa thông tin cá nhân.
<b>Pre-condition(s) (Điều kiện tiên quyết):</b>	<p>Người dùng đã đăng nhập vào hệ thống và có quyền cập nhật thông tin cá nhân.</p> <p>Người dùng đã truy cập giao diện chỉnh sửa thông tin cá nhân</p>

<b>Post-condition(s) (Kết quả sau khi hoàn thành):</b>	Thông tin cá nhân của người dùng được cập nhật trong hệ thống.
<b>Basic Flow (Luồng chính):</b>	<ul style="list-style-type: none"> <li>o Người dùng chỉnh sửa thông tin cá nhân cần thay đổi (số điện thoại, địa chỉ, v.v.).</li> <li>o Người dùng nhấn nút “Thay đổi”.</li> <li>o Hệ thống kiểm tra thông tin (Định dạng) và tính hợp lệ dữ liệu thành công.</li> <li>o Hệ thống lưu lại các thay đổi và thông báo cập nhật thành công.</li> </ul>
<b>Alternative Flow(s) (Luồng thay thế nếu có):</b>	Không có
<b>Exception Flow(s) (Luồng ngoại lệ nếu có):</b>	Nếu thông tin cập nhật không hợp lệ (số điện thoại sai, địa chỉ không đúng định dạng), hệ thống sẽ thông báo lỗi và yêu cầu chỉnh sửa lại.
<b>Business Rules (Quy tắc nghiệp vụ liên quan)</b>	<ul style="list-style-type: none"> <li>• Thông tin cá nhân của người dùng phải được bảo mật và không chia sẻ cho bên thứ ba.</li> <li>• Thông tin được cập nhật phải hợp lệ và đúng định dạng.</li> </ul>
<b>Non-Functional Requirements (Yêu cầu phi chức năng nếu có)</b>	<ul style="list-style-type: none"> <li>• Quá trình cập nhật thông tin cá nhân không được vượt quá 5 giây.</li> </ul>

- Quá trình cập nhật thông tin cá nhân không được vượt quá 5 giây.

#### Nhóm 7: Đăng ký dịch vụ hợp đồng

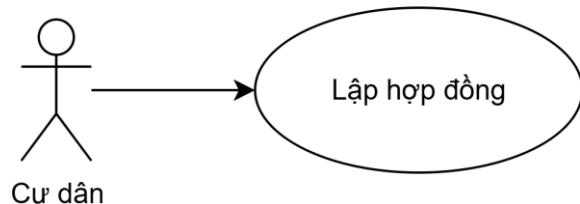


Figure 14: Use Case: Lập hợp đồng

Table 24: Đặc tả UC lập hợp đồng

<b>Use Case ID</b>	UC7.1
<b>Use Case Name</b>	Lập hợp đồng
<b>Description (Tóm tắt mục tiêu nghiệp vụ của Use Case):</b>	Cư dân tạo hợp đồng mới và chọn dịch vụ cần sử dụng
<b>Actor(s)</b>	Cư dân, Hệ thống
<b>Priority</b>	Cao
<b>Trigger (Điều kiện khởi phát Use Case)</b>	Cư dân muốn tạo hợp đồng mới và chọn dịch vụ.
<b>Pre-condition(s) (Điều kiện tiên quyết):</b>	Cư dân đã đăng nhập vào hệ thống. Cư dân đã truy cập vào trang quản lý hợp đồng.
<b>Post-condition(s) (Kết quả sau khi hoàn thành):</b>	Hợp đồng mới được tạo và dịch vụ được chọn.

<b>Basic Flow (Luồng chính):</b>	<ul style="list-style-type: none"> <li>o Cư dân click tạo hợp đồng mới.</li> <li>o Hệ thống kiểm tra quyền hạn của cư dân thành công</li> <li>o Cư dân chọn mẫu hợp đồng</li> <li>o Hệ thống tự động chọn các dịch vụ bắt buộc</li> <li>o Cư dân chọn dịch vụ thêm (wifi,...).</li> <li>o Cư dân nhập thông tin yêu cầu (căn hộ, thời gian hợp đồng, các điều khoản, v.v.).</li> <li>o Cư dân xác nhận đồng ý các điều khoản.</li> <li>o Hệ thống kiểm tra tính hợp lệ của hợp đồng thành công</li> <li>o Hệ thống tạo hợp đồng mới và lưu lại thông tin dịch vụ đã chọn.</li> </ul>
<b>Alternative Flow(s) (Luồng thay thế nếu có):</b>	Không có
<b>Exception Flow(s) (Luồng ngoại lệ nếu có):</b>	<ul style="list-style-type: none"> <li>• Nếu có lỗi trong quá trình tạo hợp đồng (ví dụ: lỗi kết nối, dữ liệu không hợp lệ), hệ thống sẽ thông báo lỗi và yêu cầu thử lại.</li> <li>• Nếu cư dân không đủ quyền hạn lập hợp đồng thì chuyển đến trang Hotline:..., để yêu cầu ban quản lý cấp quyền</li> </ul>
<b>Business Rules (Quy tắc nghiệp vụ liên quan)</b>	<ul style="list-style-type: none"> <li>• Dịch vụ phải có sẵn trong hệ thống và có thể lựa chọn cho hợp đồng.</li> <li>• Cư dân phải được ban quản lý cấp quyền đủ mới có thể lập hợp đồng trên hệ thống</li> <li>• Dịch vụ bắt buộc phải có: thuê nhà, điện, nước, rác sinh hoạt, ...</li> <li>• Hệ thống phải kiểm tra tính hợp lệ của các thông tin yêu cầu của hợp đồng.</li> <li>• Các dịch vụ bắt buộc không thể sửa.</li> <li>• Hệ thống phải kiểm tra tính hợp lệ của hợp đồng như không bỏ trống các thông tin cần thiết, không sai định dạng, ....</li> </ul>
<b>Non-Functional Requirements (Yêu cầu phi chức năng nếu có)</b>	<ul style="list-style-type: none"> <li>• Quá trình cập nhật thông tin cá nhân không được vượt quá 5 giây.</li> </ul>

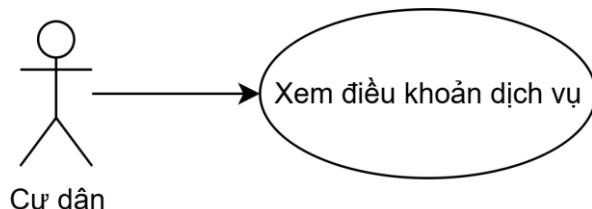


Figure 15: Xem điều khoản dịch vụ

Table 25: Đặc tả UC Xem điều khoản dịch vụ

Use Case ID	UC7.2
Use Case Name	Xem điều khoản dịch vụ

## Báo cáo Thực tập Tốt nghiệp Đại học

<b>Description (Tóm tắt mục tiêu nghiệp vụ của Use Case):</b>	Cư dân xem chi tiết các điều khoản của dịch vụ trước khi ký hợp đồng.
<b>Actor(s)</b>	Cư dân, Hệ thống
<b>Priority</b>	Trung bình
<b>Trigger (Điều kiện khởi phát Use Case)</b>	Cư dân muốn xem chi tiết điều khoản của dịch vụ trước khi ký hợp đồng
<b>Pre-condition(s) (Điều kiện tiên quyết):</b>	Cư dân đã chọn dịch vụ cần xem điều khoản. Cư dân đã truy cập vào trang điều khoản dịch vụ tại homepage hoặc tại chi tiết dịch vụ trong hợp đồng.
<b>Post-condition(s) (Kết quả sau khi hoàn thành):</b>	Cư dân xem được điều khoản dịch vụ và quyết định ký hợp đồng.
<b>Basic Flow (Luồng chính):</b>	<ul style="list-style-type: none"> <li>o Cư dân chọn dịch vụ cần xem.</li> <li>o Hệ thống hiển thị các điều khoản chi tiết của dịch vụ.</li> </ul>
<b>Alternative Flow(s) (Luồng thay thế nếu có):</b>	Không có
<b>Exception Flow(s) (Luồng ngoại lệ nếu có):</b>	Nếu có lỗi trong việc tải điều khoản dịch vụ, hệ thống sẽ thông báo lỗi và yêu cầu thử lại.
<b>Business Rules (Quy tắc nghiệp vụ liên quan)</b>	Điều khoản dịch vụ phải rõ ràng, đầy đủ và hợp pháp.
<b>Non-Functional Requirements (Yêu cầu phi chức năng nếu có)</b>	Điều khoản dịch vụ phải được hiển thị trong vòng 3 giây.

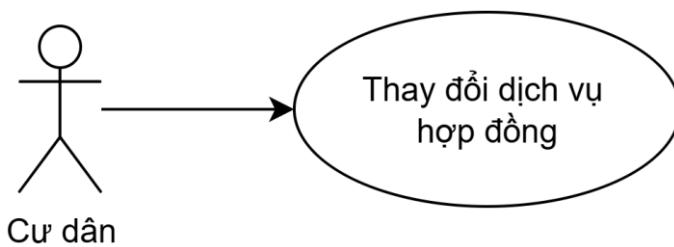


Figure 16: Thay đổi dịch vụ hợp đồng

Table 26: Thay đổi dịch vụ hợp đồng

<b>Use Case ID</b>	UC7.3
<b>Use Case Name</b>	Chỉnh sửa dịch vụ hợp đồng
<b>Description (Tóm tắt mục tiêu nghiệp vụ của Use Case):</b>	1. Cư dân chỉnh sửa dịch vụ trong hợp đồng nếu có thay đổi.
<b>Actor(s)</b>	Cư dân, Hệ thống
<b>Priority</b>	Trung bình
<b>Trigger (Điều kiện khởi phát Use Case)</b>	Cư dân đã có hợp đồng và dịch vụ muốn thay đổi. Cư dân đã truy cập vào trang quản lý hợp đồng và chọn hợp đồng muốn sửa.
<b>Pre-condition(s) (Điều kiện tiên quyết):</b>	Hệ thống thêm các dịch vụ trong hợp đồng trong trạng thái chờ duyệt,
<b>Post-condition(s) (Kết quả sau khi hoàn thành):</b>	Cư dân xem được điều khoản dịch vụ và quyết định ký hợp đồng.

<p><b>Basic Flow (Luồng chính):</b></p>	<ul style="list-style-type: none"> <li>o Cư dân chọn loại thay đổi hợp đồng.</li> <li>o Hệ thống lấy các loại thay đổi: thay đổi gói cước, thêm dịch vụ mới, thay đổi nội dung dịch vụ đang đăng ký (ngày kết thúc), hủy dịch vụ.</li> <li>o Cư dân chỉnh sửa/thêm dịch vụ trong hợp đồng.</li> <li>o Cư dân nhập các thông tin cần thiết.</li> <li>o Cư dân xác nhận đồng ý các điều khoản và click “Cập nhật”.</li> <li>o Hệ thống kiểm tra tính hợp lệ của hợp đồng thành công.</li> <li>o Hệ thống cập nhật thông tin hợp đồng với dịch vụ đã thay đổi.</li> <li>o Hệ thống gửi thông báo đến BQL để chờ duyệt.</li> </ul>
<p><b>Alternative Flow(s) (Luồng thay thế nếu có):</b></p>	<p>Không có</p>
<p><b>Exception Flow(s) (Luồng ngoại lệ nếu có):</b></p>	<ul style="list-style-type: none"> <li>• Nếu dịch vụ thay đổi không được phép, hệ thống sẽ thông báo lỗi và yêu cầu cư dân chọn dịch vụ hợp lệ</li> <li>• Nếu có lỗi trong quá trình cập nhật hợp đồng, hệ thống sẽ thông báo và yêu cầu thử lại.</li> </ul>
<p><b>Business Rules (Quy tắc nghiệp vụ liên quan)</b></p>	<ul style="list-style-type: none"> <li>• Dịch vụ thay đổi phải trong phạm vi hợp đồng và không vi phạm các điều khoản đã ký.</li> <li>• Hệ thống phải kiểm tra tính hợp lệ của hợp đồng như không bỏ trống các thông tin cần thiết, không sai định dạng, ....</li> <li>• Thay đổi dịch vụ chỉ bao gồm: <ul style="list-style-type: none"> <li>o Thêm dịch vụ mới: Cư dân chọn dịch vụ muốn thêm, cung cấp thông tin cần thiết. Hệ thống tạo Phụ lục Dịch vụ mới với trạng thái Chờ duyệt và tạo yêu cầu thay đổi.</li> <li>o Hủy dịch vụ: Cư dân chọn dịch vụ muốn hủy, xác nhận. Hệ thống chuyển trạng thái Phụ lục Dịch vụ thành Chờ duyệt, và tạo yêu cầu thay đổi.</li> <li>o Cập nhật thông tin dịch vụ: Cư dân yêu cầu cập nhật thông tin liên quan đến dịch vụ đã đăng ký (ví dụ: thay đổi biển số xe, thay đổi gói dịch vụ).</li> </ul> </li> </ul>
<p><b>Non-Functional Requirements (Yêu cầu phi chức năng nếu có)</b></p>	<p>Quá trình chỉnh sửa hợp đồng và dịch vụ phải thực hiện trong thời gian không quá 5 giây.</p>

**Nhóm 8: Tra cứu hợp đồng**

# Báo cáo Thực tập Tốt nghiệp Đại học

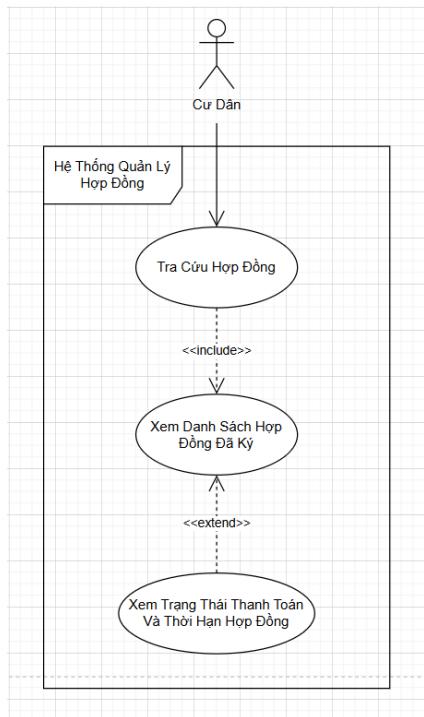


Figure 17: Tra cứu hợp đồng

Table 27: Đặc tả Xem danh sách hợp đồng đã ký

Thành phần	Nội dung
Tên Use Case	Xem danh sách hợp đồng đã ký
Tác nhân chính	Cư dân
Mục tiêu	Cho phép cư dân xem danh sách các hợp đồng mà họ đã ký với Ban Quản Lý
Tiền điều kiện	<ul style="list-style-type: none"> <li>Cư dân đã đăng nhập thành công vào hệ thống.</li> <li>Hệ thống đã lưu trữ các hợp đồng của cư dân.</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>Cư dân truy cập chức năng "Tra cứu hợp đồng" từ giao diện chính.</li> <li>Hệ thống truy vấn dữ liệu và hiển thị danh sách các hợp đồng cư dân đã ký, với thông tin cơ bản: mã hợp đồng, loại hợp đồng, ngày ký, căn hộ, trạng thái.</li> <li>Cư dân có thể chọn một hợp đồng để xem chi tiết.</li> </ol>
Luồng thay thế	<p>AF1 – Không có hợp đồng:</p> <ul style="list-style-type: none"> <li>Nếu cư dân chưa ký hợp đồng nào, hệ thống hiển thị thông báo: "Bạn chưa có hợp đồng nào đã ký."</li> </ul>
Hậu điều kiện	Cư dân đã xem được danh sách các hợp đồng của mình.

Table 28: Đặc tả Xem trạng thái thanh toán và thời hạn hợp đồng

Thành phần	Nội dung
Tên Use Case	Xem chi tiết hợp đồng
Tác nhân chính	Cư dân
Mục tiêu	Cho phép cư dân xem chi tiết một hợp đồng, bao gồm thời hạn, trạng thái hợp đồng, danh sách dịch vụ và tình trạng thanh toán của các dịch vụ đi kèm
Tiền điều kiện	<ul style="list-style-type: none"> <li>– Cư dân đã đăng nhập vào hệ thống.</li> <li>– Cư dân đã chọn một hợp đồng từ danh sách hợp đồng đã ký.</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>1. Cư dân nhấn nút “Xem chi tiết” ở một dòng hợp đồng trong danh sách.</li> <li>2. Hệ thống hiển thị chi tiết hợp đồng, gồm: <ul style="list-style-type: none"> <li>– Thông tin chung: mã hợp đồng, loại hợp đồng, ngày ký, căn hộ, trạng thái.</li> <li>– Dịch vụ đi kèm: tên dịch vụ, đơn giá, chu kỳ thanh toán.</li> <li>– Thời hạn: ngày bắt đầu – ngày kết thúc của từng dịch vụ.</li> <li>– Thanh toán: tổng số hóa đơn, tổng tiền, số tiền đã thanh toán, số tiền còn lại.</li> </ul> </li> </ol>
Luồng thay thế	<p>AF1 – Lỗi kết nối:</p> <ul style="list-style-type: none"> <li>– Nếu hệ thống không truy xuất được dữ liệu, hiển thị thông báo: “Không thể truy cập thông tin hợp đồng. Vui lòng thử lại sau.”</li> </ul>
Hậu điều kiện	Cư dân xem được đầy đủ thông tin chi tiết của hợp đồng, bao gồm thông tin chung, dịch vụ đi kèm, thời hạn và tình trạng thanh toán.

Table 29 Đặc tả Xem trạng thái thanh toán và thời hạn hợp đồng

Thành phần	Nội dung
Tên Use Case	Xem chi tiết hợp đồng
Tác nhân chính	Cư dân

Báo cáo Thực tập Tốt nghiệp Đại học

Mục tiêu	Cho phép cư dân xem chi tiết một hợp đồng, bao gồm thời hạn, trạng thái hợp đồng, danh sách dịch vụ và tình trạng thanh toán của các dịch vụ đi kèm
Tiền điều kiện	<ul style="list-style-type: none"> <li>– Cư dân đã đăng nhập vào hệ thống.</li> <li>– Cư dân đã chọn một hợp đồng từ danh sách hợp đồng đã ký.</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>1. Cư dân nhấn nút “Xem chi tiết” ở một dòng hợp đồng trong danh sách.</li> <li>2. Hệ thống hiển thị chi tiết hợp đồng, gồm: <ul style="list-style-type: none"> <li>– Thông tin chung: mã hợp đồng, loại hợp đồng, ngày ký, căn hộ, trạng thái.</li> <li>– Dịch vụ đi kèm: tên dịch vụ, đơn giá, chu kỳ thanh toán.</li> <li>– Thời hạn: ngày bắt đầu – ngày kết thúc của từng dịch vụ.</li> <li>– Thanh toán: tổng số hóa đơn, tổng tiền, số tiền đã thanh toán, số tiền còn lại.</li> </ul> </li> </ol>
Luồng thay thế	<p>AF1 – Lỗi kết nối:</p> <ul style="list-style-type: none"> <li>– Nếu hệ thống không truy xuất được dữ liệu, hiển thị thông báo: “Không thể truy cập thông tin hợp đồng. Vui lòng thử lại sau.”</li> </ul>
Hậu điều kiện	Cư dân xem được đầy đủ thông tin chi tiết của hợp đồng, bao gồm thông tin chung, dịch vụ đi kèm, thời hạn và tình trạng thanh toán.

**Nhóm 9: Thanh toán**

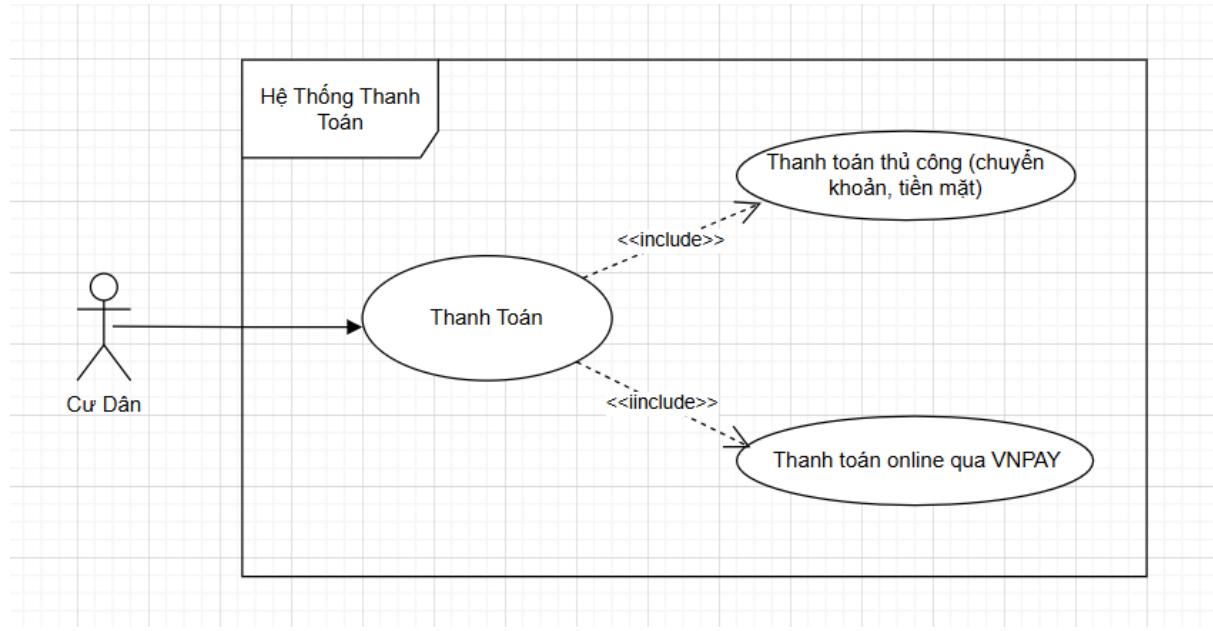


Figure 18: UC Thanh toán

Table 30: Đặc tả Thanh toán thủ công (chuyển khoản, tiền mặt)

Thành phần	Nội dung
Tên Use Case	Thanh toán thủ công (chuyển khoản, tiền mặt)
Tác nhân chính	Cư dân
Mục tiêu	Cho phép cư dân thanh toán hóa đơn bằng hình thức chuyển khoản ngân hàng hoặc tiền mặt tại văn phòng
Tiền điều kiện	<ul style="list-style-type: none"> <li>Cư dân đã đăng nhập hệ thống.</li> <li>Hóa đơn chưa thanh toán đã được tạo.</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>Cư dân truy cập vào mục "Hóa đơn của tôi".</li> <li>Hệ thống hiển thị danh sách hóa đơn chưa thanh toán.</li> <li>Cư dân chọn hóa đơn muốn thanh toán.</li> <li>Hệ thống hiển thị chi tiết hóa đơn.</li> <li>Cư dân chọn hình thức thanh toán thủ công: <ul style="list-style-type: none"> <li>Nếu chọn chuyển khoản: hệ thống hiển thị hướng dẫn chuyển khoản (thông tin tài khoản ngân hàng, nội dung chuyển khoản).</li> <li>Nếu chọn tiền mặt: hệ thống hiển thị địa điểm và thời gian thanh toán tại văn phòng.</li> </ul> </li> <li>Cư dân tiến hành thanh toán.</li> </ol>

	<p>7. Hệ thống tạo giao dịch "Chờ xác nhận" cho hóa đơn tương ứng.</p> <p>8. Ban quản lý kiểm tra thông tin và thực hiện xác nhận:</p> <ul style="list-style-type: none"> <li>• Nếu hợp lệ: cập nhật hóa đơn sang trạng thái "Đã thanh toán", giao dịch thành "Thành công", và gửi thông báo xác nhận cho cư dân.</li> <li>• Nếu không hợp lệ: cập nhật giao dịch thành "Thất bại", giữ trạng thái hóa đơn là "Chưa thanh toán", và gửi lý do từ chối cho cư dân.</li> </ul>
Luồng thay thế	<p>AF1 – Cư dân thoát khỏi giao diện thanh toán: Hệ thống không tạo giao dịch, luồng kết thúc.</p> <p>AF2 – BQL chưa xác nhận: Hệ thống giữ hóa đơn ở trạng thái "Chưa thanh toán", giao dịch ở trạng thái "Chờ xác nhận".</p>
Hậu điều kiện	<ul style="list-style-type: none"> <li>- Hóa đơn được đánh dấu "Đã thanh toán" khi có xác nhận từ Ban quản lý.</li> <li>- Giao dịch được ghi nhận thành công/thất bại tùy theo kết quả xác minh.</li> </ul>

Table 31: Đặc tả Thanh toán online qua VNPay

Thành phần	Nội dung
Tên Use Case	Thanh toán online qua VNPay
Tác nhân chính	Cư dân
Mục tiêu	Cho phép cư dân thực hiện thanh toán hóa đơn thông qua cổng thanh toán VNPay một cách nhanh chóng và tiện lợi
Tiền điều kiện	<p>Cư dân đã đăng nhập hệ thống</p> <ul style="list-style-type: none"> <li>• Có ít nhất một hóa đơn chưa thanh toán</li> <li>• Cư dân có tài khoản ngân hàng liên kết VNPay</li> <li>• Kết nối mạng và cổng VNPay hoạt động ổn định</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>1. Cư dân truy cập mục "Hóa đơn của tôi"</li> <li>2. Hệ thống hiển thị danh sách hóa đơn chưa thanh toán</li> <li>3. Cư dân chọn hóa đơn cần thanh toán</li> <li>4. Hệ thống hiển thị chi tiết hóa đơn</li> <li>5. Cư dân chọn "Thanh toán online qua VNPay"</li> <li>6. Hệ thống chuyển hướng đến trang thanh toán của VNPay</li> </ol>

	<p>7. Cư dân thực hiện nhập thông tin thẻ, xác thực OTP  8. Hệ thống nhận kết quả từ VNPAY:</p> <ul style="list-style-type: none"> <li>– Nếu giao dịch thành công: cập nhật hóa đơn → “Đã thanh toán”</li> <li>– Nếu thất bại: giữ nguyên trạng thái, báo lỗi</li> </ul>
Luồng thay thế	<p>AF1 – Giao dịch thất bại từ VNPAY  → Hệ thống hiển thị thông báo lỗi, hóa đơn vẫn “Chưa thanh toán”</p> <p>AF2 – Mất kết nối mạng trong quá trình thanh toán  → Hệ thống hiển thị lỗi kỹ thuật, hướng dẫn thử lại</p>
Hậu điều kiện	Hóa đơn được cập nhật trạng thái “Đã thanh toán” nếu nhận được phản hồi thành công từ VNPAY

#### Nhóm 10: Nhận thông báo

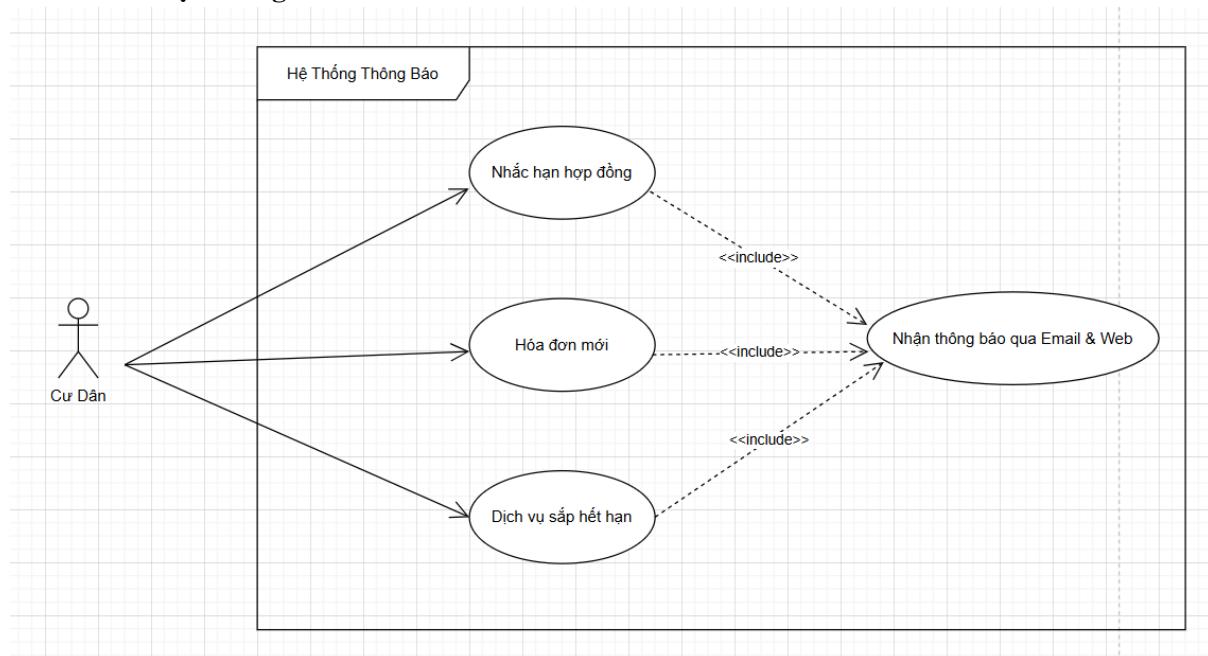


Figure 19: UC Nhận thông báo

Table 32: Đặc tả Cư dân nhận nhắc nhở thời hạn hợp đồng

Thành phần	Nội dung
Tên Use Case	UC1 - Nhắc nhở thời hạn hợp đồng
Tác nhân chính	Cư dân

Báo cáo Thực tập Tốt nghiệp Đại học

Mục tiêu	Cư dân được nhắc trước khi hợp đồng sắp hết hạn để chủ động gia hạn/tra cứu.
Tiền điều kiện	<ul style="list-style-type: none"> <li>Cư dân có ít nhất một hợp đồng đang hiệu lực.</li> <li>Cư dân đã đăng ký tài khoản và có quyền truy cập cổng web.</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>Đến mốc thời gian nhắc (vd. trước 30/15/7 ngày), hệ thống tạo thông báo “Hợp đồng sắp hết hạn”.</li> <li>Hệ thống phân phối thông báo qua email và hiển thị trong mục Thông báo trên web.</li> <li>Cư dân mở thông báo để xem chi tiết (ngày hết hạn, gợi ý thao tác).</li> <li>Cư dân có thể bấm đến trang Hợp đồng để xem/gia hạn theo quy trình riêng.</li> </ol>
Luồng thay thế	<p>AF1 – Không có hợp đồng sắp hết hạn: hệ thống không phát thông báo trong kỳ.</p> <p>AF2 – Cư dân tắt email/không có email hợp lệ: hệ thống chỉ hiển thị trên web.</p>
Hậu điều kiện	Thông báo xuất hiện trong danh sách thông báo; nếu email hợp lệ, cư dân nhận thêm email tương ứng.

Table 33: Đặc tả Cư dân nhận thông báo khi có hóa đơn mới

Thành phần	Nội dung
Tên Use Case	UC2 - Thông báo khi có hóa đơn mới
Tác nhân chính	Cư dân
Mục tiêu	Cư dân được thông báo ngay khi hệ thống phát hành hóa đơn mới để kịp thời thanh toán.
Tiền điều kiện	<ul style="list-style-type: none"> <li>Cư dân có hợp đồng/dịch vụ đang hoạt động.</li> <li>Hệ thống đã lập hóa đơn cho kỳ thanh toán mới.</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>Khi hệ thống lập hóa đơn mới, tự động tạo thông báo với các thông tin: kỳ tính, số tiền, hạn thanh toán.</li> <li>Hệ thống gửi thông báo qua email và hiển thị tại mục Thông báo trên web.</li> </ol>

	3) Cư dân mở thông báo để xem chi tiết và có thể bấm liên kết để truy cập trang hóa đơn.
Luồng thay thế	AF1 – Có nhiều hóa đơn cùng lúc: hệ thống gộp thành thông báo tóm tắt với liên kết xem danh sách chi tiết. AF2 – Email không hợp lệ: chỉ hiển thị trên web.
Hậu điều kiện	Cư dân biết được hóa đơn mới và có đường dẫn nhanh để thanh toán.

Table 34: Đặc tả Cư dân nhận thông báo khi sắp hết hạn dịch vụ

Thành phần	Nội dung
Tên Use Case	UC3 - Thông báo khi sắp hết hạn dịch vụ
Tác nhân chính	Cư dân
Mục tiêu	Nhắc nhở cư dân về các dịch vụ sắp hết hạn để kịp gia hạn hoặc xử lý.
Tiền điều kiện	- Cư dân có ít nhất một dịch vụ trong hợp đồng sắp hết hạn.  - Hệ thống đã thiết lập thời điểm gửi nhắc (vd. trước 7 ngày).
Luồng chính	1) Đến thời điểm nhắc, hệ thống xác định các dịch vụ sắp hết hạn.  2) Hệ thống gửi thông báo qua email và hiển thị tại mục Thông báo trên web.  3) Cư dân mở thông báo để xem chi tiết dịch vụ và hướng dẫn gia hạn.
Luồng thay thế	AF1 – Không có dịch vụ sắp hết hạn: không gửi thông báo trong kỳ.  AF2 – Email không hợp lệ: chỉ hiển thị trên web.
Hậu điều kiện	Cư dân biết được dịch vụ nào sắp hết hạn và có thể thực hiện gia hạn kịp thời.

Table 35: Đặc tả Cư dân nhận và xem thông báo qua email & trên web

Thành phần	Nội dung
Tên Use Case	UC4 - Nhận thông báo qua email và hiển thị trên giao diện web
Tác nhân chính	Cư dân
Mục tiêu	Đảm bảo cư dân nhận được thông báo qua hai kênh: email và mục Thông báo trên web; có thể xem, mở chi tiết, đánh dấu đã xem.
Tiền điều kiện	<ul style="list-style-type: none"> <li>- Cư dân đã đăng nhập để xem thông báo trên web.</li> <li>- Email của cư dân tồn tại (đối với kênh email).</li> </ul>
Luồng chính	<ol style="list-style-type: none"> <li>1) Khi có thông báo (từ UC1/UC2/UC3), hệ thống đồng thời:           <ol style="list-style-type: none"> <li>(a) gửi email đến địa chỉ của cư dân;</li> <li>(b) hiển thị thẻ thông báo mới trên web (icon/badge).</li> </ol> </li> <li>2) Cư dân mở mục Thông báo để xem danh sách, gồm tiêu đề, thời gian, trạng thái (mới/đã xem).</li> <li>3) Cư dân chọn một thông báo để xem chi tiết nội dung và liên kết hành động (xem hợp đồng/xem hóa đơn).</li> <li>4) Hệ thống đánh dấu thông báo là đã xem và cập nhật bộ đếm.</li> </ol>
Luồng thay thế	<p>AF1 – Không muôn nhận email: cư dân tắt tùy chọn email (nếu có), hệ thống vẫn hiển thị trên web.</p> <p>AF2 – Đánh dấu tất cả đã xem: cư dân chọn “Đánh dấu tất cả”, hệ thống cập nhật trạng thái các thông báo.</p>
Hậu điều kiện	Thông báo đã được cư dân đọc/đánh dấu; kênh email và web hoạt động nhất quán về nội dung.

### 3 Sơ đồ hoạt động (Activity diagram)

#### Nhóm 0: Đặc biệt:

UC0.1 – Ban quản lý đăng nhập Hệ thống

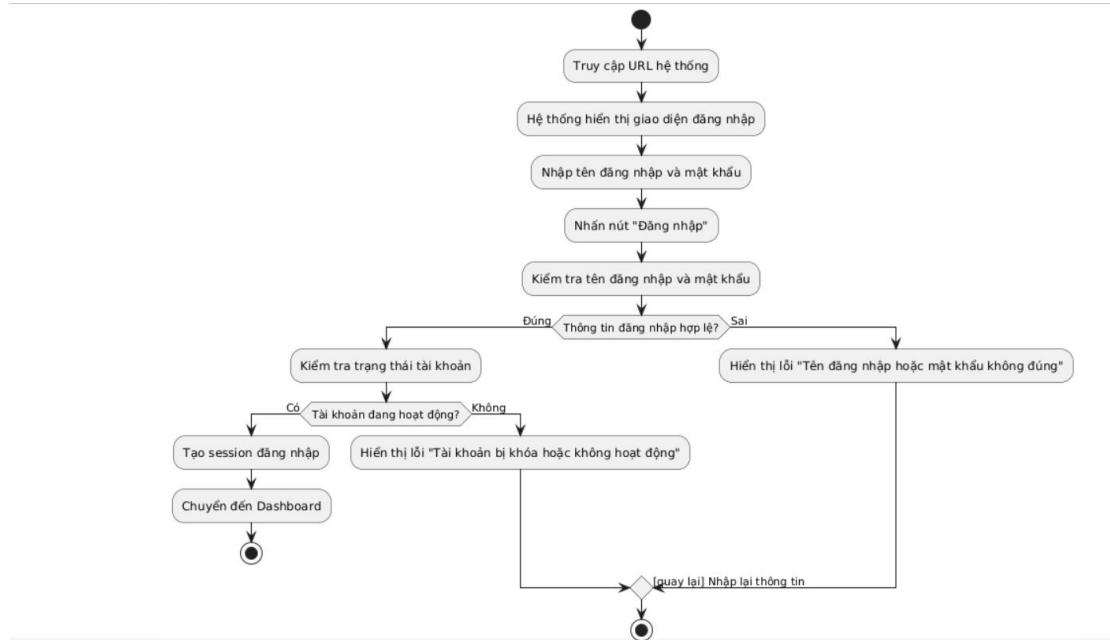


Figure 20: AD Ban quản lý đăng nhập Hệ thống

### Nhóm 1: Quản lý tài khoản cư dân

#### UC1.2 – Duyệt tài khoản

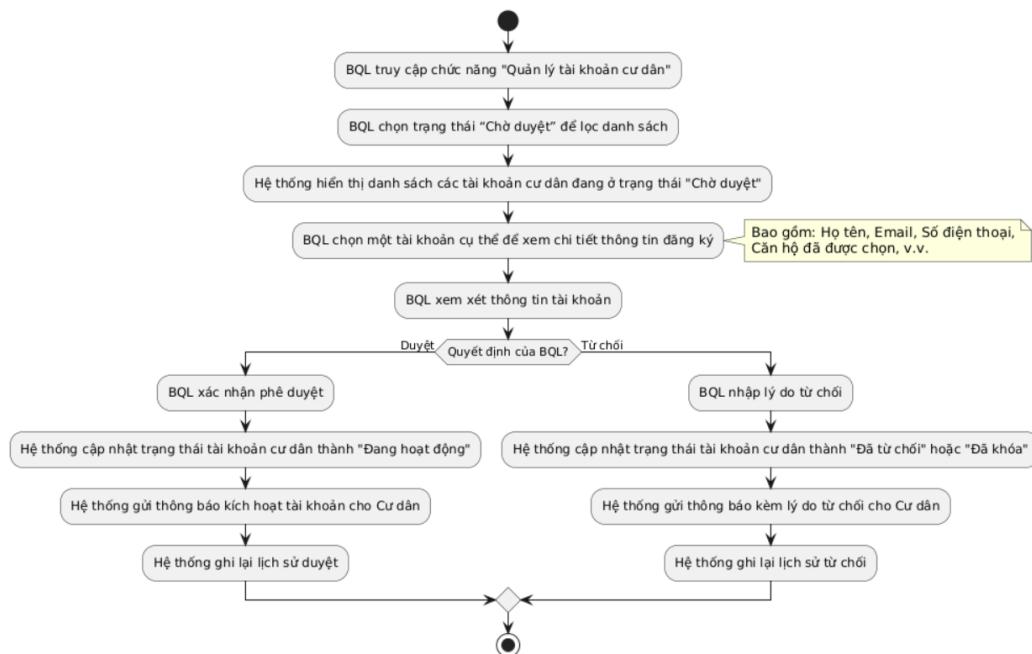


Figure 21: Duyệt tài khoản

#### UC1.3 – Khóa / Mở khóa

# Báo cáo Thực tập Tốt nghiệp Đại học

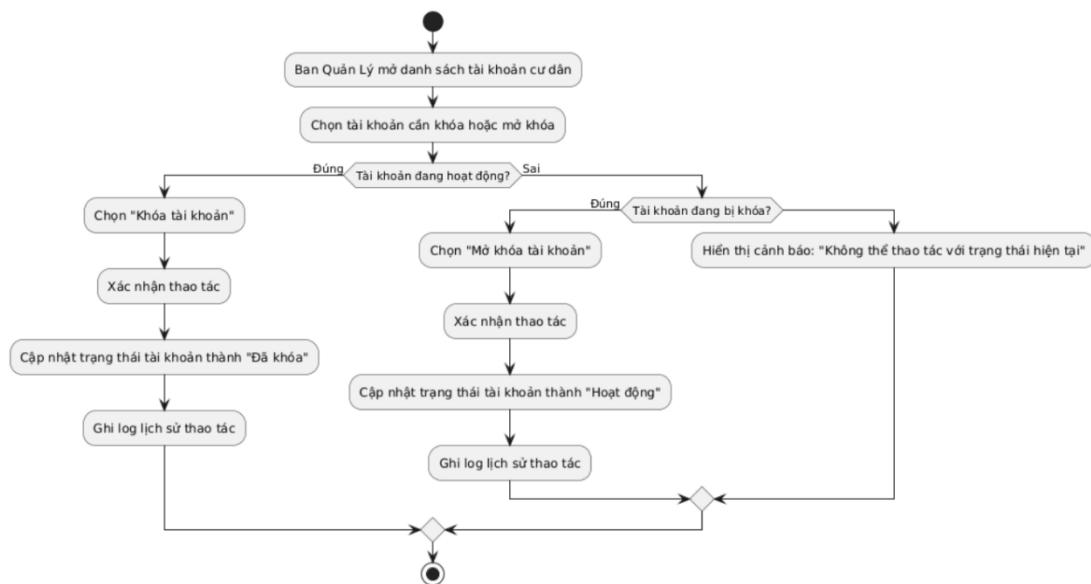


Figure 22: AD Khóa / Mở khóa

## Nhóm 2. Quản lý dịch vụ

UC2.1 – Khai báo dịch vụ mới

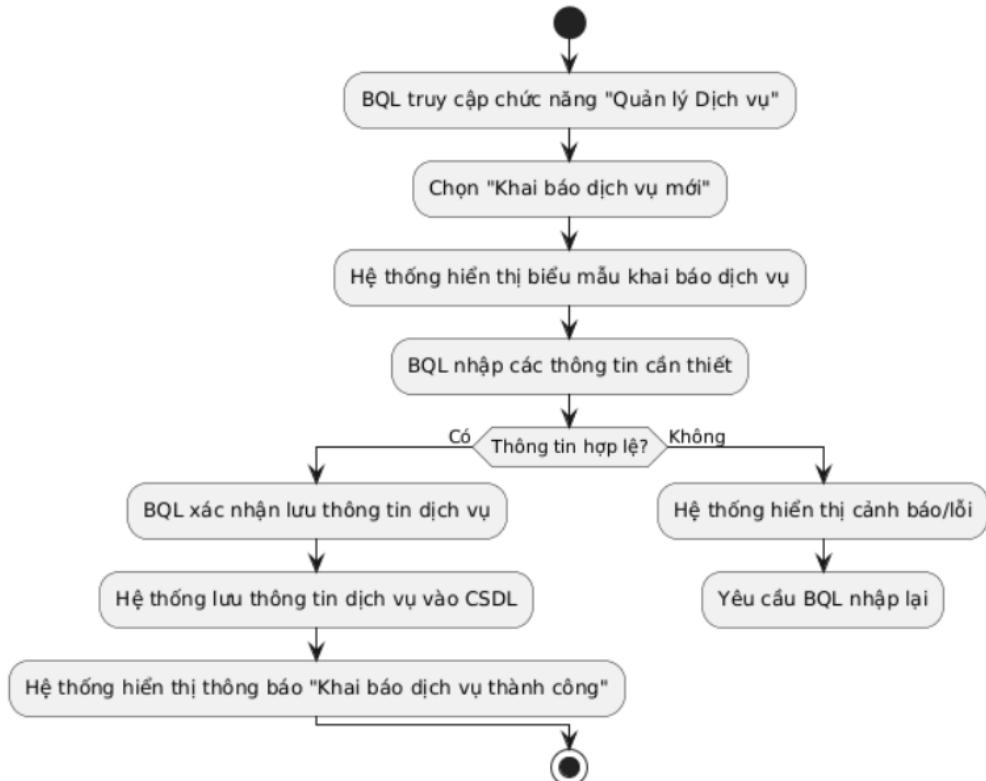


Figure 23: AD Khai báo dịch vụ mới

Báo cáo Thực tập Tốt nghiệp Đại học  
 UC2.2 – Quản lý Cấu hình Chi tiết Dịch vụ



Figure 24: AD Quản lý Cấu hình Chi tiết Dịch vụ

UC2.3 – Chính sửa Thông tin & Trạng thái Dịch vụ đã khai báo

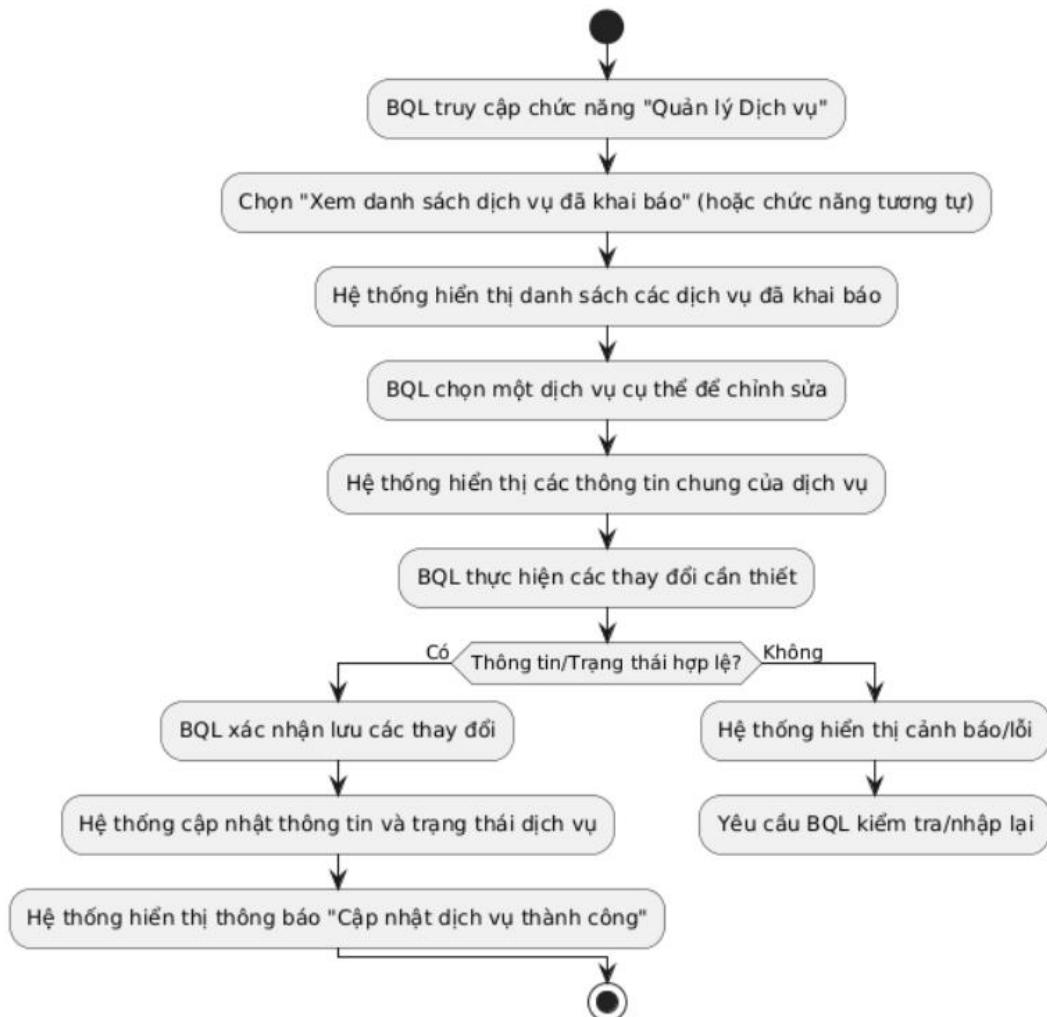


Figure 25: AD Chỉnh sửa Thông tin & Trạng thái Dịch vụ đã khai báo

### Nhóm 3. Quản lý mẫu hợp đồng

UC3.1 – Tạo mẫu hợp đồng dịch vụ

# Báo cáo Thực tập Tốt nghiệp Đại học

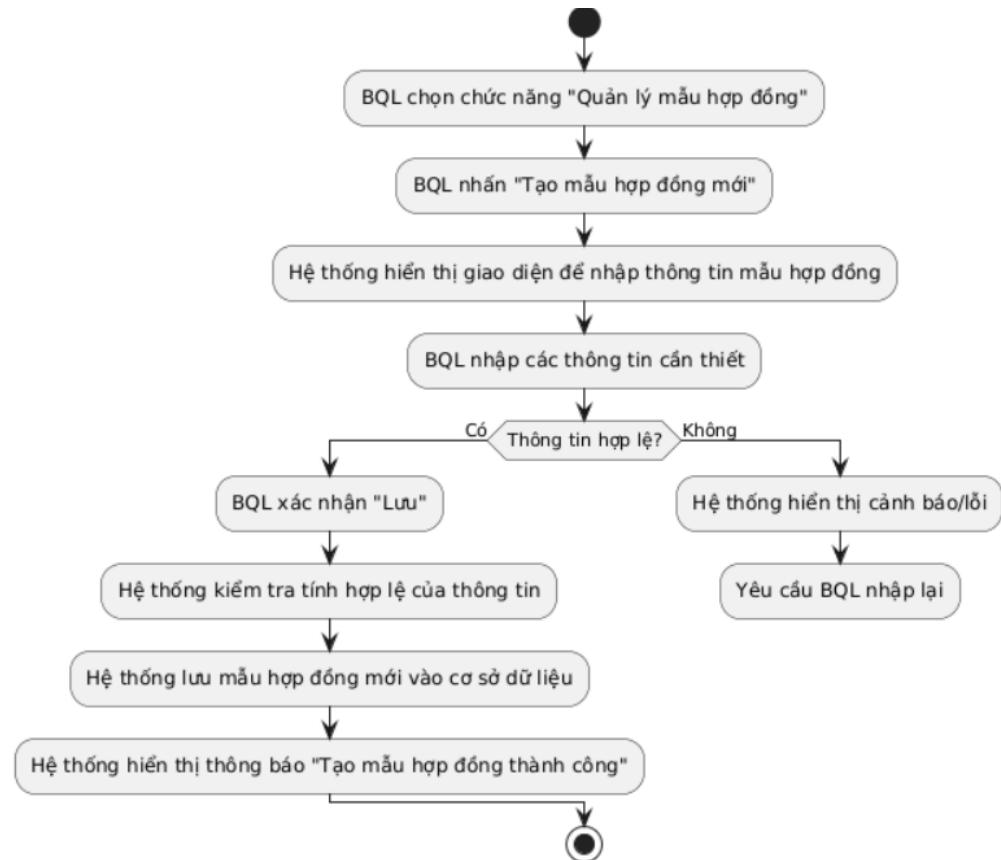


Figure 26: AD Tạo mẫu hợp đồng dịch vụ

## UC3.2 – Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ

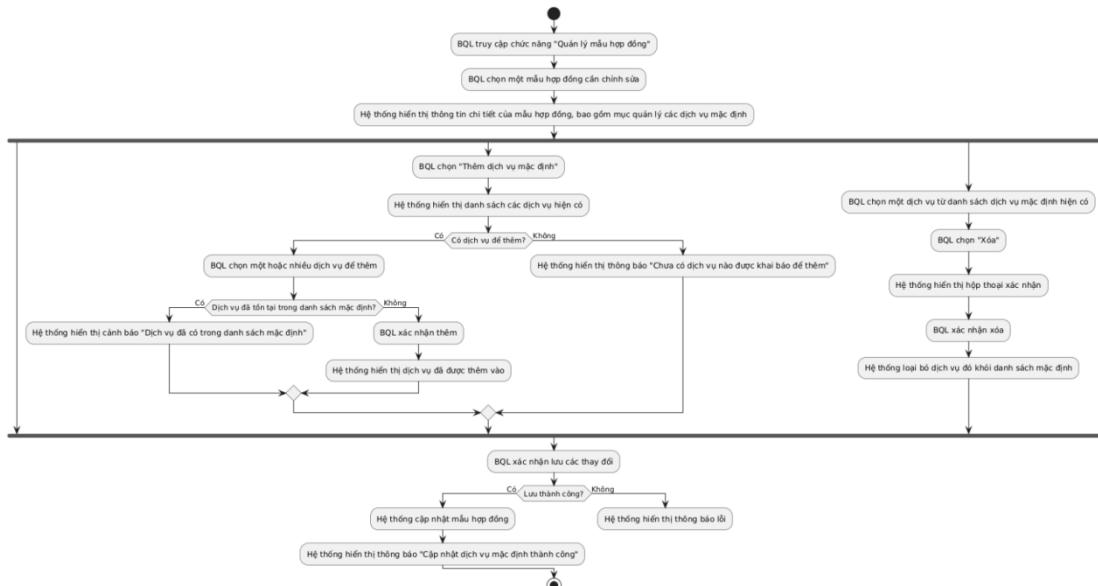


Figure 27: AD Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ

## UC3.4 – Chính sửa mẫu

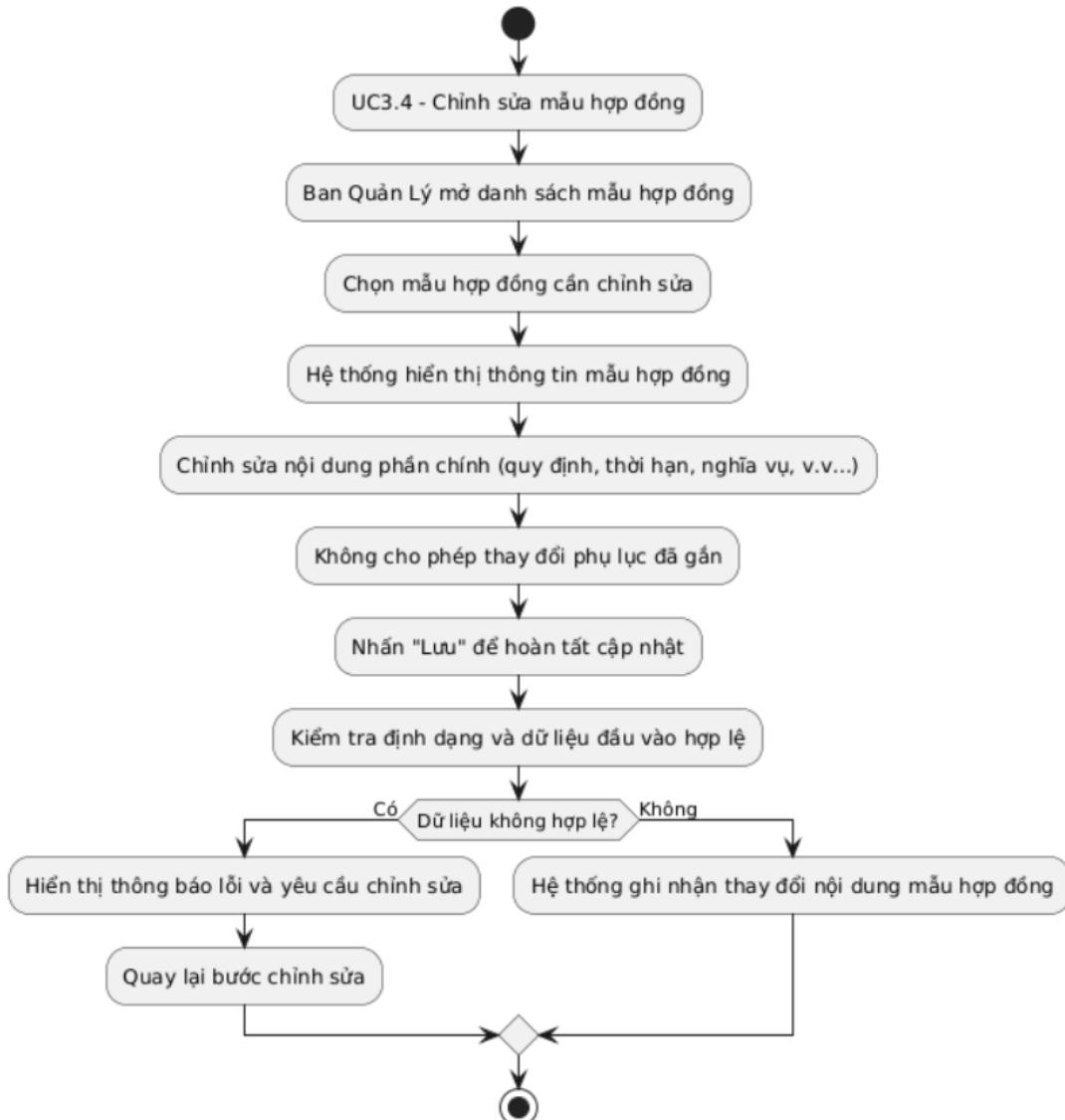


Figure 28: AD Chỉnh sửa mẫu

#### Nhóm 4. Quản lý hợp đồng cư dân

UC4.3 – Tự động chấm dứt do vi phạm

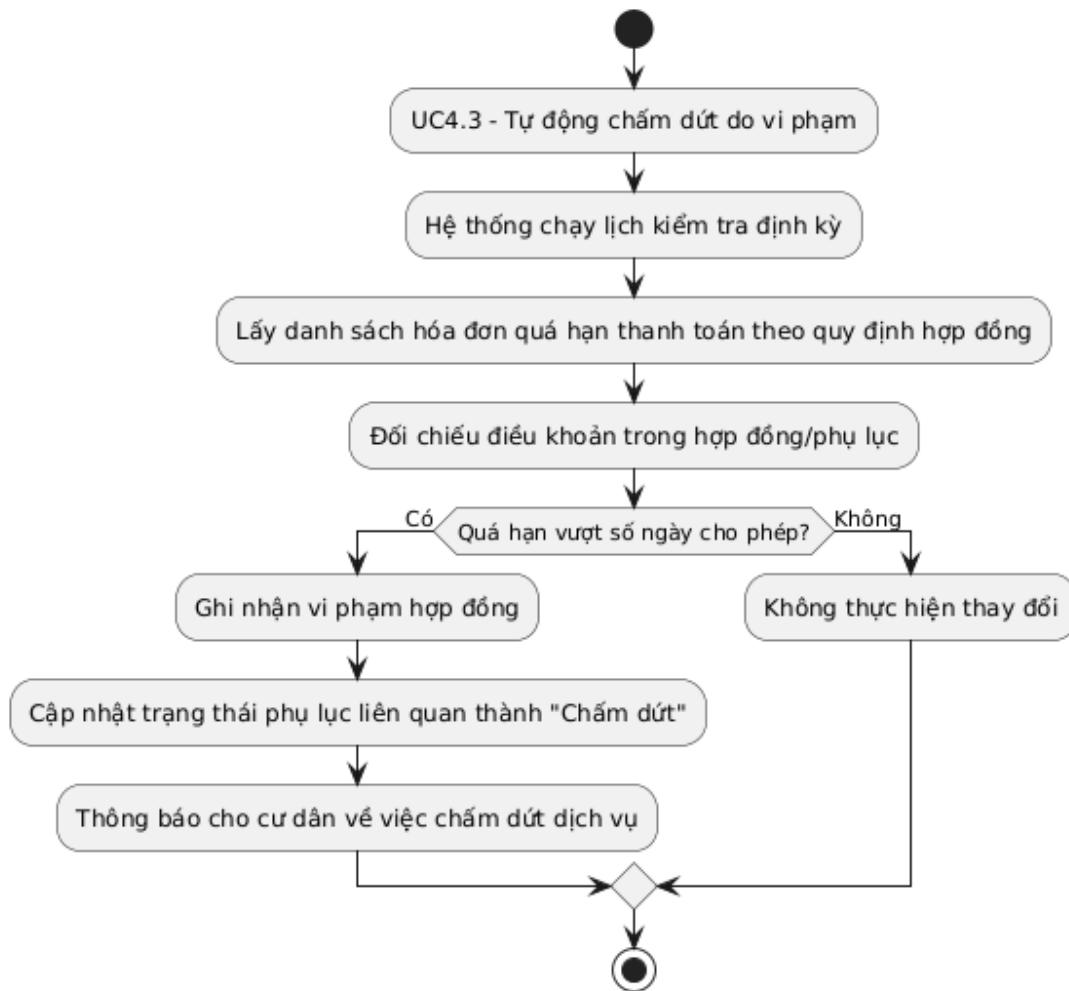


Figure 29 AD Tự động chấm dứt do vi phạm

UC4.4 – BQL chấm dứt do vi phạm nội quy

## Báo cáo Thực tập Tốt nghiệp Đại học

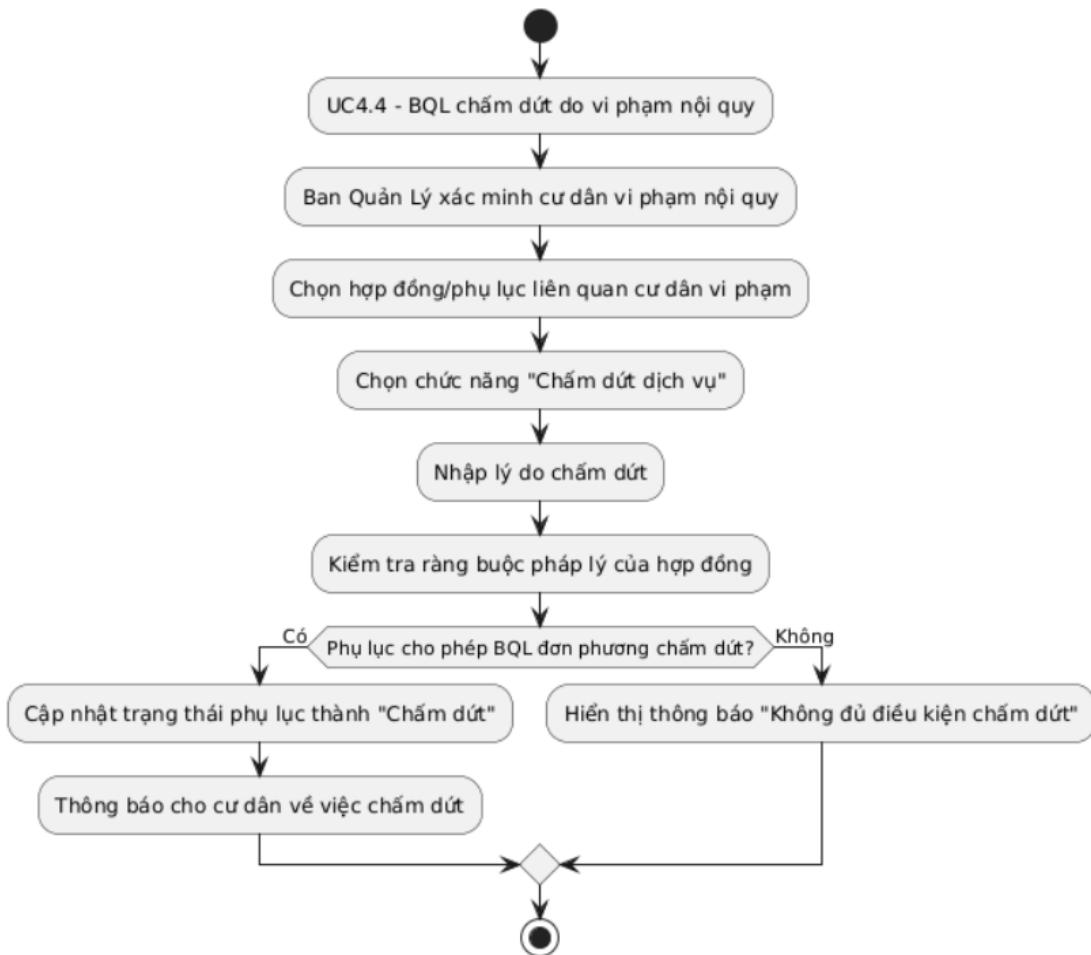


Figure 30 AD BQL chấm dứt do vi phạm nội quy

## UC4.5 – BQL đơn phương chấm dứt một dịch vụ riêng lẻ

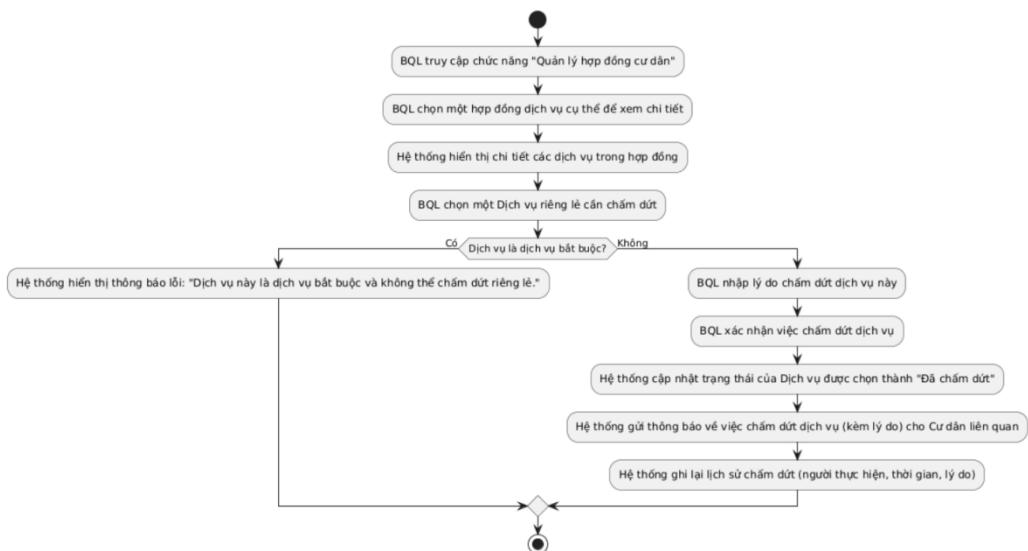


Figure 31 AD BQL đơn phương chấm dứt một dịch vụ riêng lẻ

## UC4.6 – BQL Duyệt Hợp đồng Dịch vụ mới

# Báo cáo Thực tập Tốt nghiệp Đại học

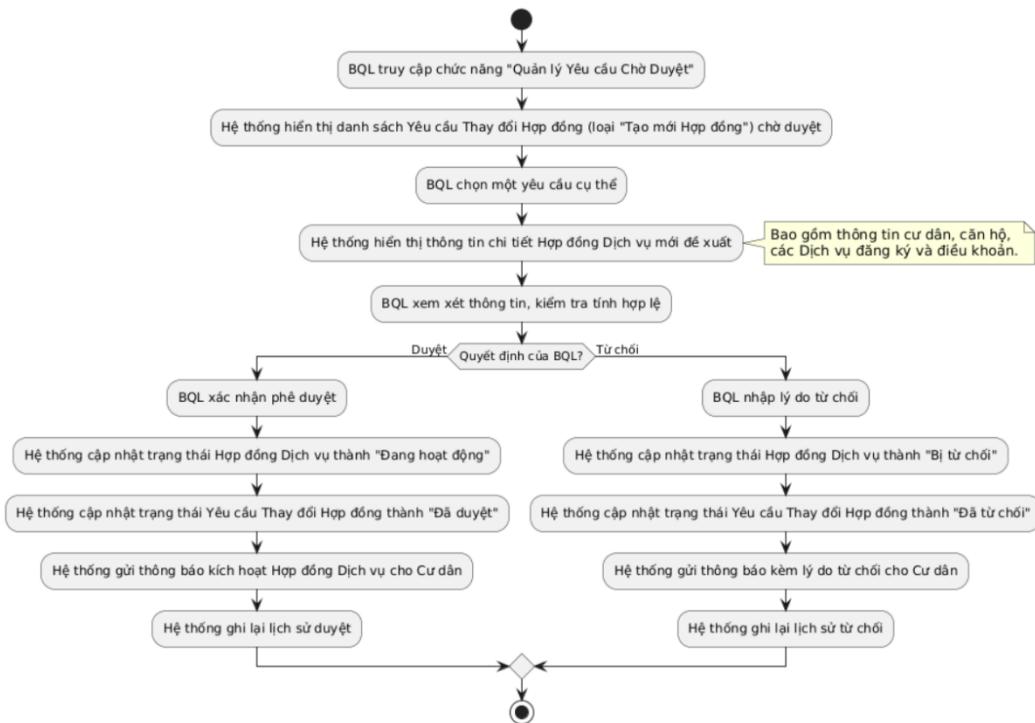


Figure 32 AD BQL Duyệt Hợp đồng Dịch vụ mới

## UC4.7 – BQL Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng

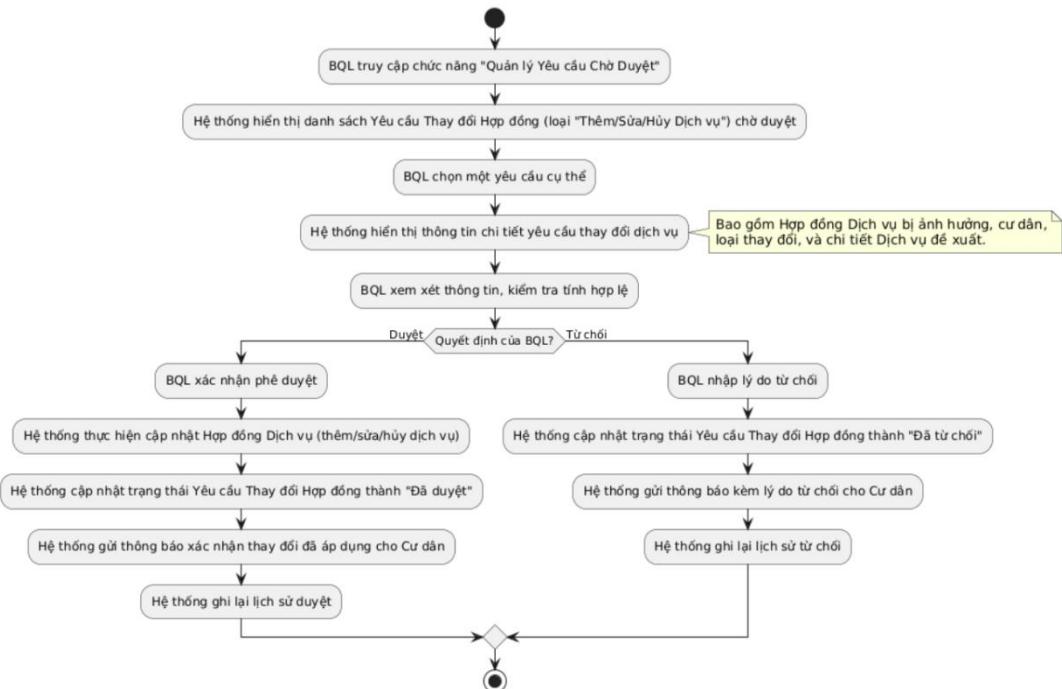


Figure 33 AD BQL Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng

## Nhóm 5. Quản lý hóa đơn và chỉ số điện/nước

# Báo cáo Thực tập Tốt nghiệp Đại học

## UC5.1 – Tự động lập hóa đơn

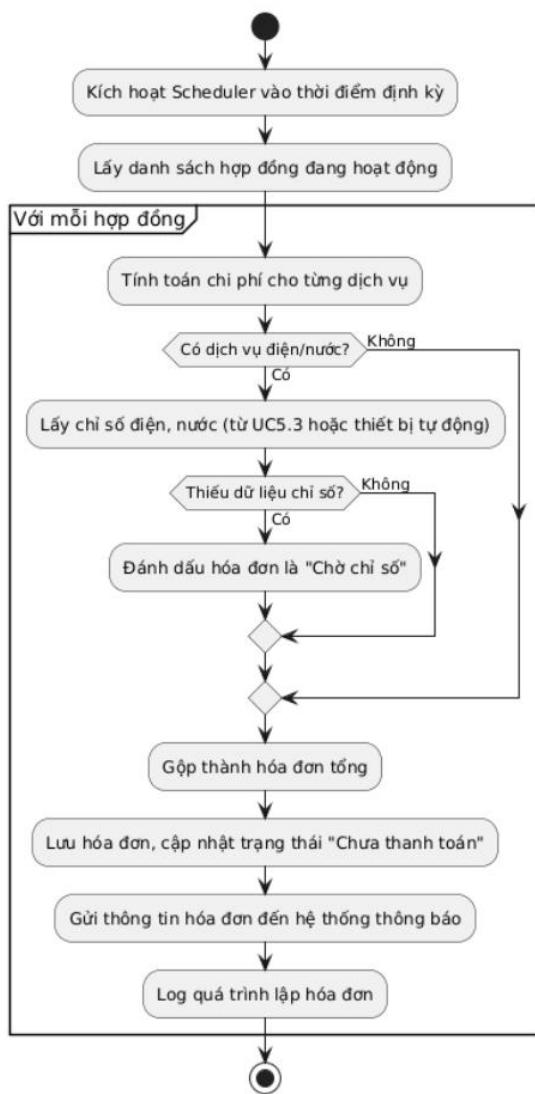


Figure 34 AD Tự động lập hóa đơn

## UC5.2 – Ghi chỉ số điện/nước

# Báo cáo Thực tập Tốt nghiệp Đại học

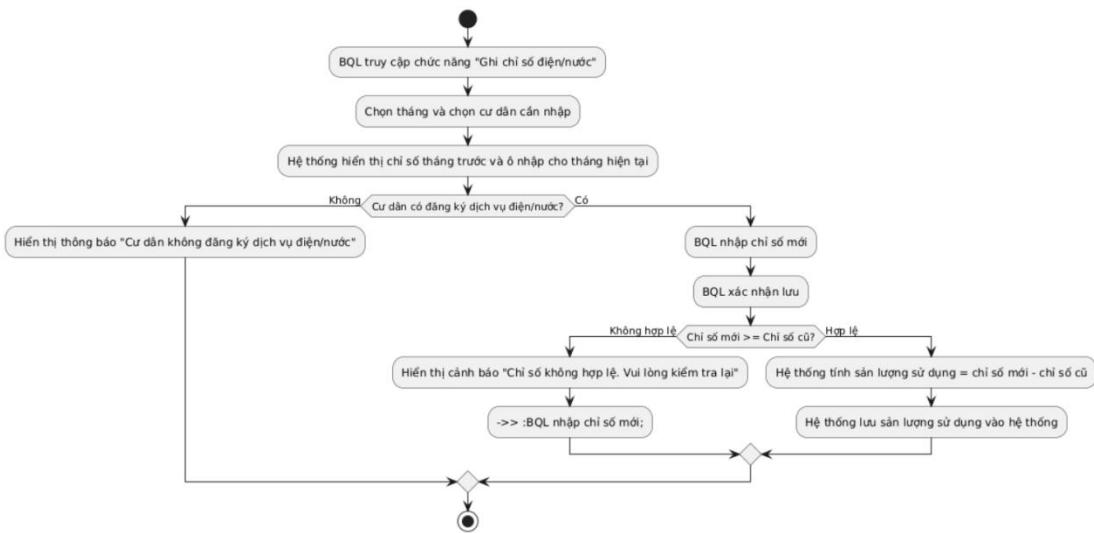


Figure 35 AD Ghi chỉ số điện/nước

## UC5.3 Xác nhận thanh toán

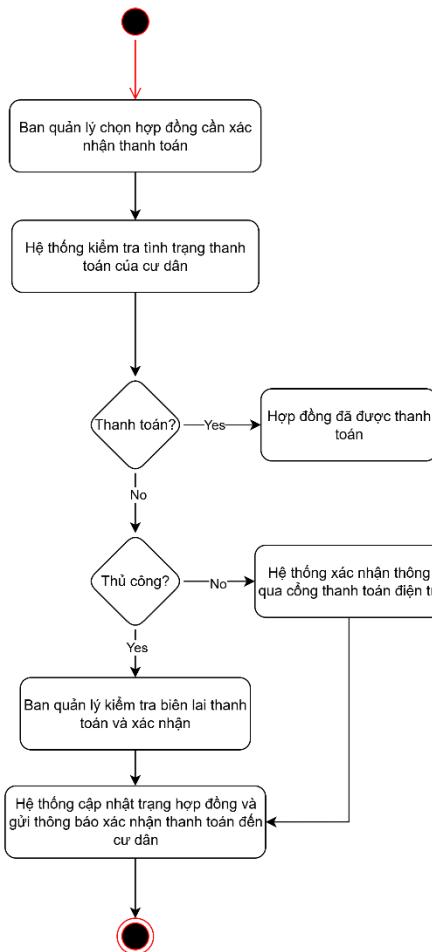


Figure 36 AD Xác nhận thanh toán

## Nhóm 6: Quản lý tài khoản cá nhân (Cư dân)

### UC6.1 Đăng ký tài khoản

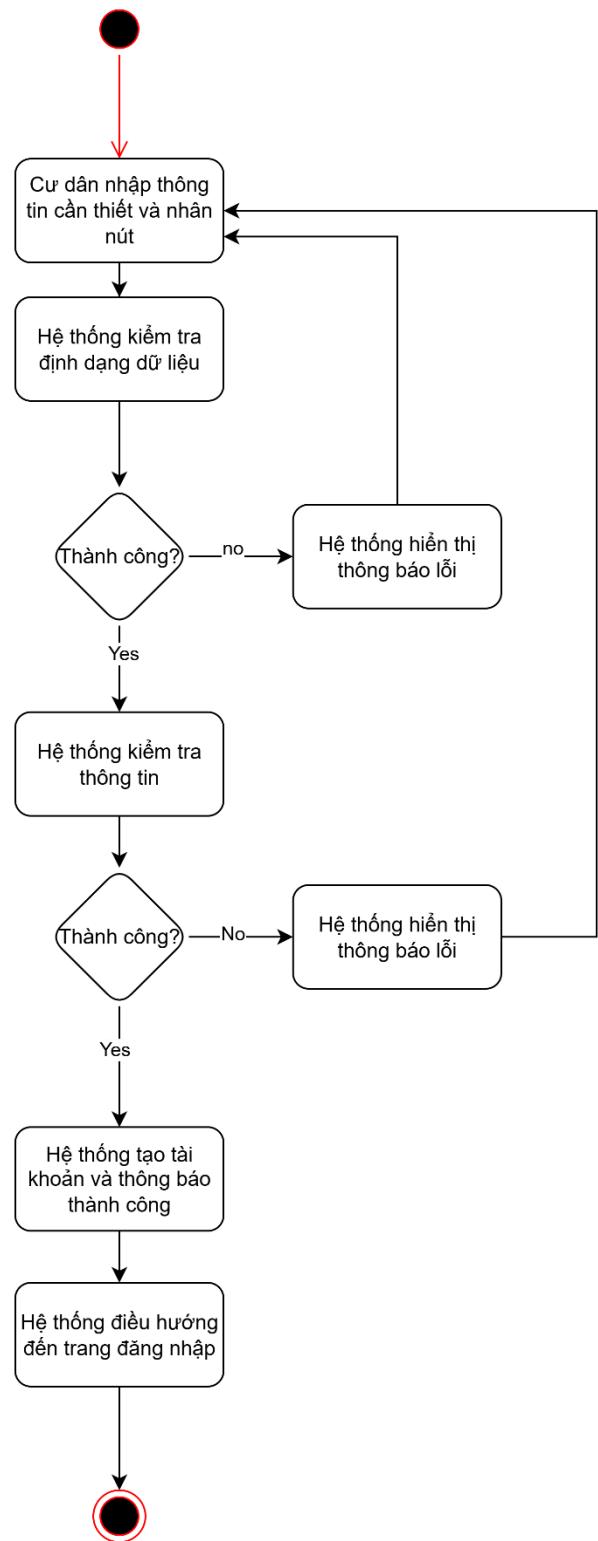


Figure 37 AD Đăng ký tài khoản

### UC6.1 Cập nhật thông tin cá nhân

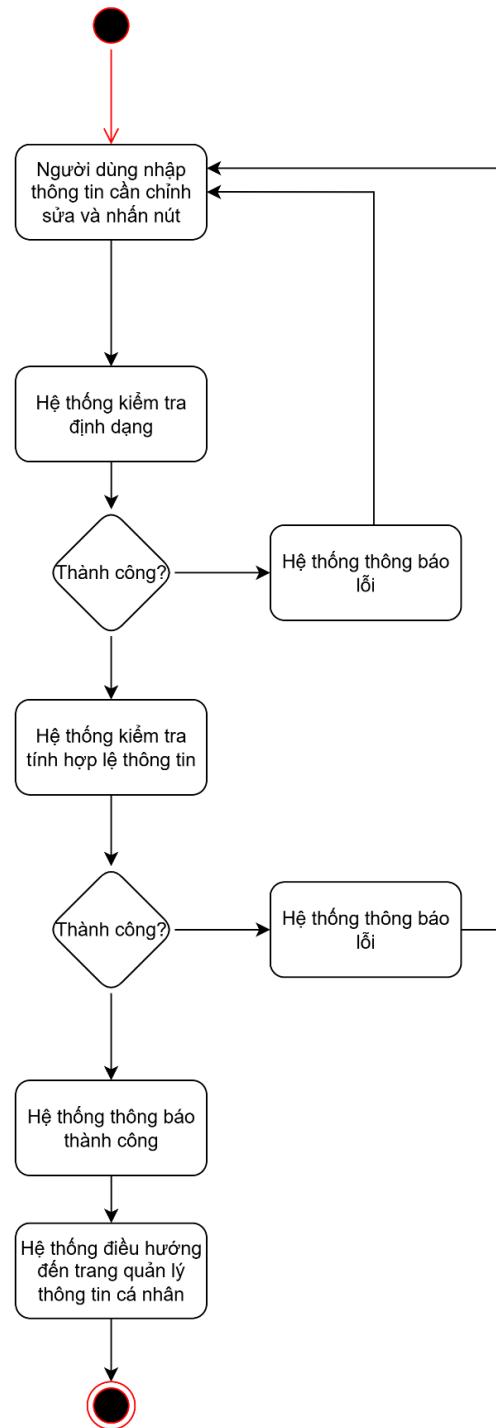


Figure 38 AD Cập nhật thông tin cá nhân

#### Nhóm 7: Đăng ký dịch vụ hợp đồng

##### UC7.1 Lập hợp đồng

# Báo cáo Thực tập Tốt nghiệp Đại học

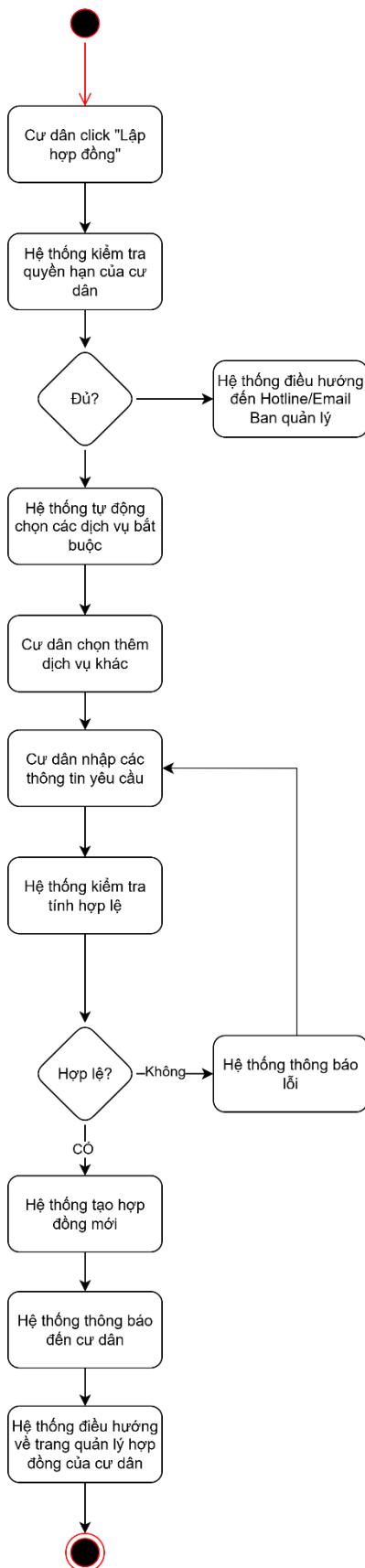


Figure 39 AD Lập hợp đồng

## UC7.2 Xem điều khoản dịch vụ

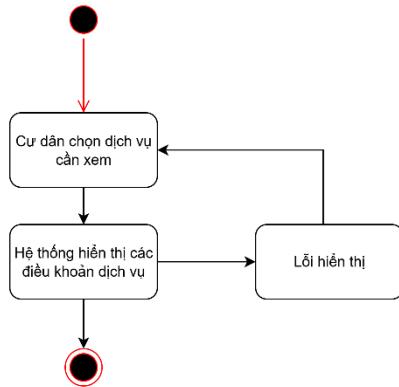


Figure 40 AD Xem điều khoản dịch vụ

### UC7.3 Thay đổi dịch vụ hợp đồng

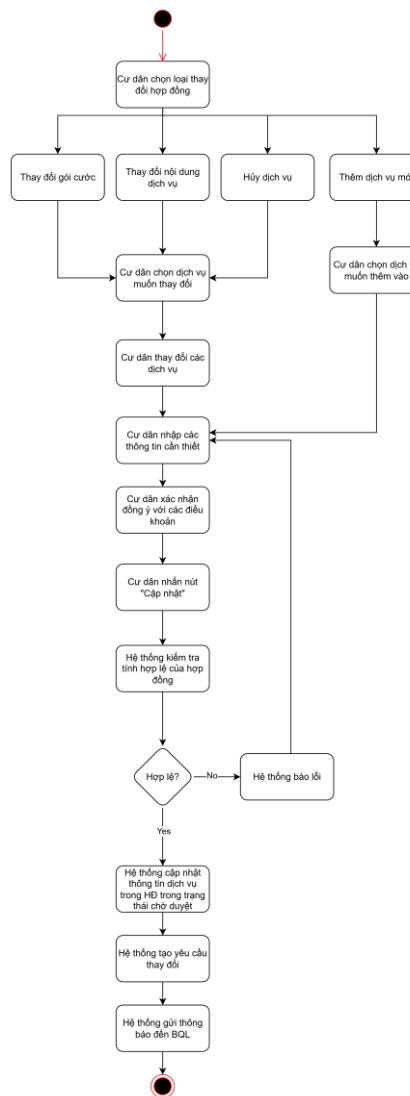


Figure 41 AD Thay đổi dịch vụ hợp đồng

### Nhóm 8: Tra cứu hợp đồng

## Xem danh sách hợp đồng đã ký



Figure 42 AD Xem danh sách hợp đồng đã ký

## Xem trạng thái thanh toán và thời hạn hợp đồng

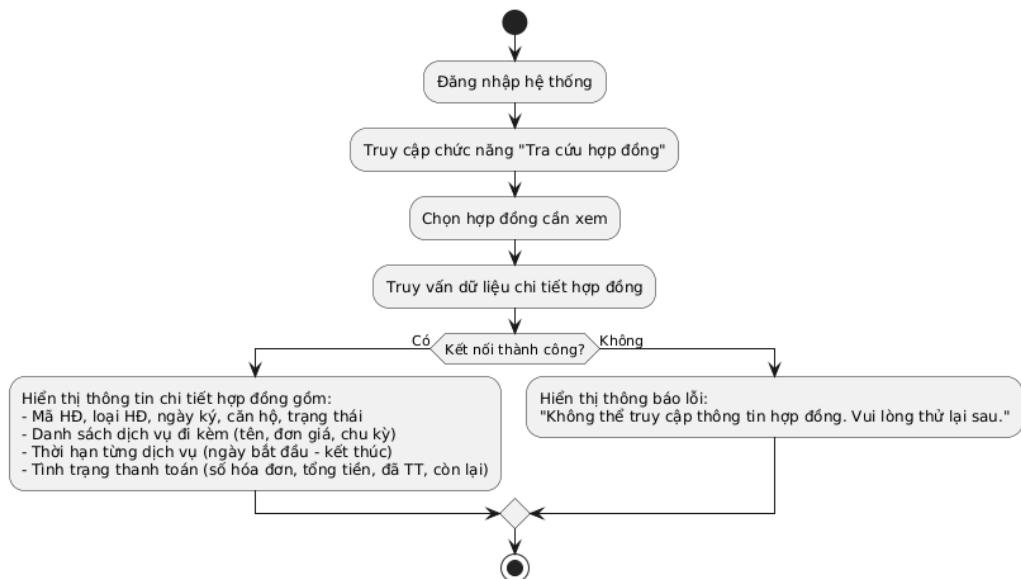


Figure 43 AD Xem trạng thái thanh toán và thời hạn hợp đồng

## Nhóm 9: Thanh toán online

Báo cáo Thực tập Tốt nghiệp Đại học  
**Thanh toán thủ công (chuyển khoản, tiền mặt)**

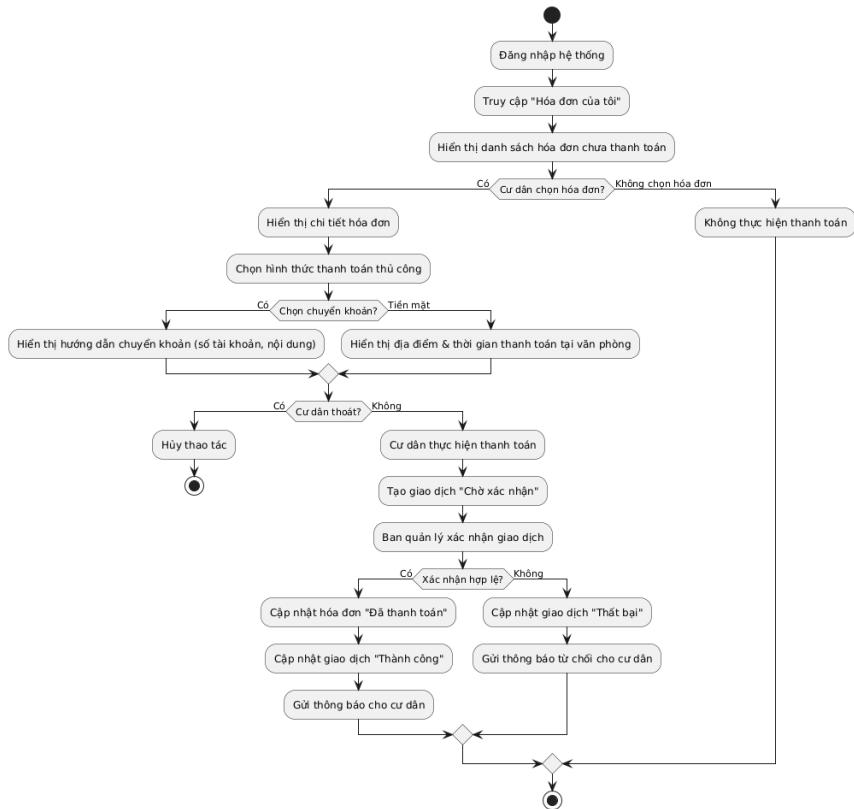


Figure 44 AD Thanh toán thủ công (chuyển khoản, tiền mặt)

**Thanh toán online qua VNPay**

# Báo cáo Thực tập Tốt nghiệp Đại học

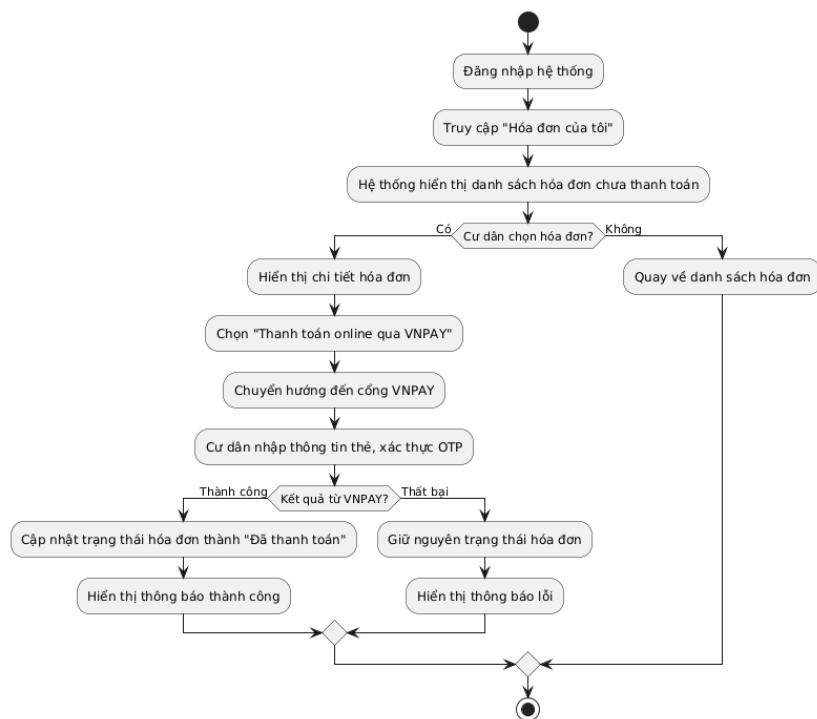


Figure 45 AD Thanh toán online qua VNPAY

## Nhóm 10: Nhận thông báo

Cư dân nhận nhắc nhở thời hạn hợp đồng

# Báo cáo Thực tập Tốt nghiệp Đại học

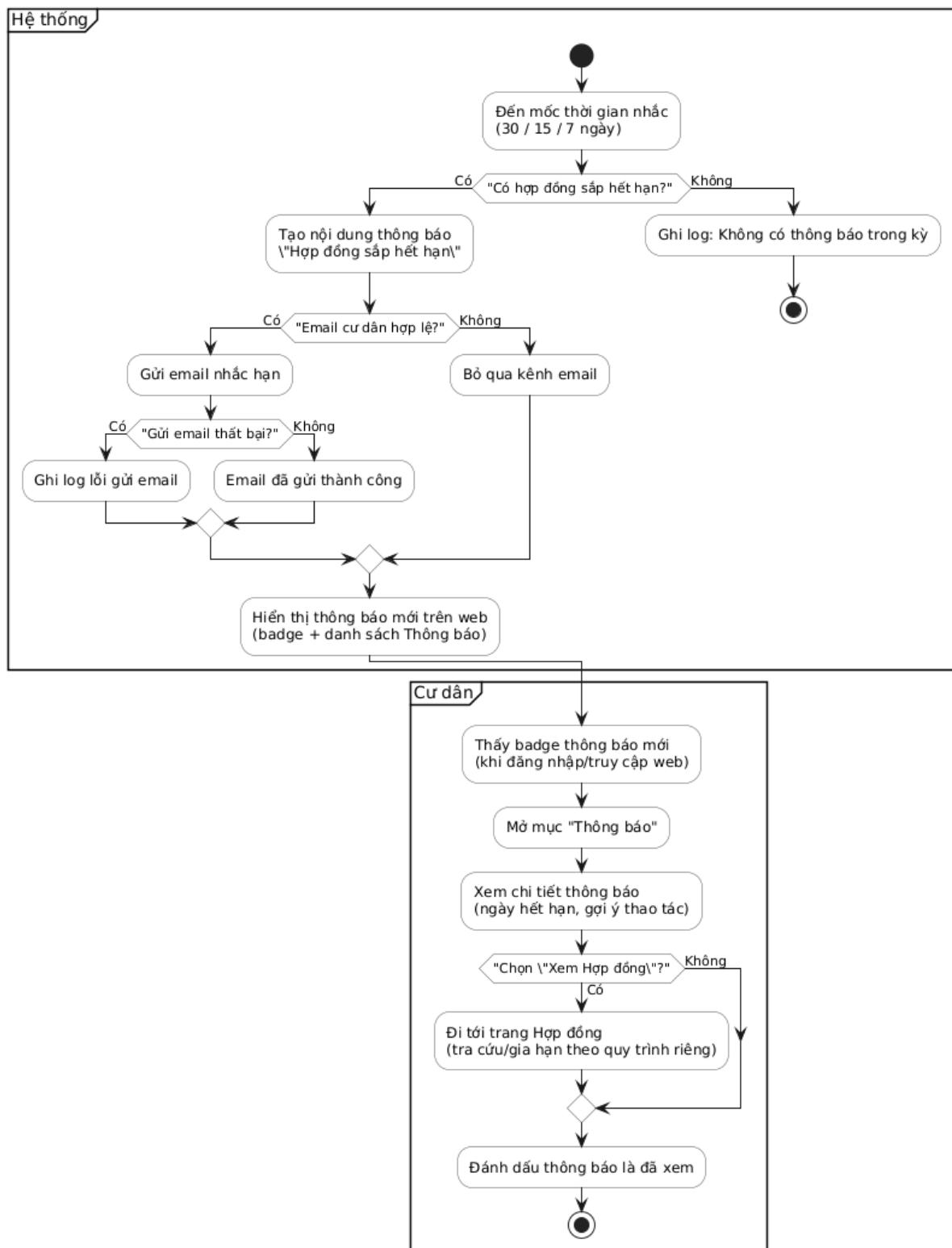


Figure 46 AD Cư dân nhận thông báo khi có hóa đơn mới

## Cư dân nhận thông báo khi có hóa đơn mới

# Báo cáo Thực tập Tốt nghiệp Đại học

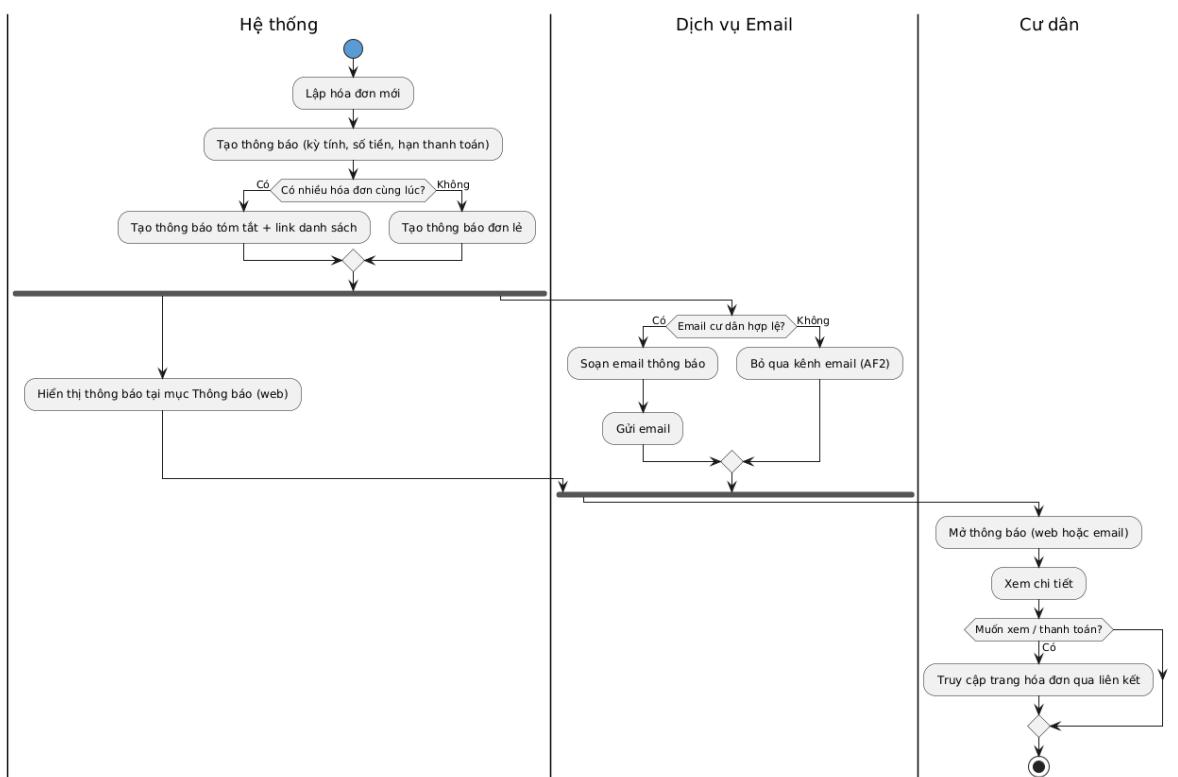


Figure 47 AD Cư dân nhận thông báo khi có hóa đơn mới

## Cư dân nhận thông báo khi sắp hết hạn dịch vụ

## Báo cáo Thực tập Tốt nghiệp Đại học

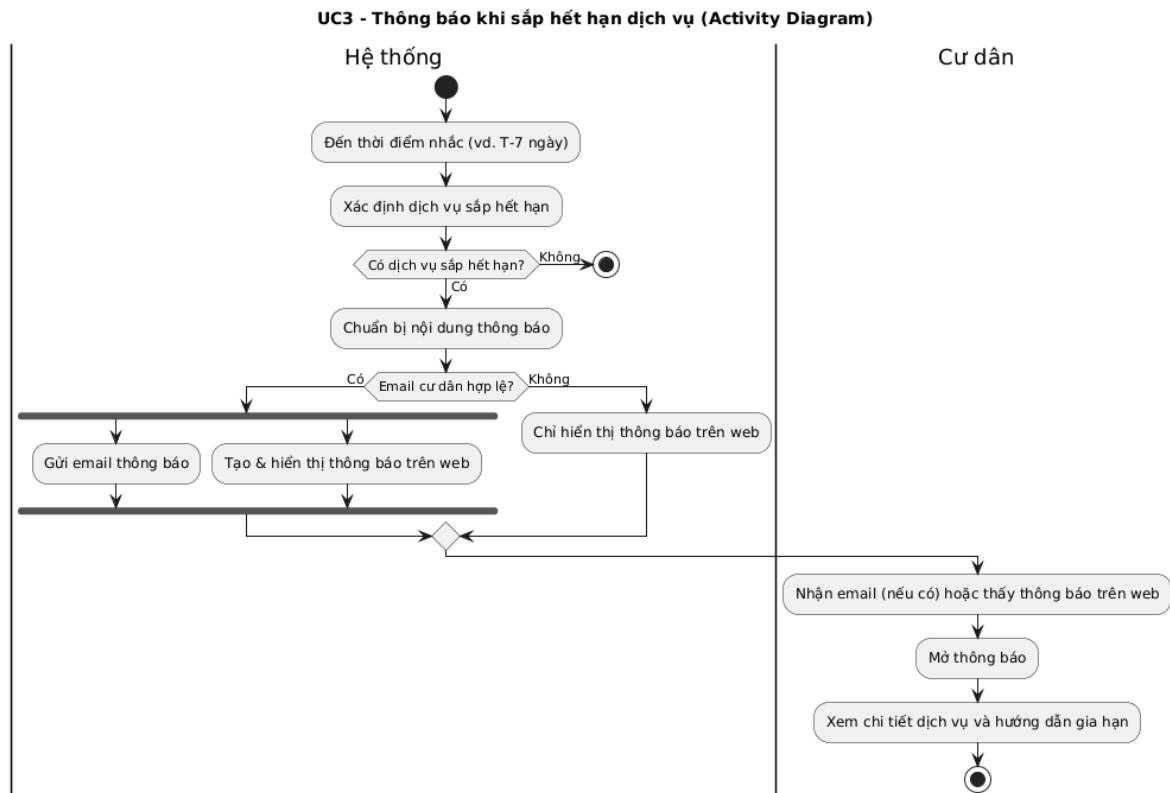


Figure 48 AD Cư dân nhận thông báo khi sắp hết hạn dịch vụ

## Cư dân nhận và xem thông báo qua email & trên web

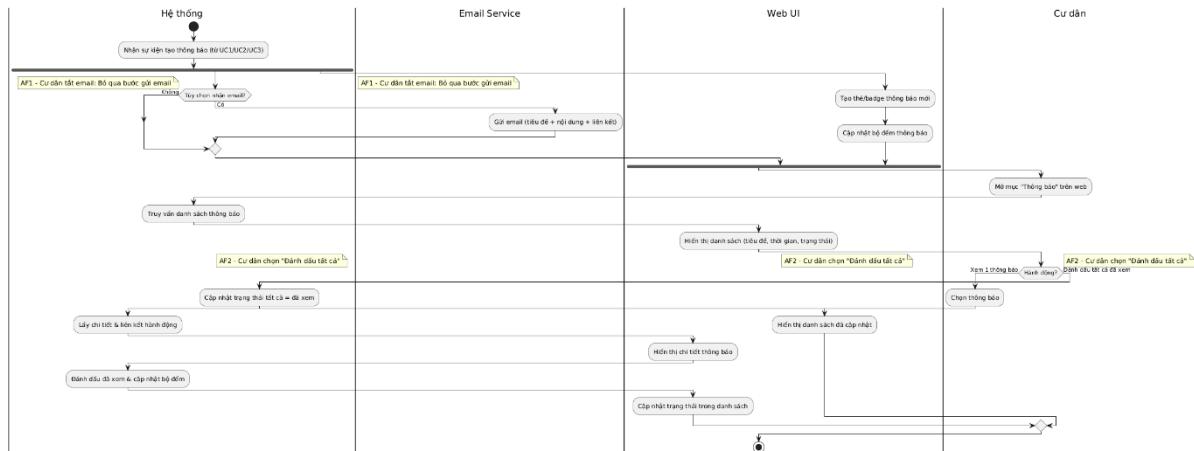


Figure 49 AD Cư dân nhận và xem thông báo qua email & trên web

## 4 Sơ đồ tuần tự (Sequence diagram)

## Nhóm 0: Đặc biệt:

## UC0.1 – Ban quản lý đăng nhập Hệ thống

# Báo cáo Thực tập Tốt nghiệp Đại học

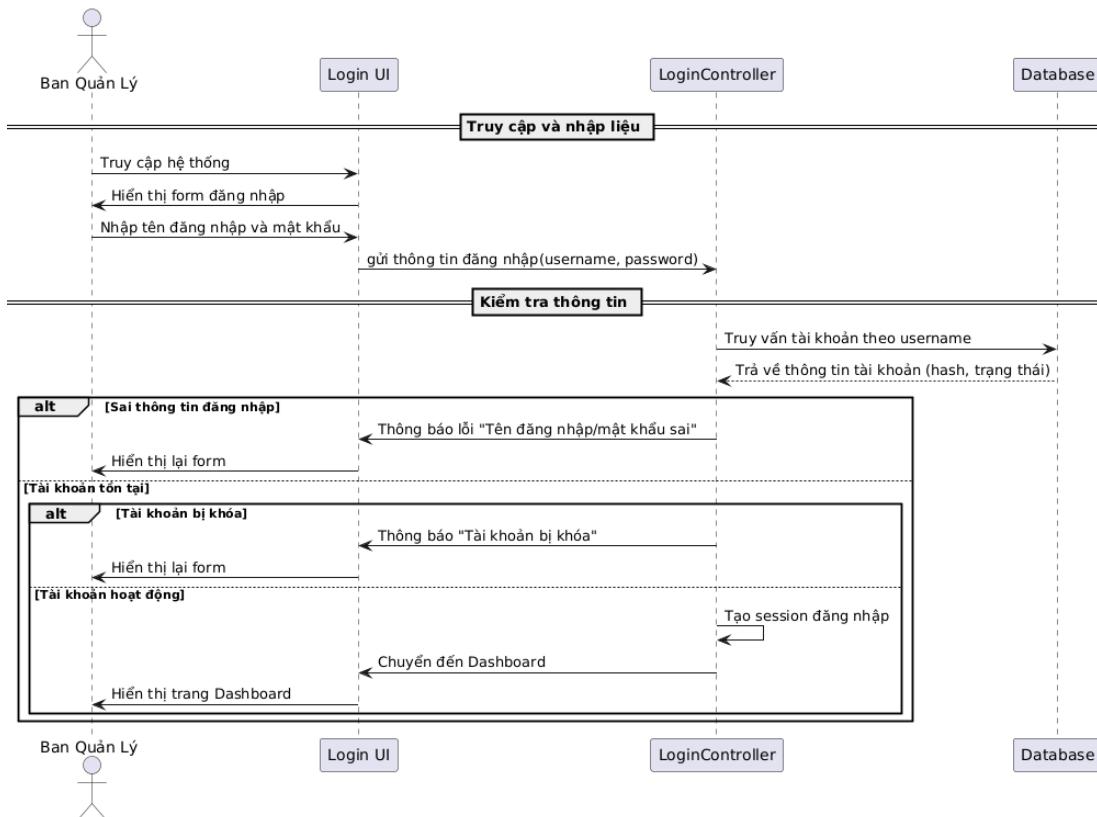


Figure 50 SD Ban quản lý đăng nhập Hệ thống

## Nhóm 1: Quản lý tài khoản cư dân

### UC1.1 – Xem danh sách cư dân đã đăng ký

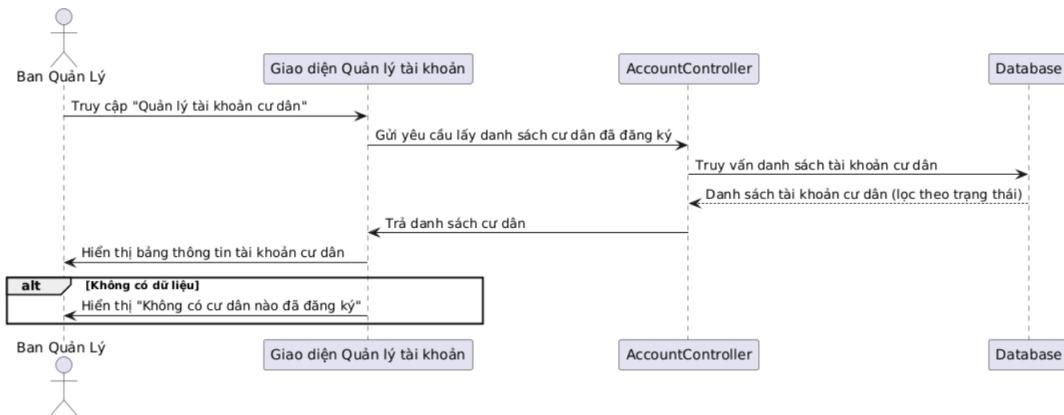


Figure 51 SD Xem danh sách cư dân đã đăng ký

### UC1.2 – Duyệt tài khoản

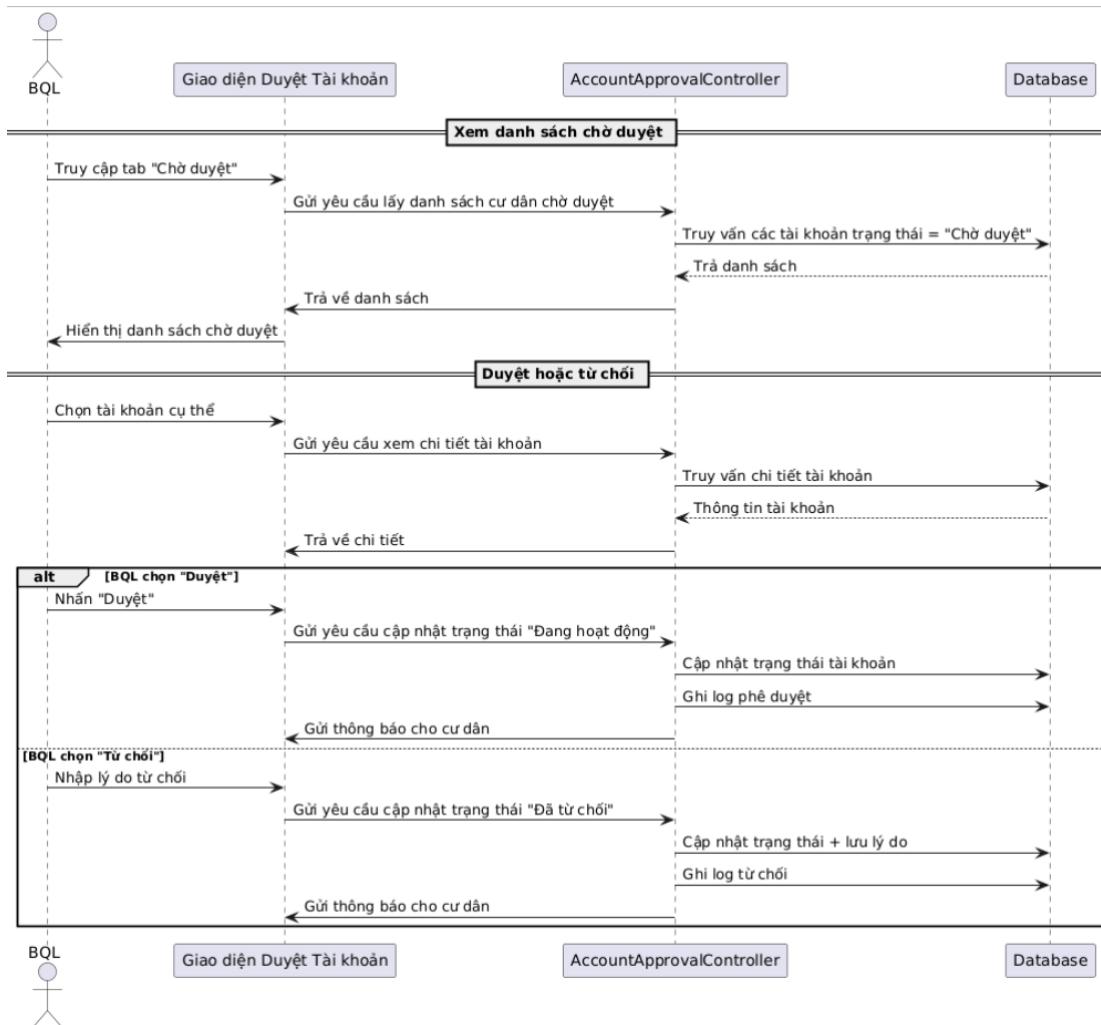


Figure 52 SD Duyệt tài khoản

UC1.3 – Khóa / Mở khóa

# Báo cáo Thực tập Tốt nghiệp Đại học



Figure 53 SD Khóa / Mở khóa

## Nhóm 2. Quản lý dịch vụ

UC2.1 – Khai báo dịch vụ mới

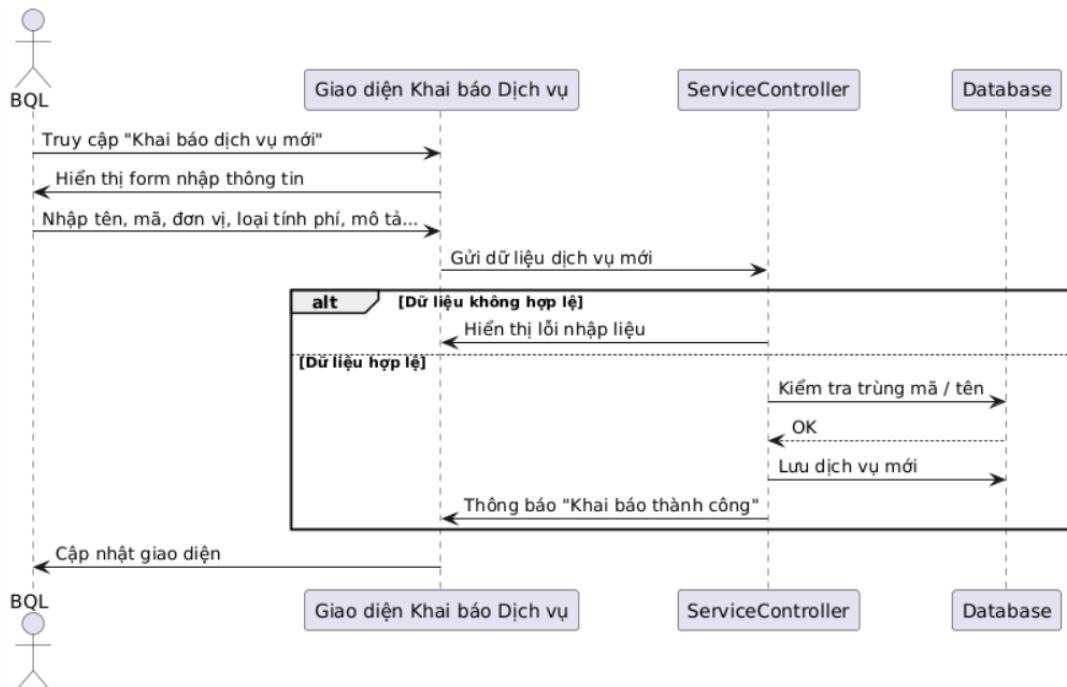


Figure 54 SD Khai báo dịch vụ mới

UC2.2 – Quản lý Cấu hình Chi tiết Dịch vụ

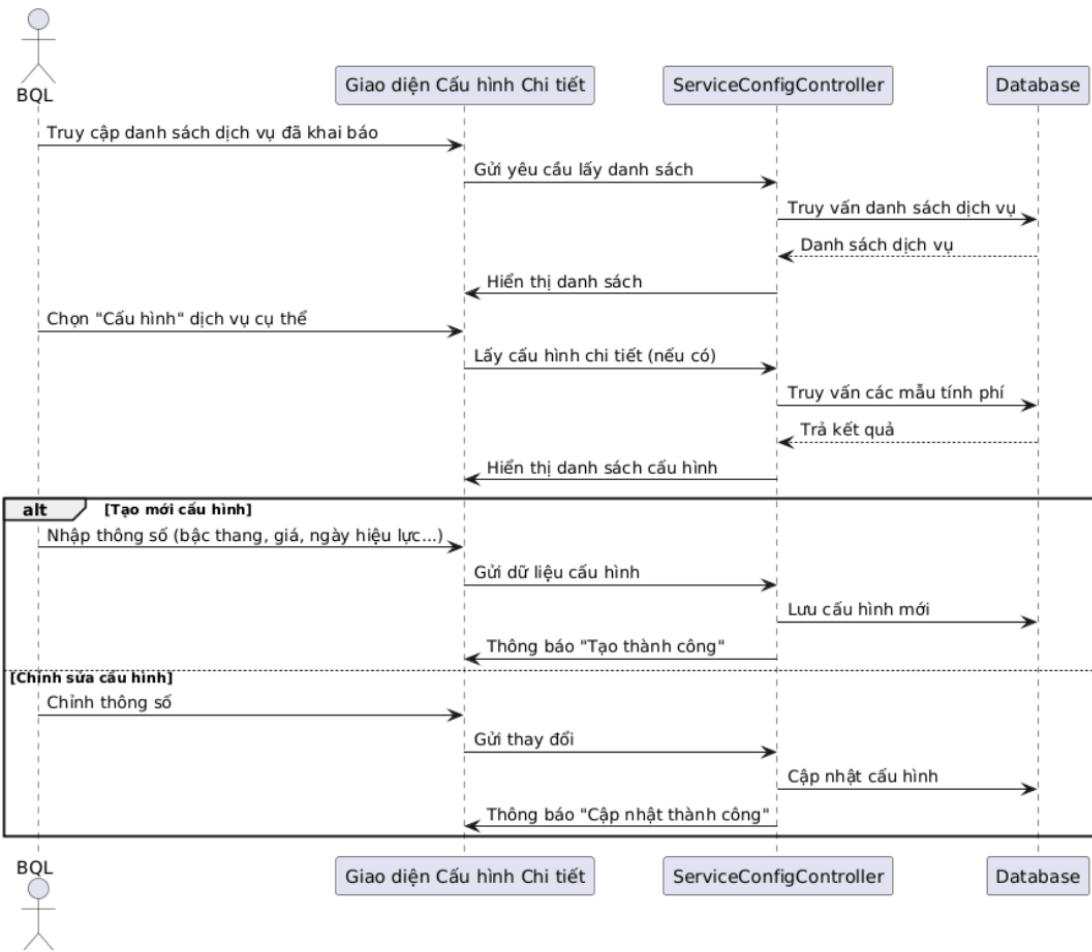


Figure 55 SD Quản lý Cấu hình Chi tiết Dịch vụ

UC2.3 – Chính sửa Thông tin & Trạng thái Dịch vụ đã khai báo

# Báo cáo Thực tập Tốt nghiệp Đại học

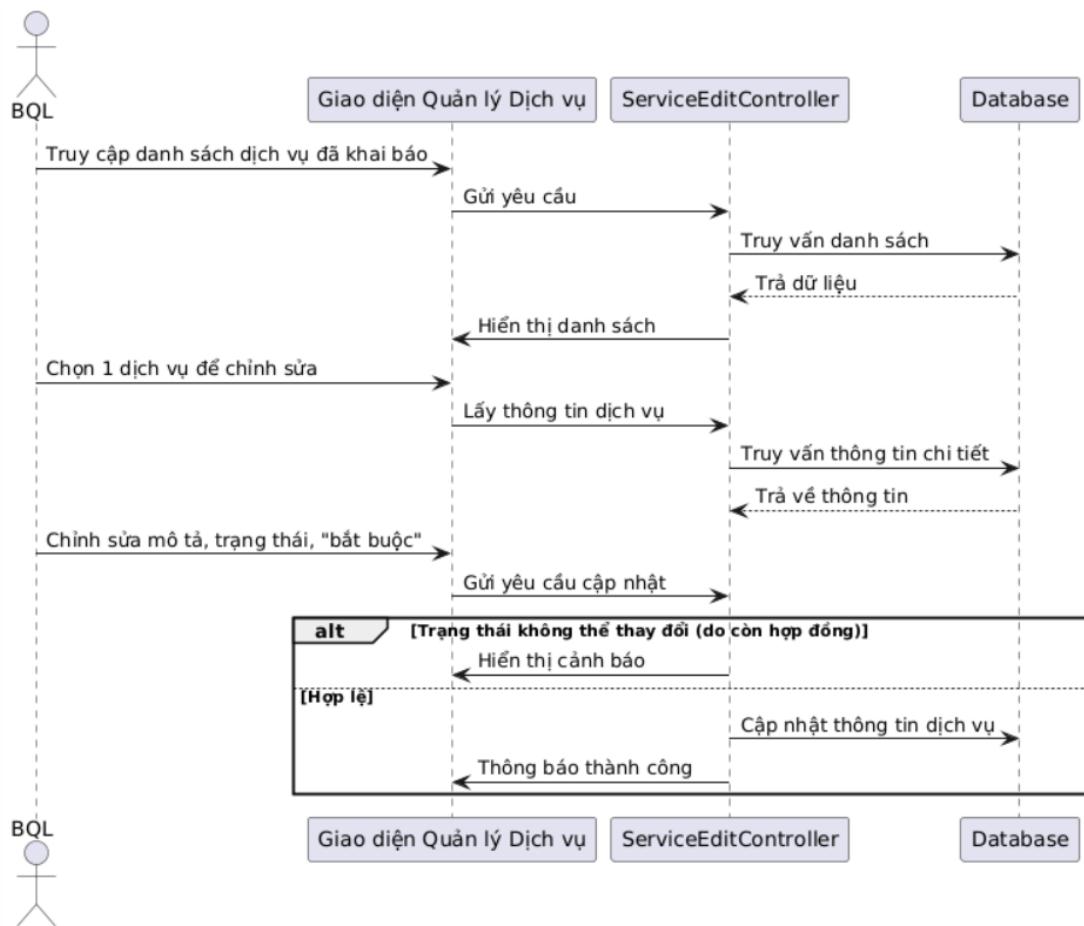


Figure 56 SD Chính sửa Thông tin & Trạng thái Dịch vụ đã khai báo

## UC2.4 – Xem danh sách dịch vụ đã khai báo

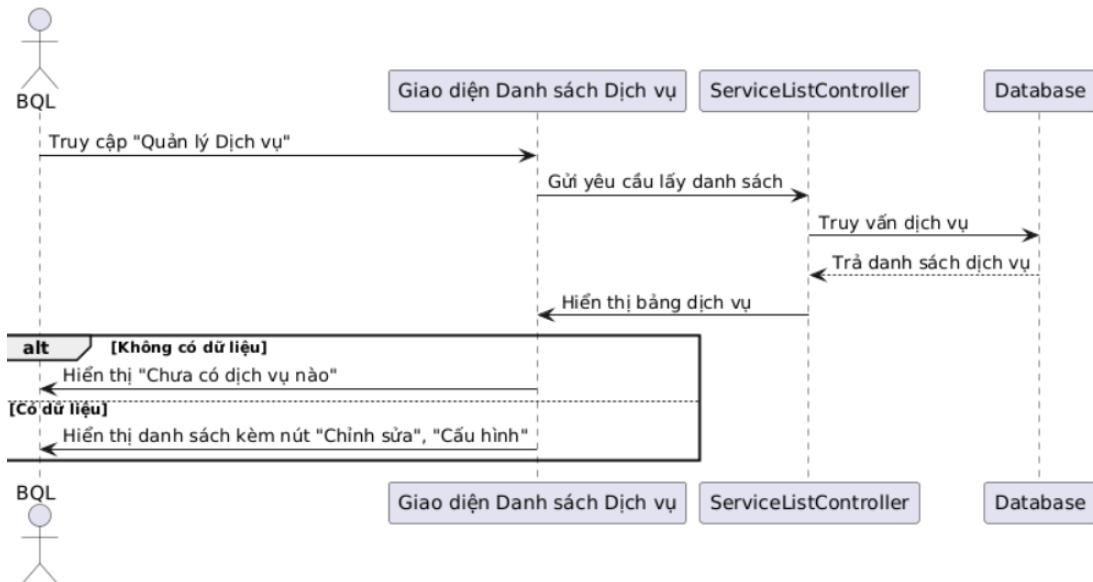


Figure 57 SD Xem danh sách dịch vụ đã khai báo

## Nhóm 3. Quản lý mẫu hợp đồng

### UC3.1 – Tạo mẫu hợp đồng dịch vụ

# Báo cáo Thực tập Tốt nghiệp Đại học

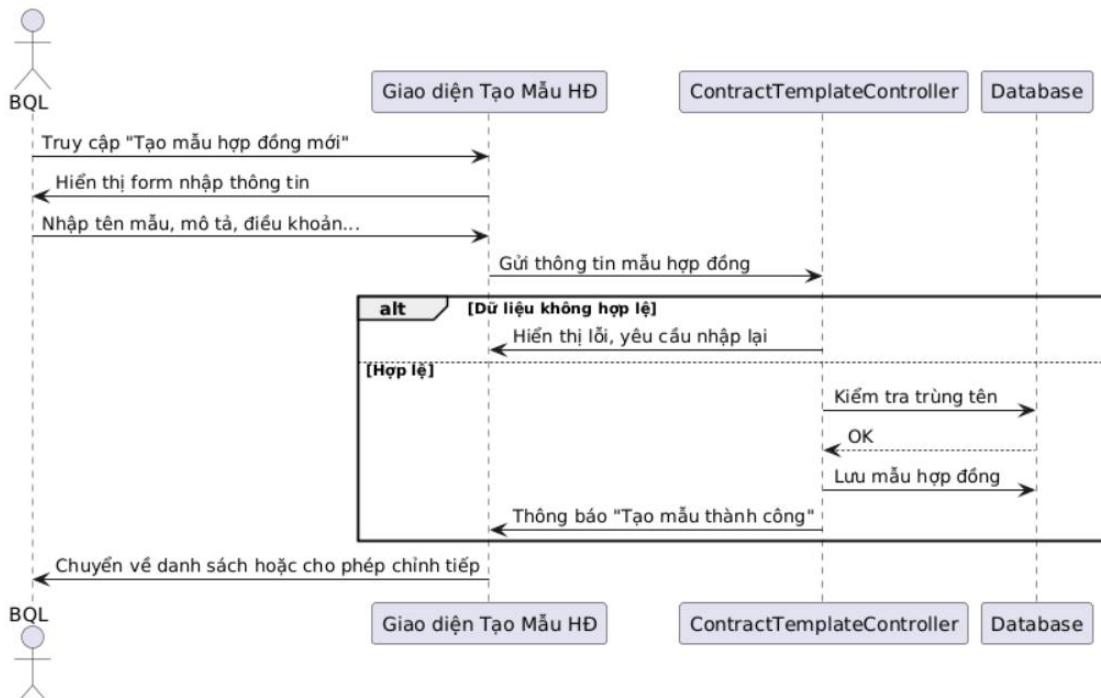


Figure 58 SD Tạo mẫu hợp đồng dịch vụ

UC3.2 – Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ

# Báo cáo Thực tập Tốt nghiệp Đại học

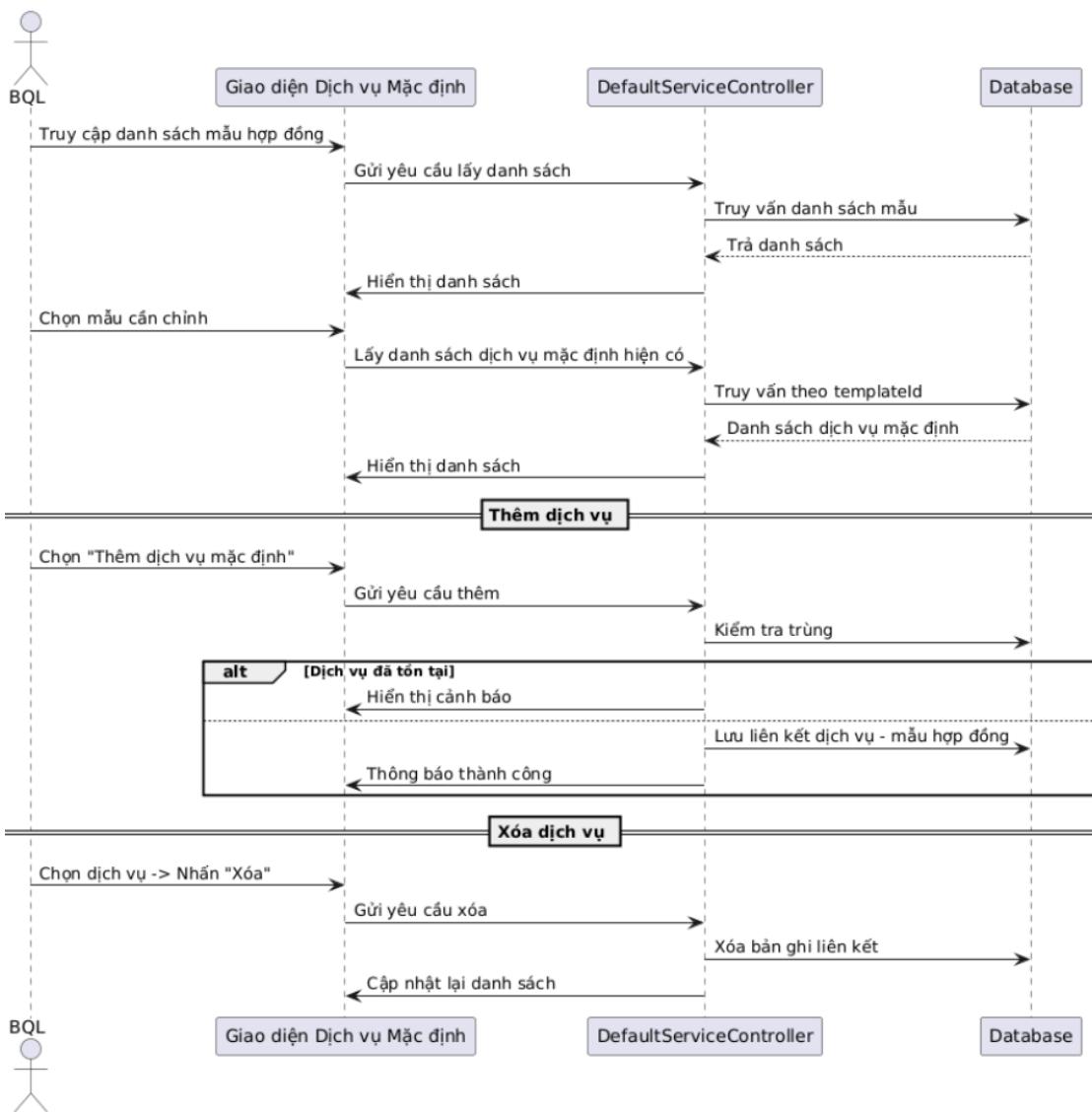


Figure 59 SD Quản lý các Dịch vụ Mặc định trong Mẫu Hợp đồng Dịch vụ

UC3.3 – Xem danh sách mẫu hợp đồng

## Báo cáo Thực tập Tốt nghiệp Đại học

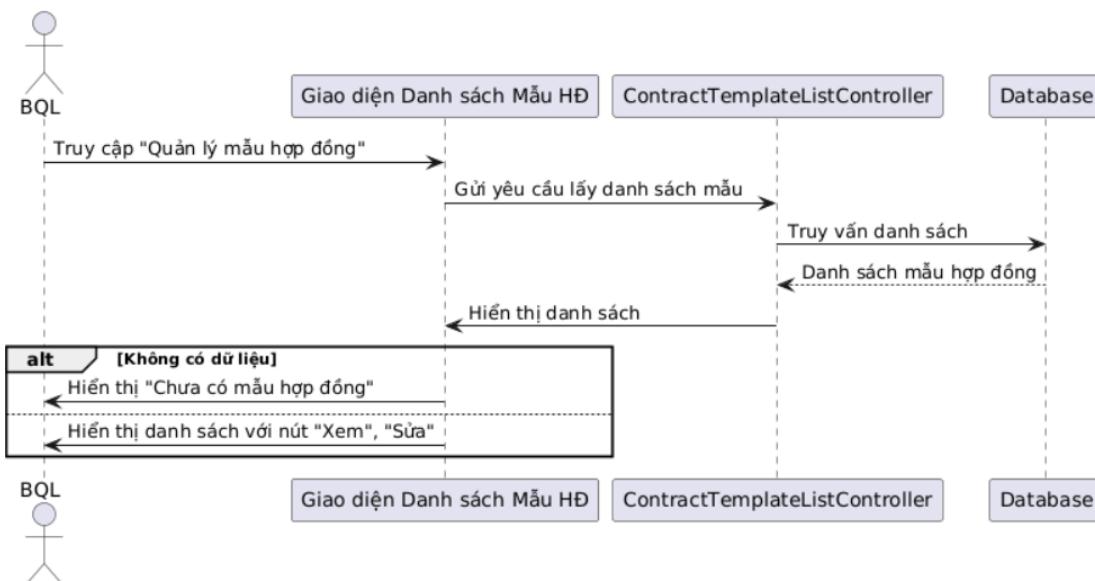


Figure 60 SD Xem danh sách mẫu hợp đồng

### UC3.4 – Chính sửa mẫu

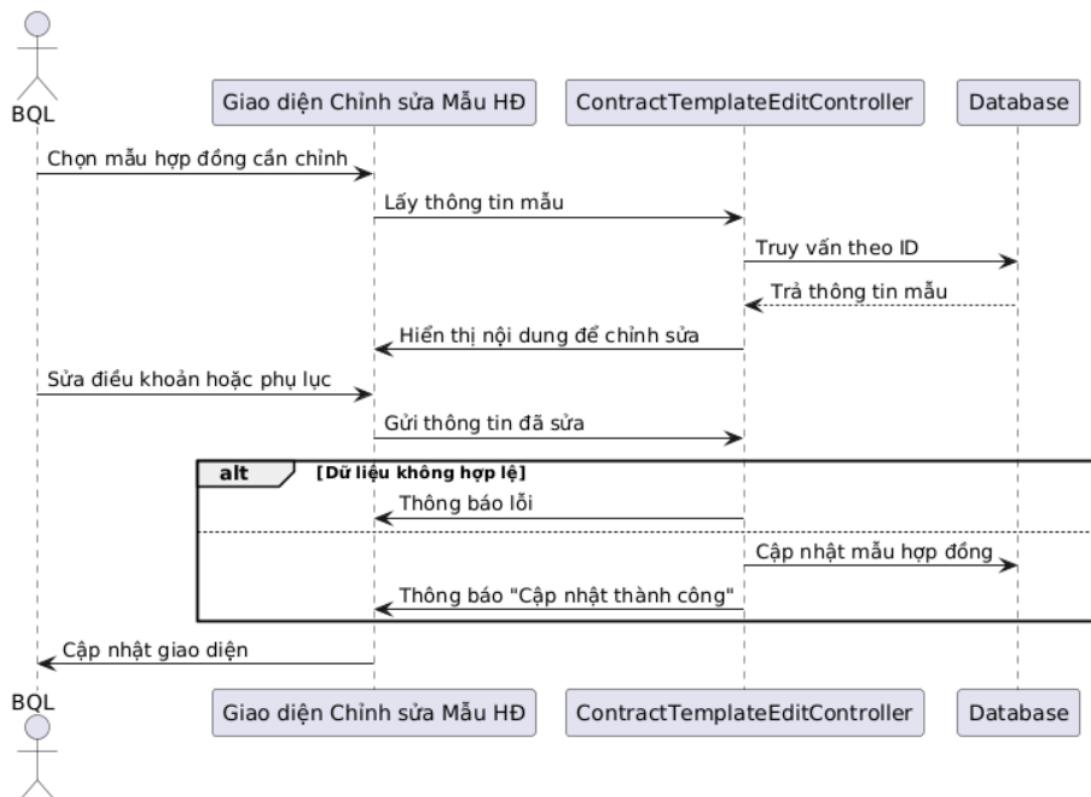


Figure 61 SD Chính sửa mẫu

## Nhóm 4. Quản lý hợp đồng cư dân

Use Case UC4.1 – Xem danh sách hợp đồng cá nhân

## Báo cáo Thực tập Tốt nghiệp Đại học

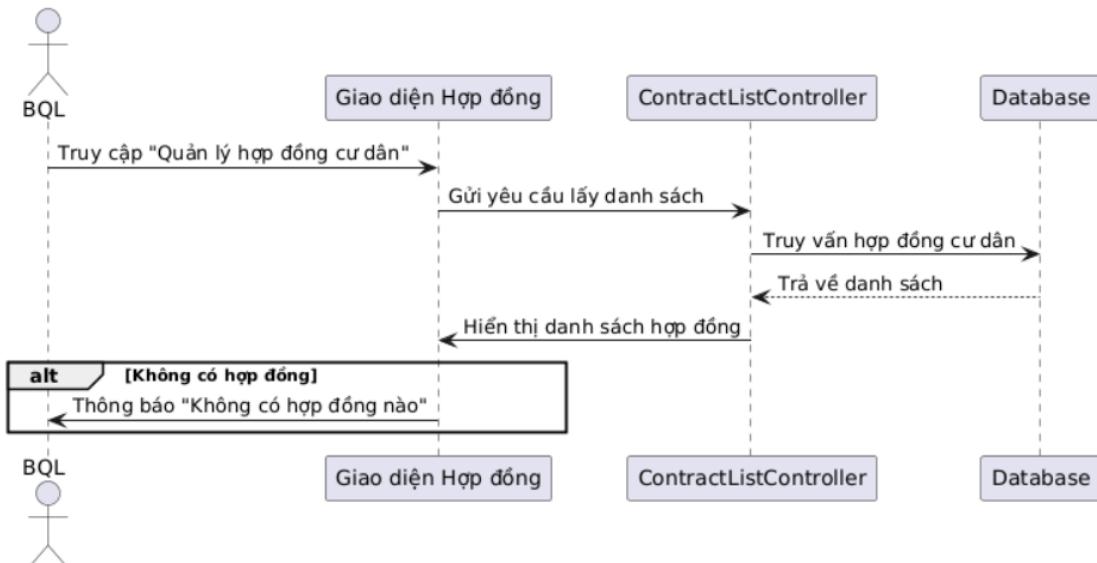


Figure 62 SD Xem danh sách hợp đồng cá nhân

### UC4.2 – Xem chi tiết hợp đồng cụ thể

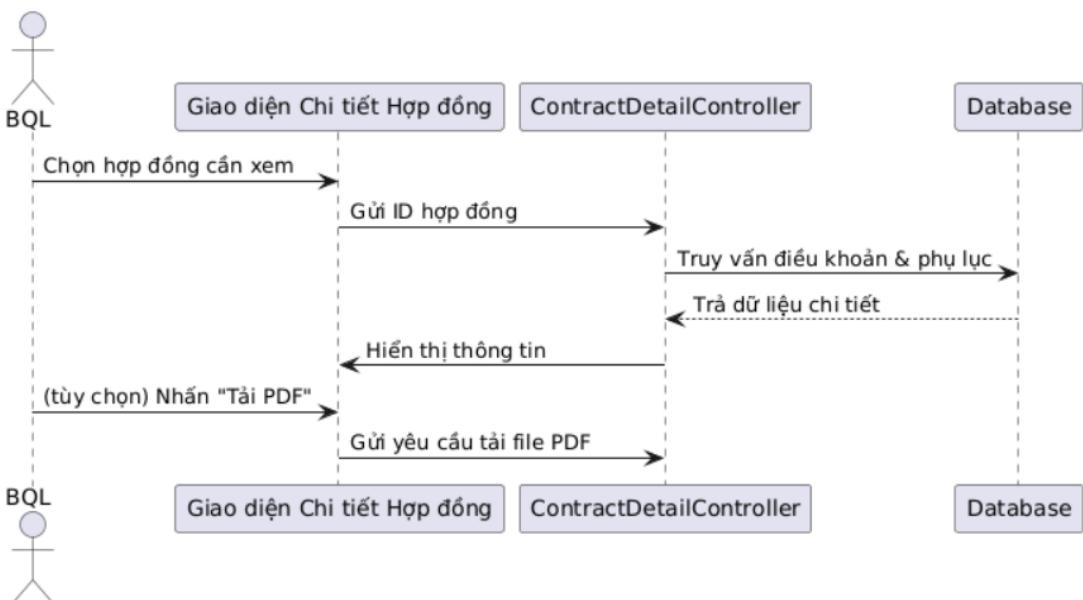


Figure 63 SD Xem chi tiết hợp đồng cụ thể

### UC4.3 – Tự động chấm dứt do vi phạm

# Báo cáo Thực tập Tốt nghiệp Đại học

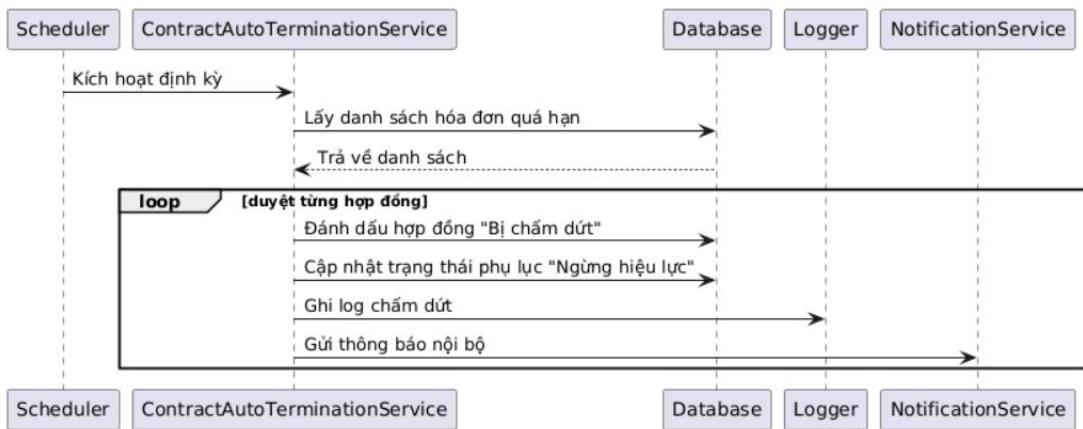


Figure 64 SD Tự động chấm dứt do vi phạm

## UC4.4 – BQL chấm dứt do vi phạm nội quy

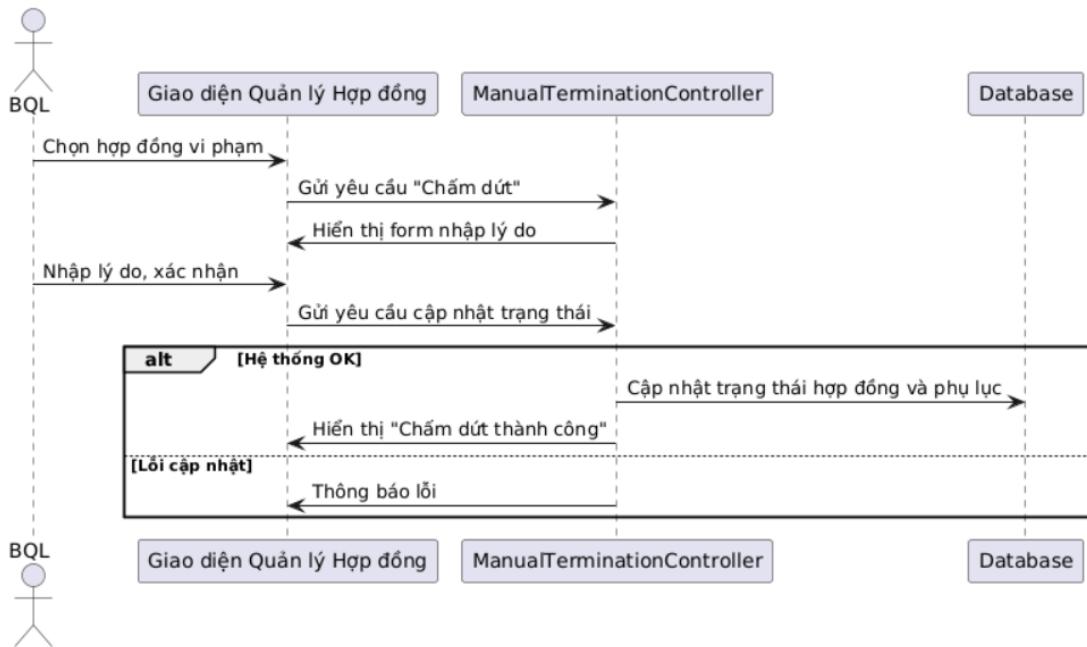


Figure 65 SD BQL chấm dứt do vi phạm nội quy

## UC4.5 – BQL đơn phương chấm dứt một dịch vụ riêng lẻ

# Báo cáo Thực tập Tốt nghiệp Đại học

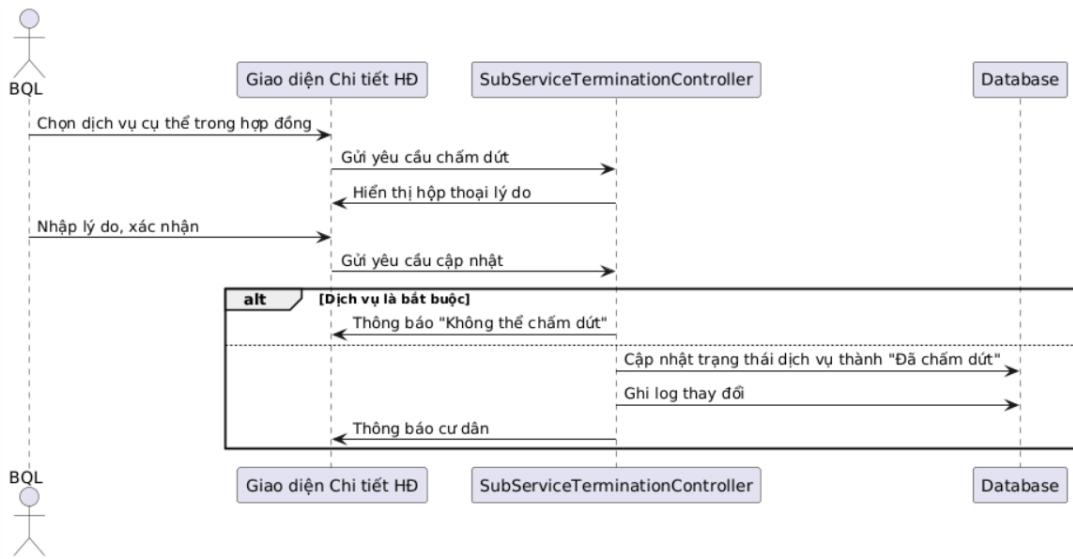


Figure 66 SD BQL đơn phương chấm dứt một dịch vụ riêng lẻ

## UC4.6 – BQL Duyệt Hợp đồng Dịch vụ mới

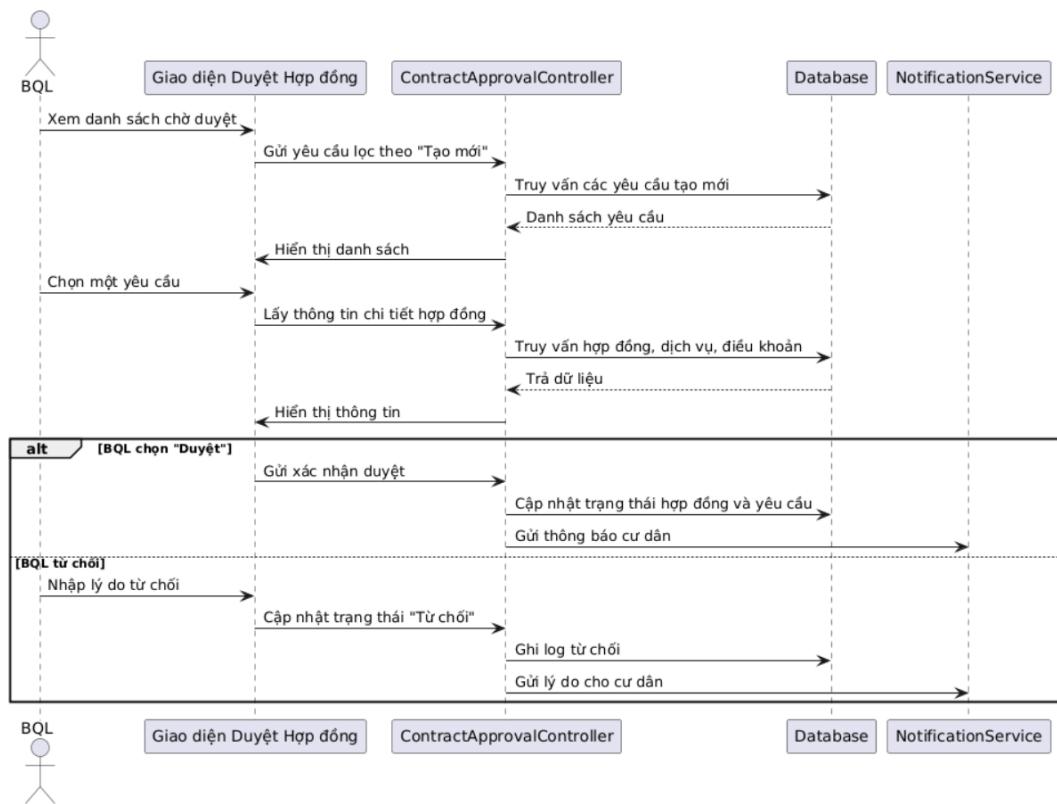


Figure 67 SD BQL Duyệt Hợp đồng Dịch vụ mới

## UC4.7 – BQL Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng

# Báo cáo Thực tập Tốt nghiệp Đại học

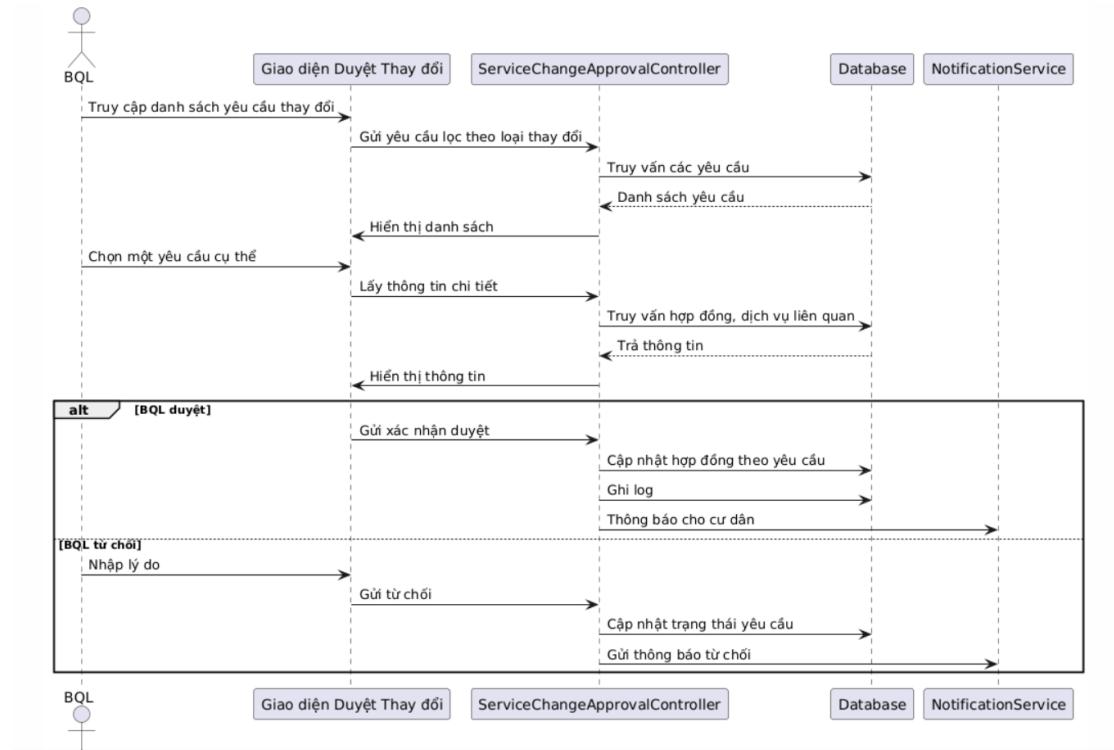


Figure 68 SD BQL Duyệt Yêu cầu Thay đổi Dịch vụ trong Hợp đồng

## Nhóm 5. Quản lý hóa đơn & chỉ số

UC5.1 – Tự động lập hóa đơn



Figure 69 SD Tự động lập hóa đơn

UC5.2 – Ghi chỉ số điện/nước

# Báo cáo Thực tập Tốt nghiệp Đại học

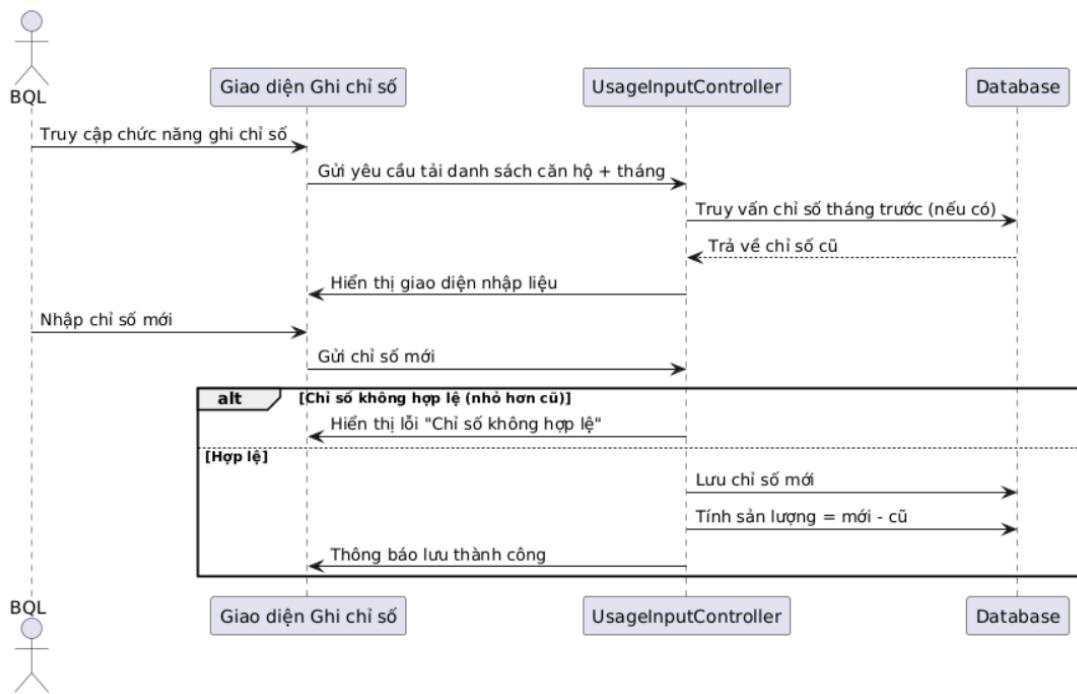


Figure 70 SD Ghi chỉ số điện/nước

## UC5.3 Xác nhận thanh toán

# Báo cáo Thực tập Tốt nghiệp Đại học

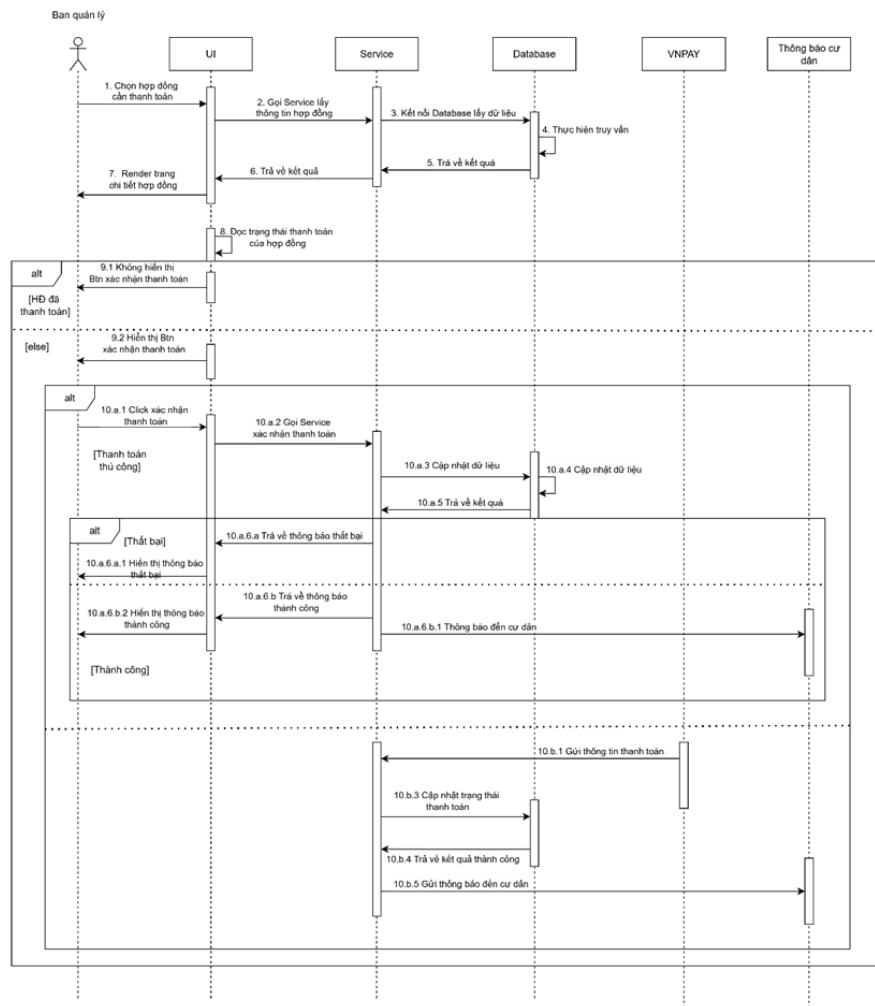


Figure 71 SD Xác nhận thanh toán

## Nhóm 6: Quản lý tài khoản cá nhân (Cư dân)

### UC6.1 Đăng ký tài khoản

# Báo cáo Thực tập Tốt nghiệp Đại học

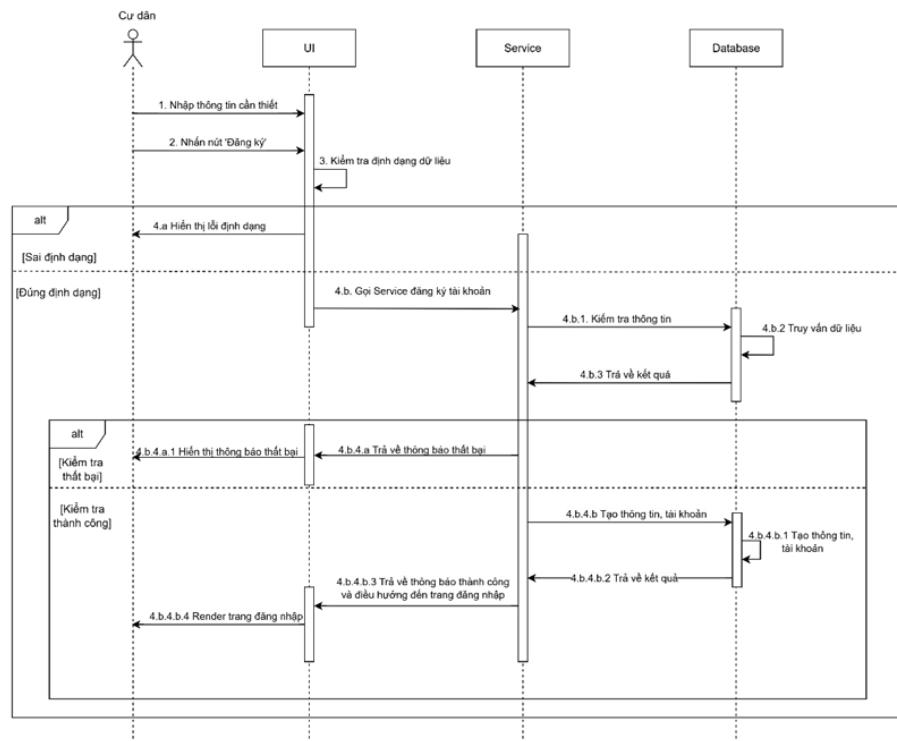


Figure 72 SD Đăng ký tài khoản

## UC6.1 Cập nhật thông tin cá nhân

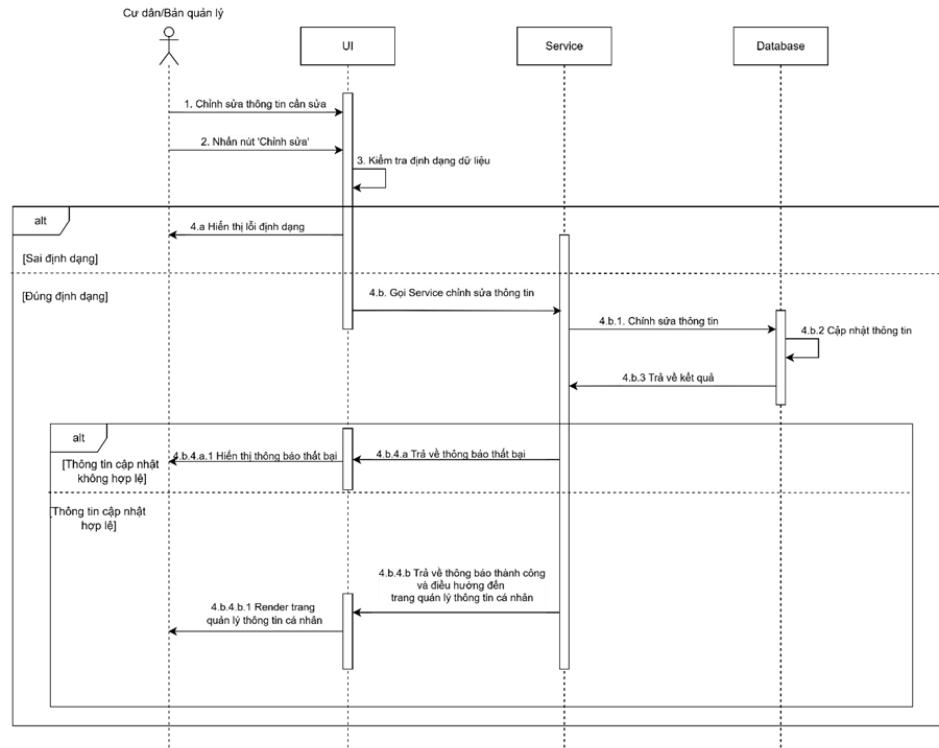


Figure 73 SD Cập nhật thông tin cá nhân

## Nhóm 7: Đăng ký dịch vụ hợp đồng

# Báo cáo Thực tập Tốt nghiệp Đại học

## UC7.1 Lập hợp đồng

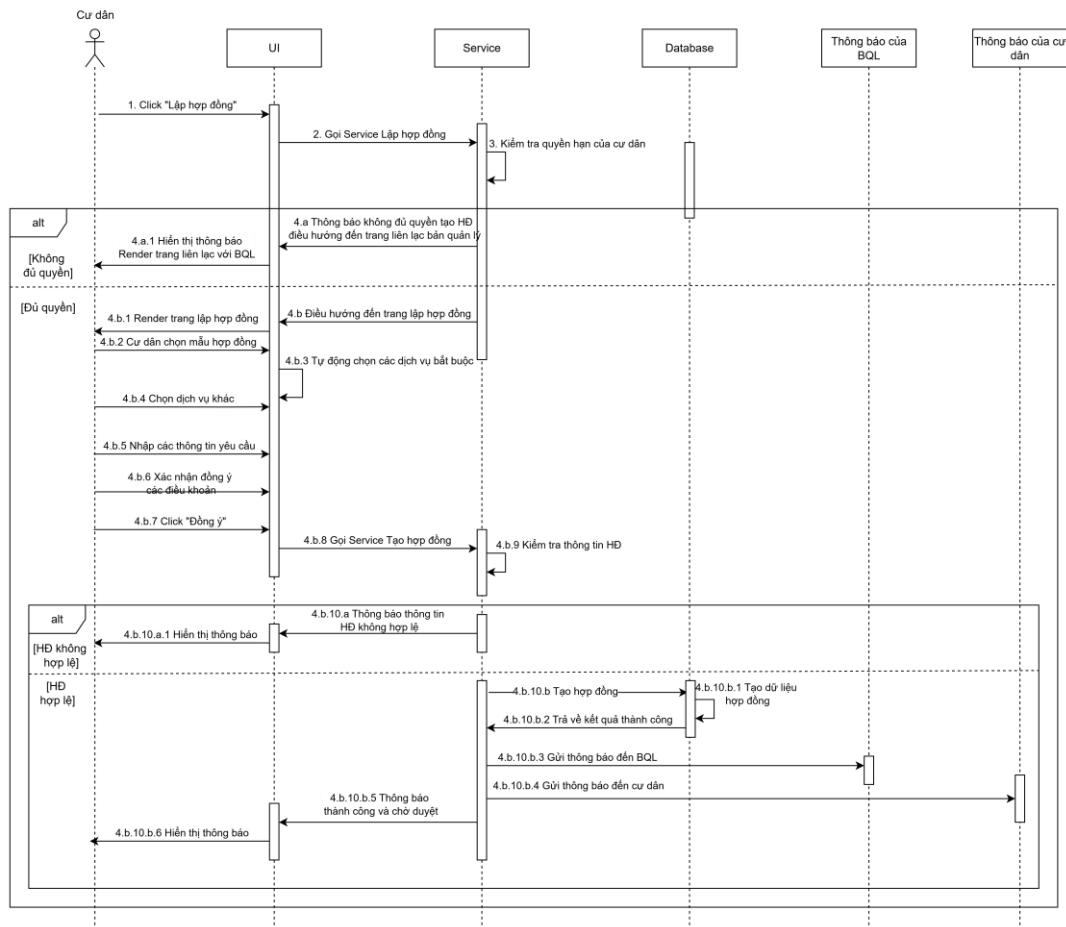


Figure 74 SD Lập hợp đồng

## UC7.2 Xem điều khoản dịch vụ

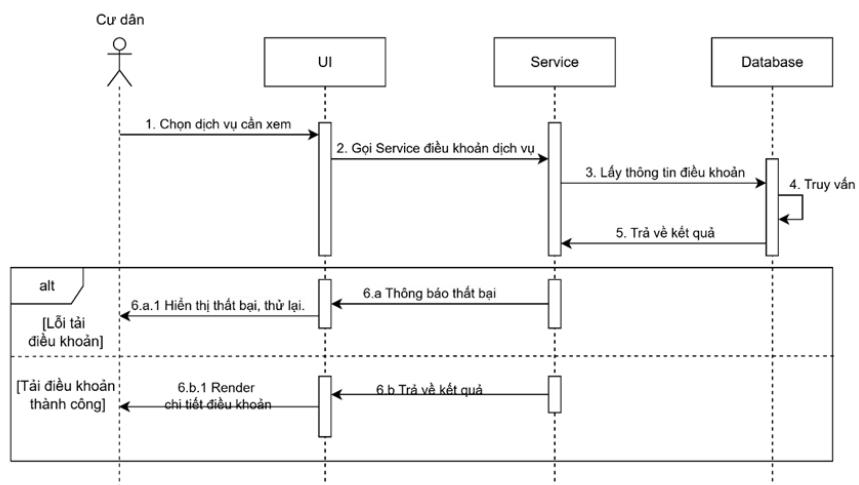


Figure 75 SD Xem điều khoản dịch vụ

# Báo cáo Thực tập Tốt nghiệp Đại học

## UC7.3 Thay đổi dịch vụ hợp đồng

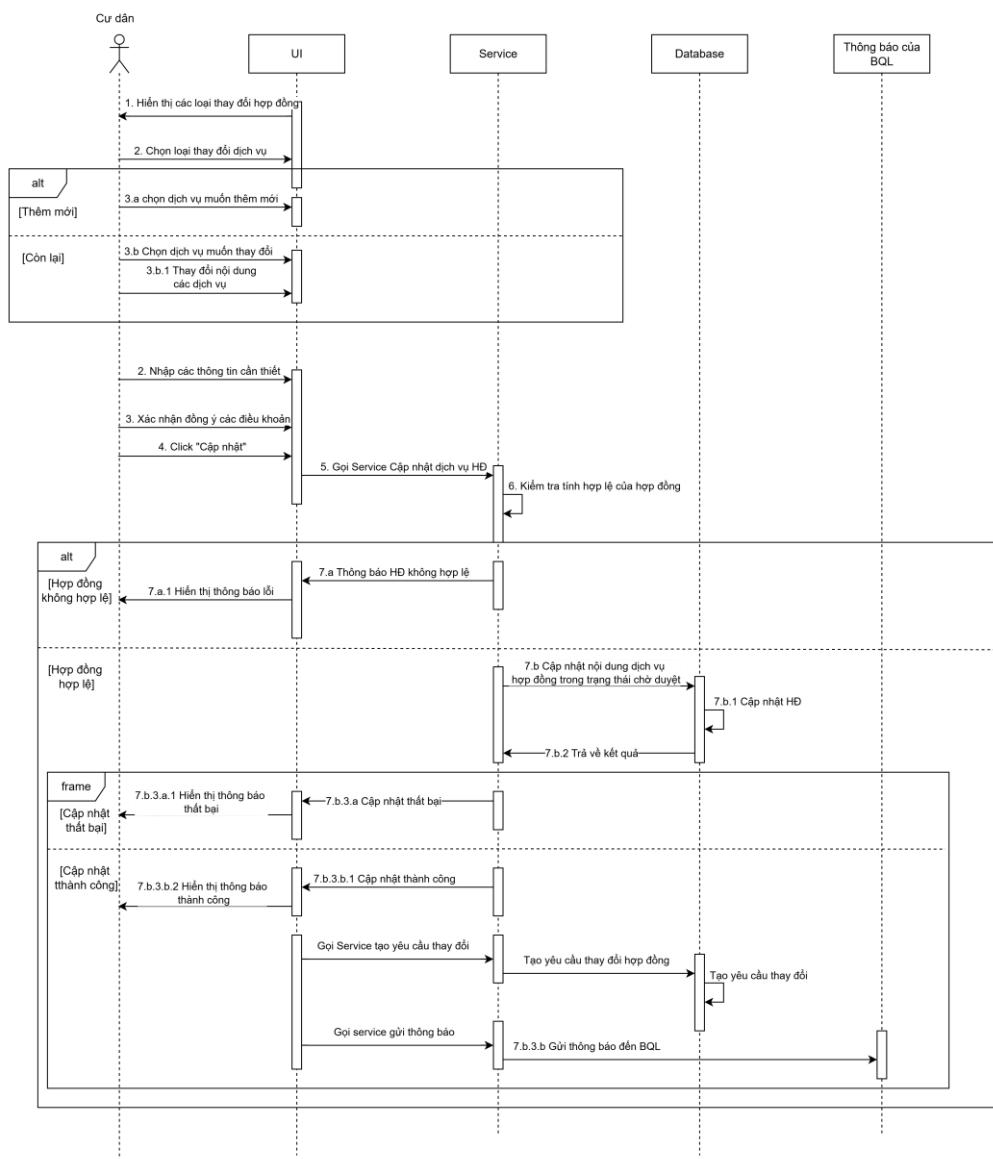


Figure 76 SD Thay đổi dịch vụ hợp đồng

## Nhóm 8: Tra cứu hợp đồng

# Báo cáo Thực tập Tốt nghiệp Đại học

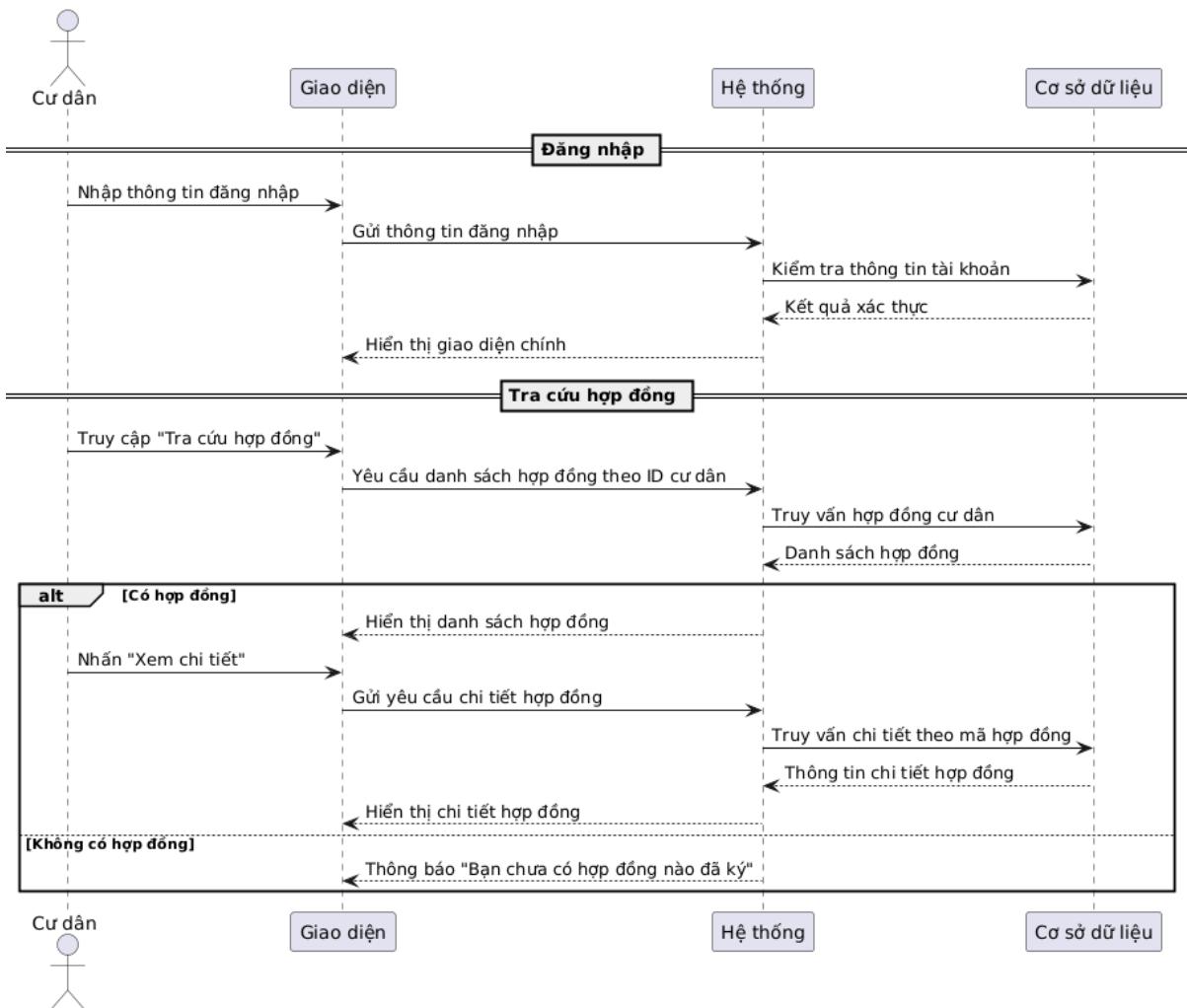


Figure 77 SD Tra cứu hợp đồng

## Nhóm 9: Thanh toán online

**Thanh toán thủ công (chuyển khoản, tiền mặt)**

# Báo cáo Thực tập Tốt nghiệp Đại học

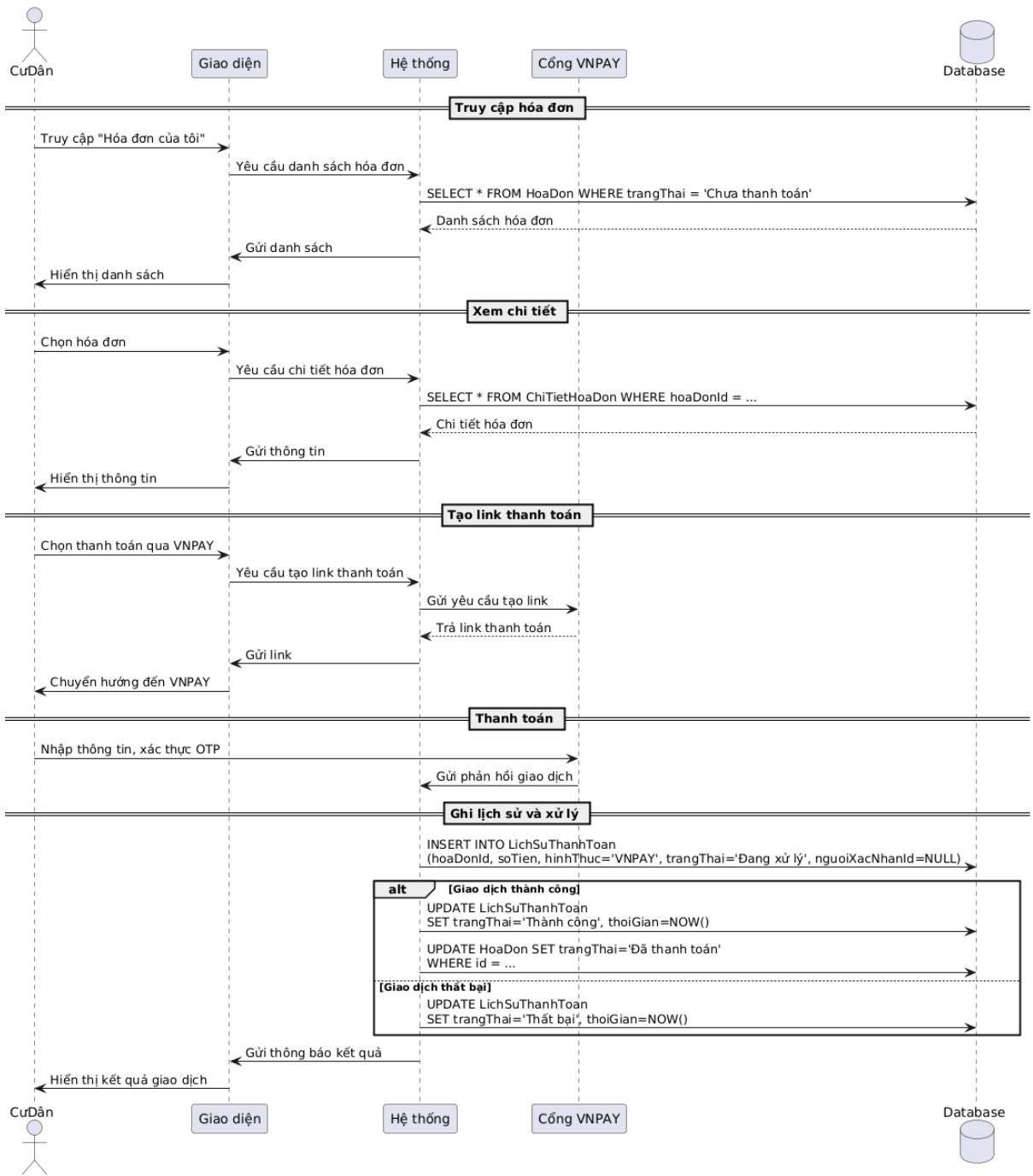


Figure 78 SD Thanh toán thủ công (chuyển khoản, tiền mặt)

## Thanh toán online qua VNPay

# Báo cáo Thực tập Tốt nghiệp Đại học

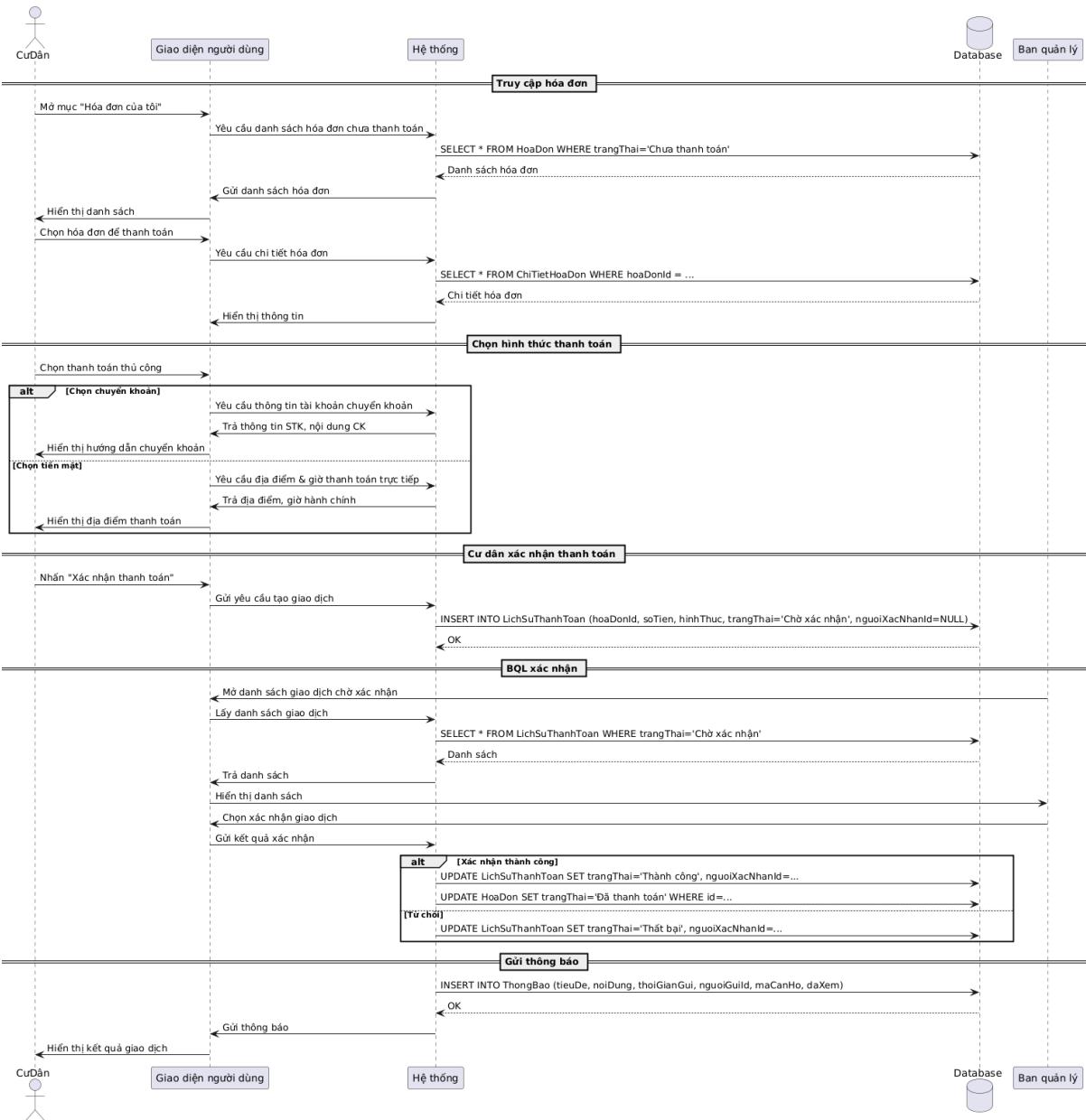


Figure 79 SD Thanh toán online qua VNPAY

## Nhóm 10: Nhận thông báo

Cư dân nhận nhắc nhở thời hạn hợp đồng

# Báo cáo Thực tập Tốt nghiệp Đại học

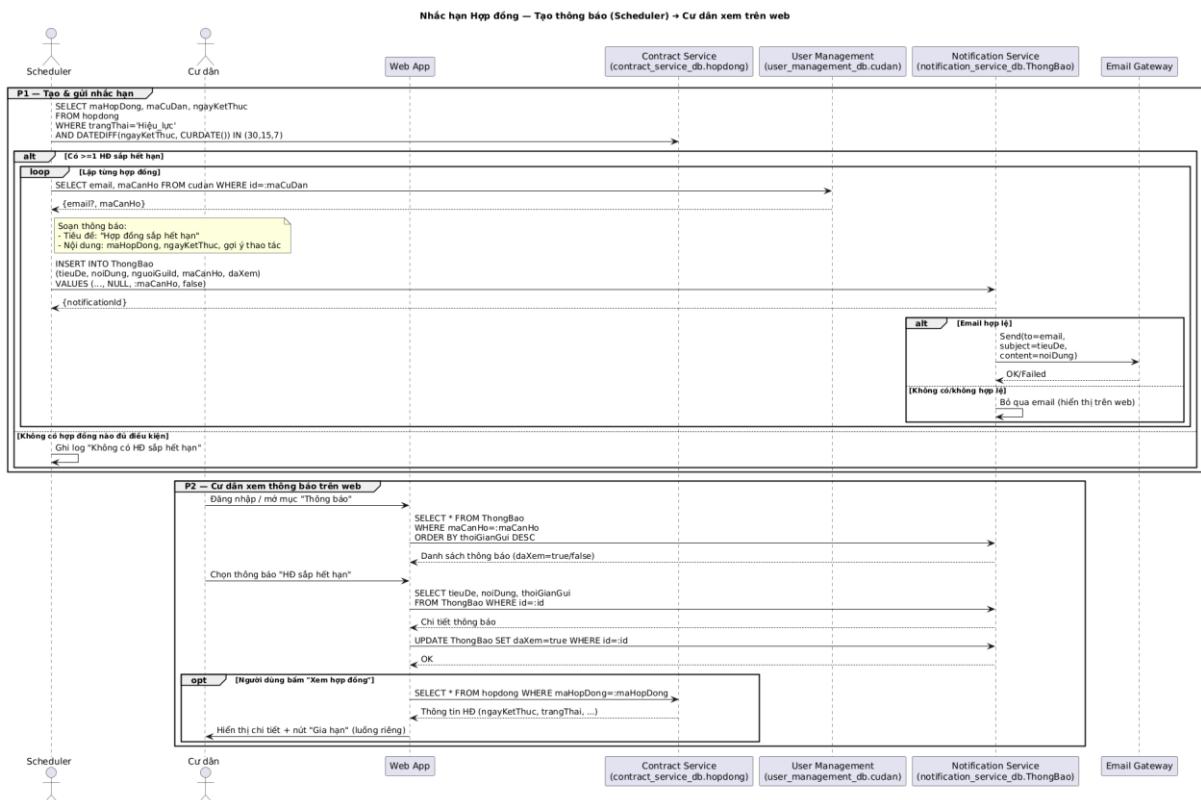


Figure 80 SD Cư dân nhận nhắc nhở thời hạn hợp đồng

## Cư dân nhận thông báo khi có hóa đơn mới

## Báo cáo Thực tập Tốt nghiệp Đại học

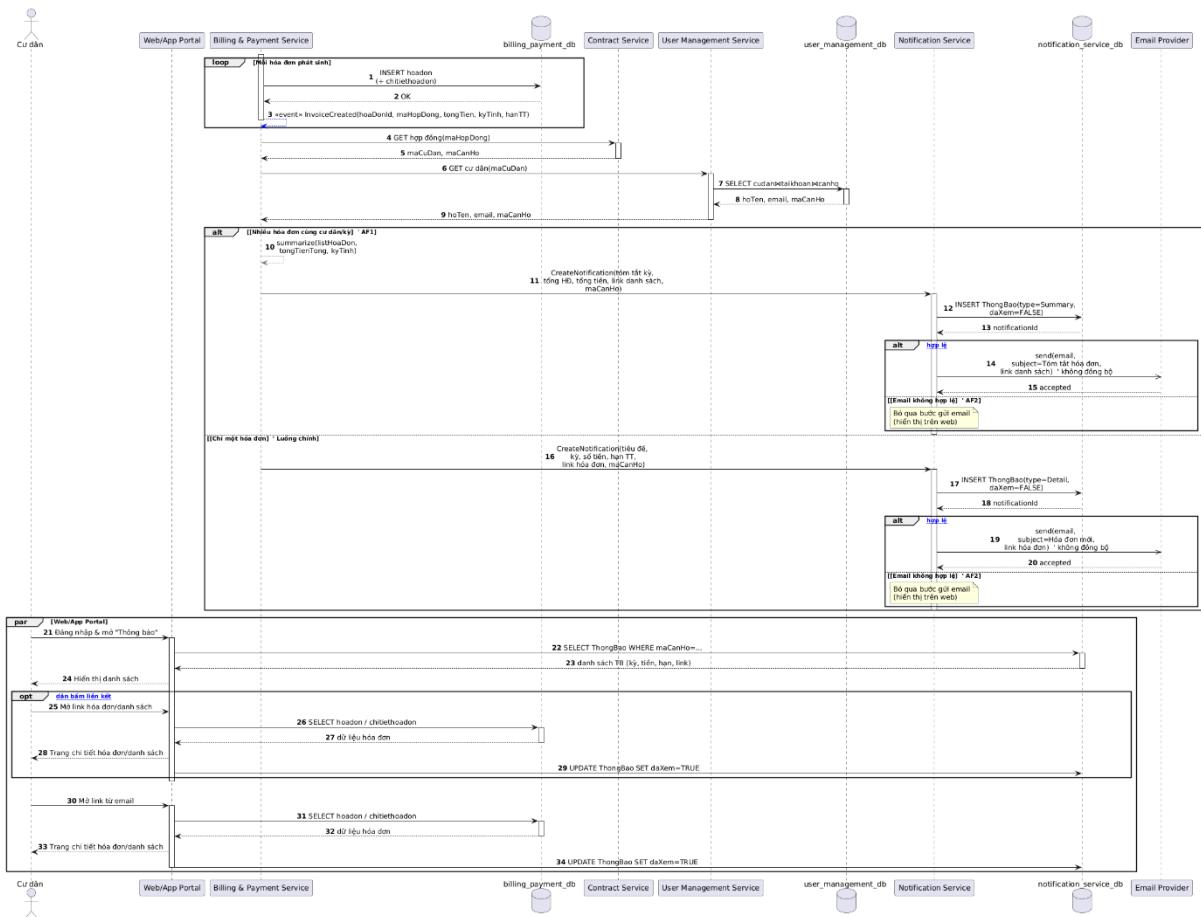


Figure 81 SD Cư dân nhận thông báo khi có hóa đơn mới

## Cư dân nhận thông báo khi sắp hết hạn dịch vụ

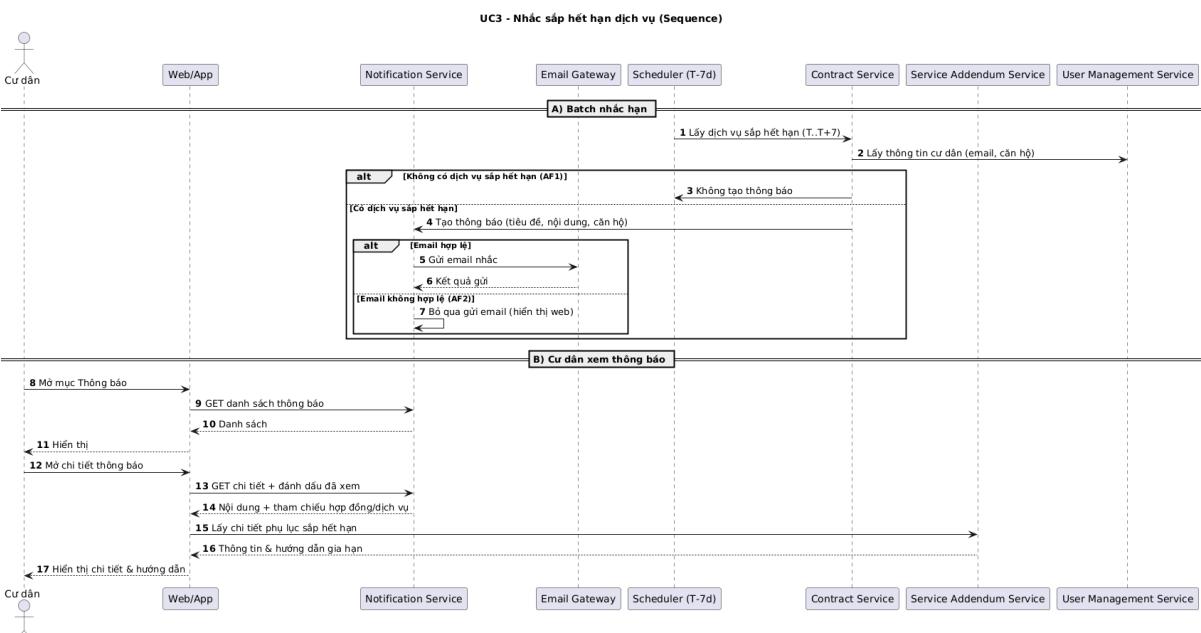


Figure 82 SD Cư dân nhận thông báo khi sắp hết hạn dịch vụ

## Cư dân nhận và xem thông báo qua email & trên web

# Báo cáo Thực tập Tốt nghiệp Đại học

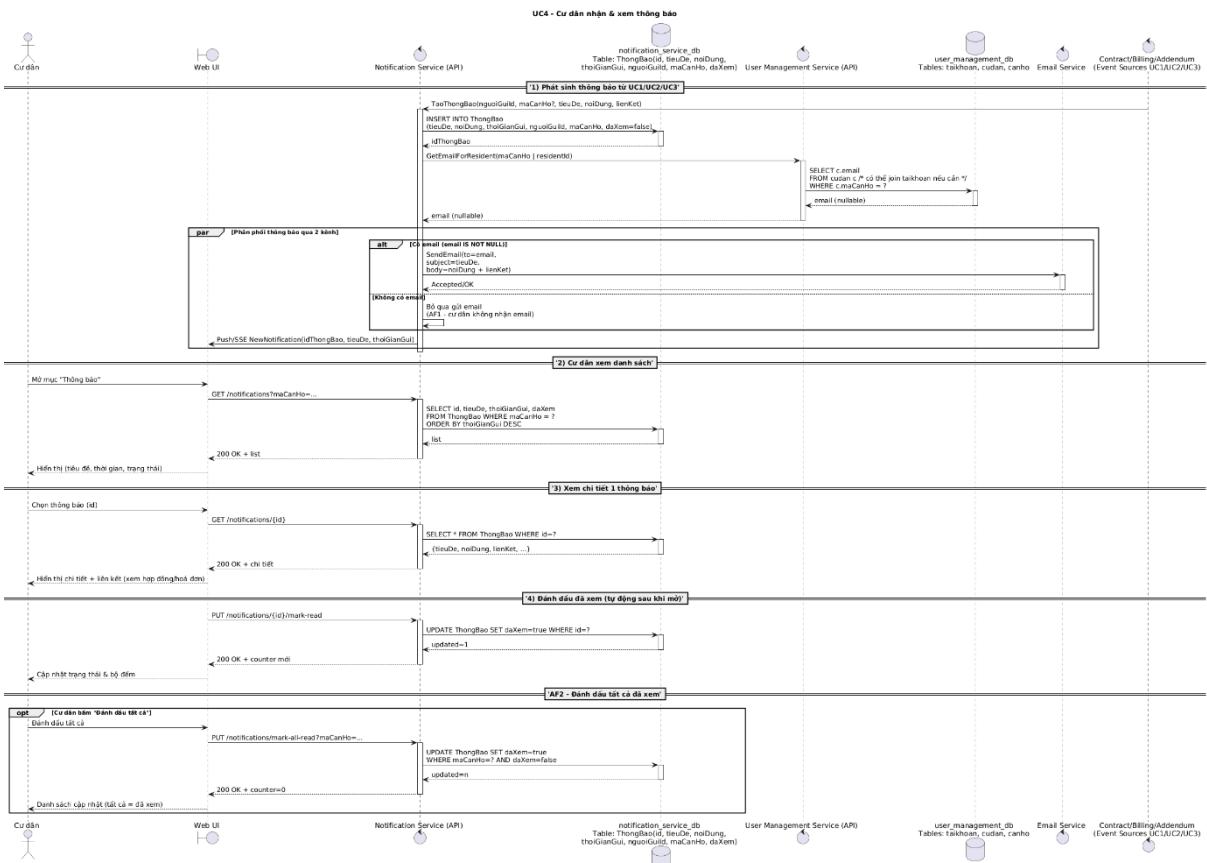


Figure 83 SD Cư dân nhận và xem thông báo qua email & trên web

## 5 Thiết kế kiến trúc hệ thống tổng thể (Diagram Microservices)

# Báo cáo Thực tập Tốt nghiệp Đại học

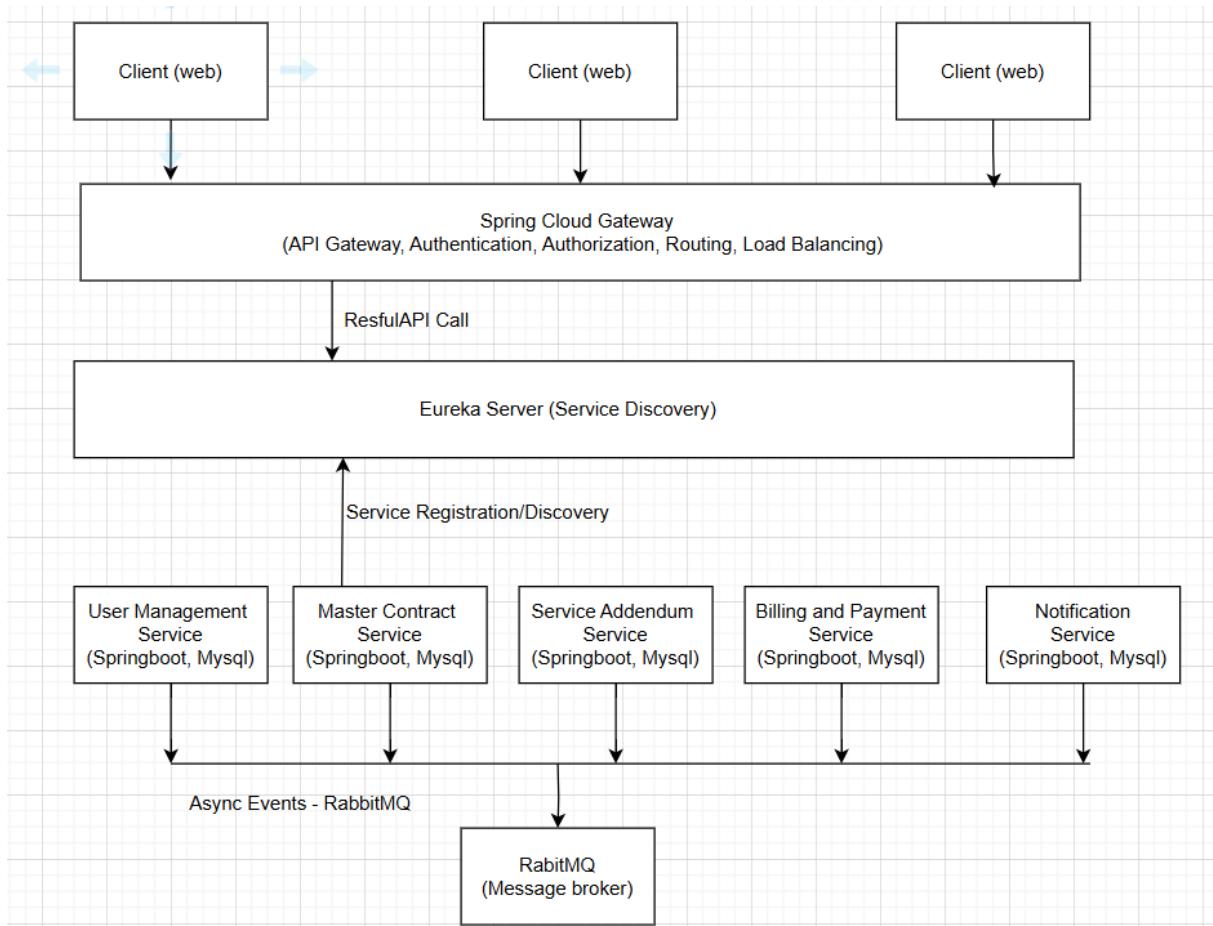


Figure 84 kiến trúc hệ thống tổng thể

## 5.1 Sơ đồ thực thể (ERD)

### Dịch vụ Quản lý Người dùng (User Management Service)

**Mô tả:** Dịch vụ này chịu trách nhiệm quản lý thông tin đăng nhập, xác thực người dùng, thông tin cá nhân của cư dân và nhân viên Ban Quản lý (BQL), cũng như thông tin về các căn hộ.

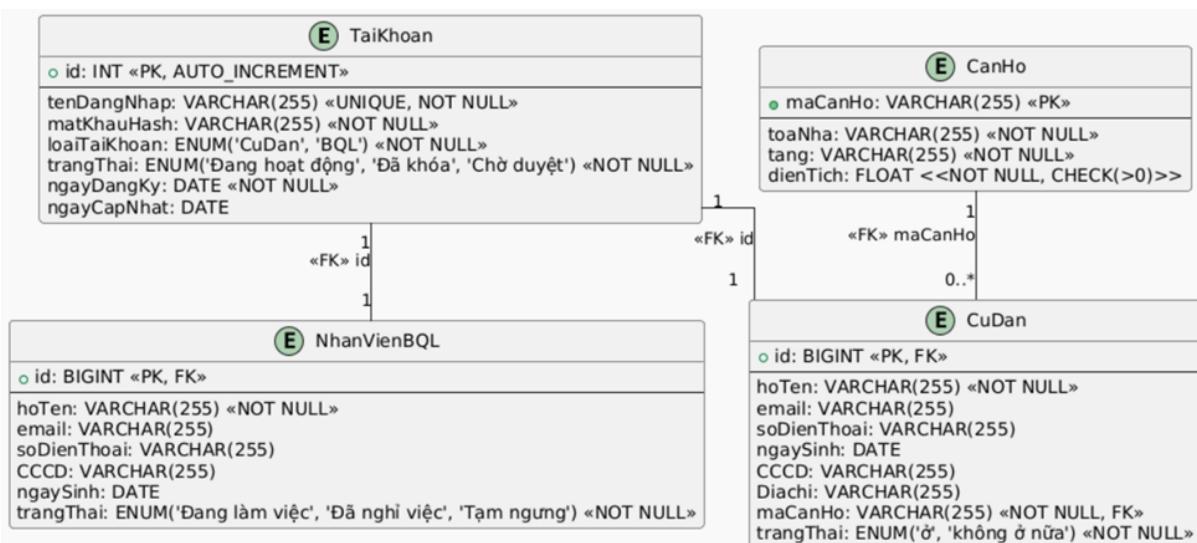


Figure 85 Dịch vụ Quản lý Người dùng

### Dịch vụ Quản lý Phụ lục Dịch vụ (Service Addendum Service)

# Báo cáo Thực tập Tốt nghiệp Đại học

**Mô tả:** Dịch vụ này quản lý danh mục các loại dịch vụ, các cấu hình giá phí theo bậc thang, các gói cước dịch vụ và các phụ lục dịch vụ cụ thể được áp dụng trong hợp đồng.

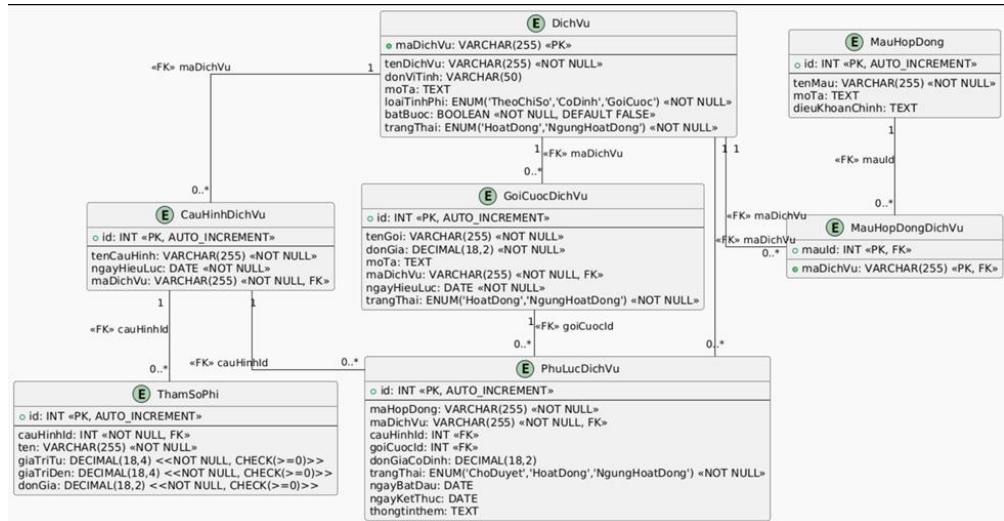


Figure 86 Dịch vụ Quản lý Phụ lục Dịch vụ

## Dịch vụ Quản lý Hợp đồng (Contract Service)

**Mô tả:** Dịch vụ này quản lý toàn bộ vòng đời của hợp đồng, từ tạo mới, xem chi tiết, đến xử lý các yêu cầu thay đổi và hủy bỏ hợp đồng.



Figure 87 Dịch vụ Quản lý Hợp đồng

## Dịch vụ Lập hóa đơn & Thanh toán (Billing & Payment Service)

**Mô tả:** Dịch vụ này chịu trách nhiệm tạo hóa đơn định kỳ, tính toán chi phí dịch vụ, ghi nhận chỉ số tiêu thụ và quản lý các giao dịch thanh toán.

# Báo cáo Thực tập Tốt nghiệp Đại học

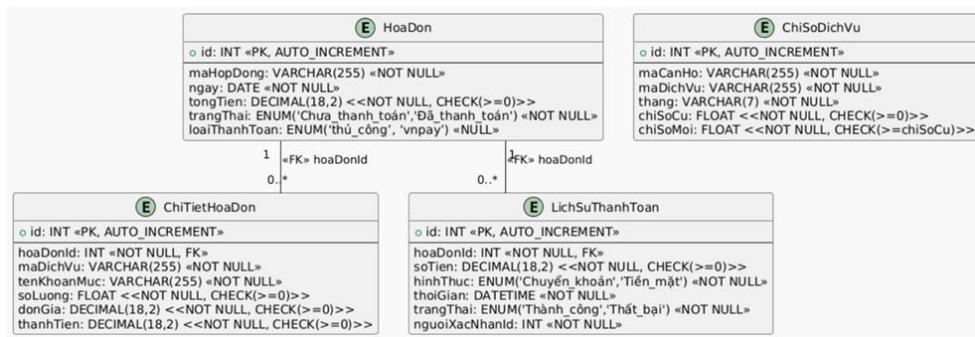


Figure 88 Dịch vụ Lập hóa đơn & Thanh toán

## Dịch vụ Thông báo (Notification Service)

**Mô tả:** Dịch vụ này chuyên biệt hóa việc gửi và quản lý các thông báo đến người dùng.

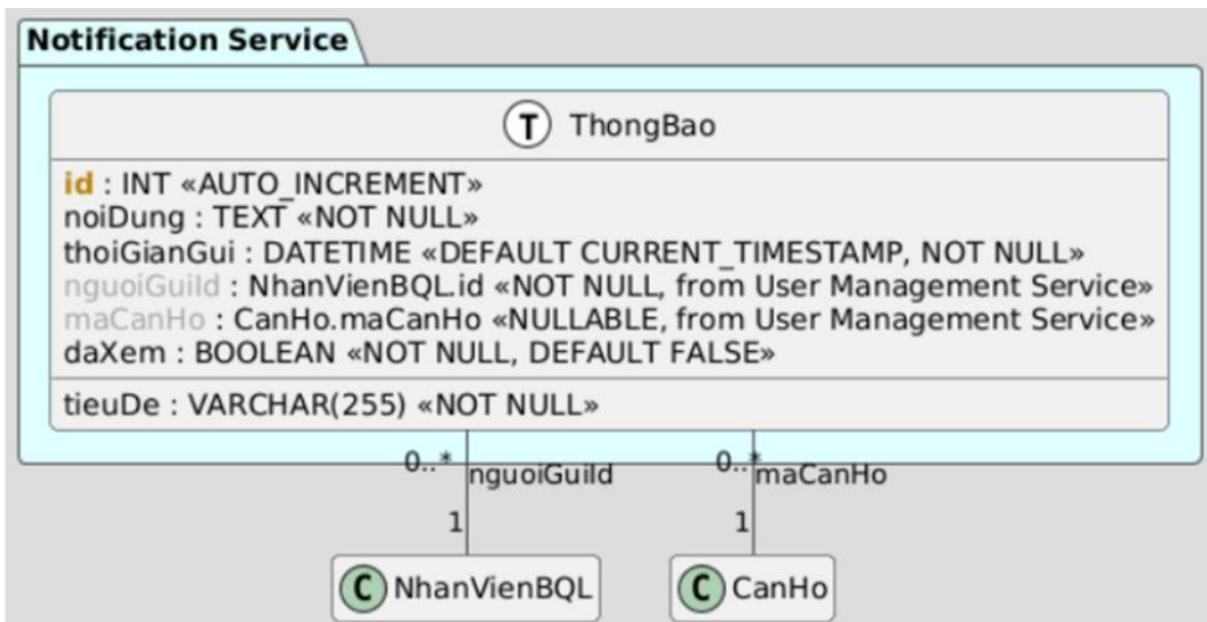


Figure 89 Dịch vụ Thông báo

## 5.2 Sơ đồ chuyển đổi trạng thái

### . Dịch vụ Quản lý Người dùng (User Management Service)

Bảng TaiKhoan

Sơ đồ này mô tả vòng đời của một tài khoản người dùng, từ khi đăng ký đến khi bị khóa hoặc kích hoạt lại.

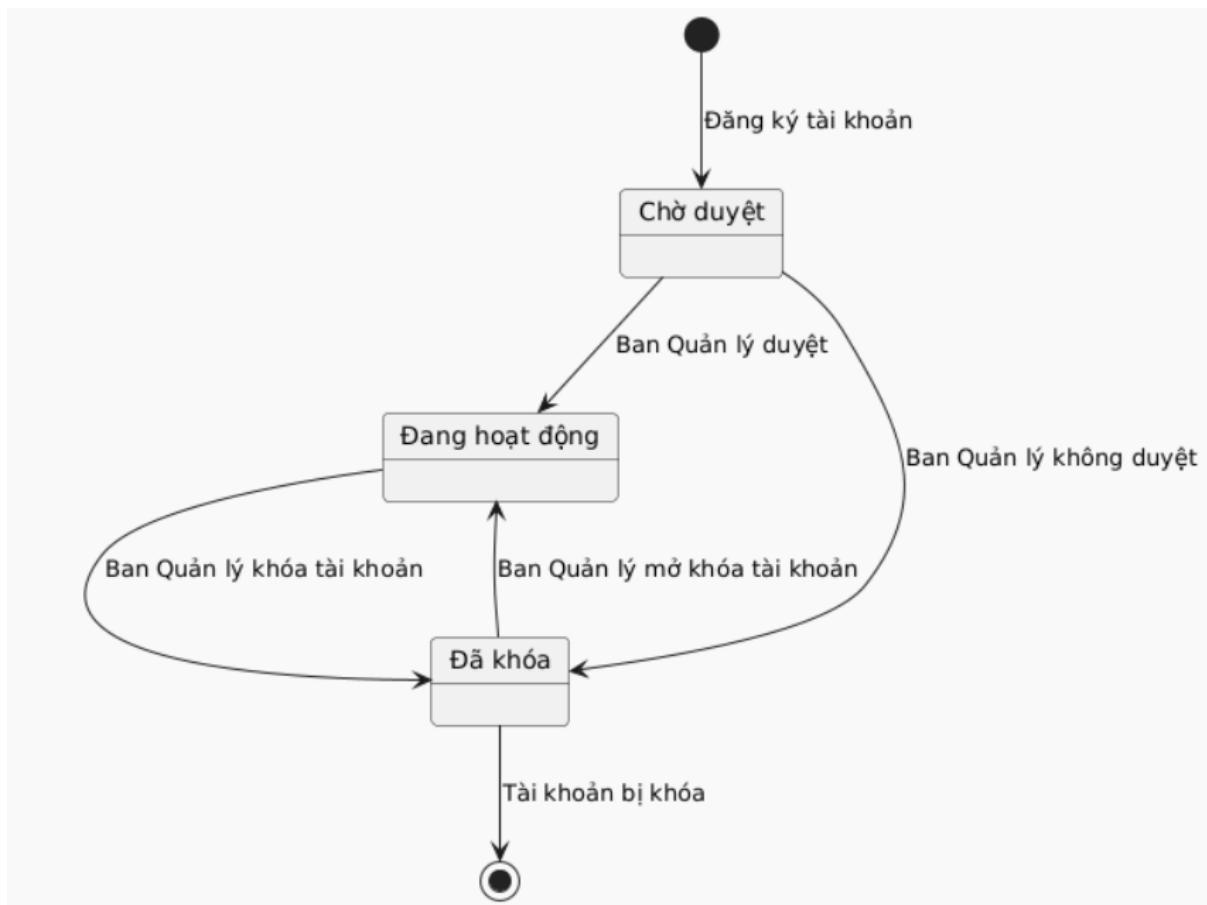


Figure 90 Sơ đồ chuyển đổi trạng thái tài khoản

### Bảng mô tả chuyển đổi trạng thái

Table 36 Chuyển đổi trạng thái tài khoản

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Đăng ký tài khoản	Chờ duyệt
Chờ duyệt	Ban Quản lý duyệt	Đang hoạt động
Chờ duyệt	Ban Quản lý không duyệt	Đã khóa
Đang hoạt động	Ban Quản lý khóa tài khoản	Đã khóa
Đã khóa	Ban Quản lý mở khóa tài khoản	Đang hoạt động
Đã khóa	Xóa tài khoản	[Kết thúc]

Sơ đồ này thể hiện các trạng thái làm việc của một nhân viên Ban Quản lý.

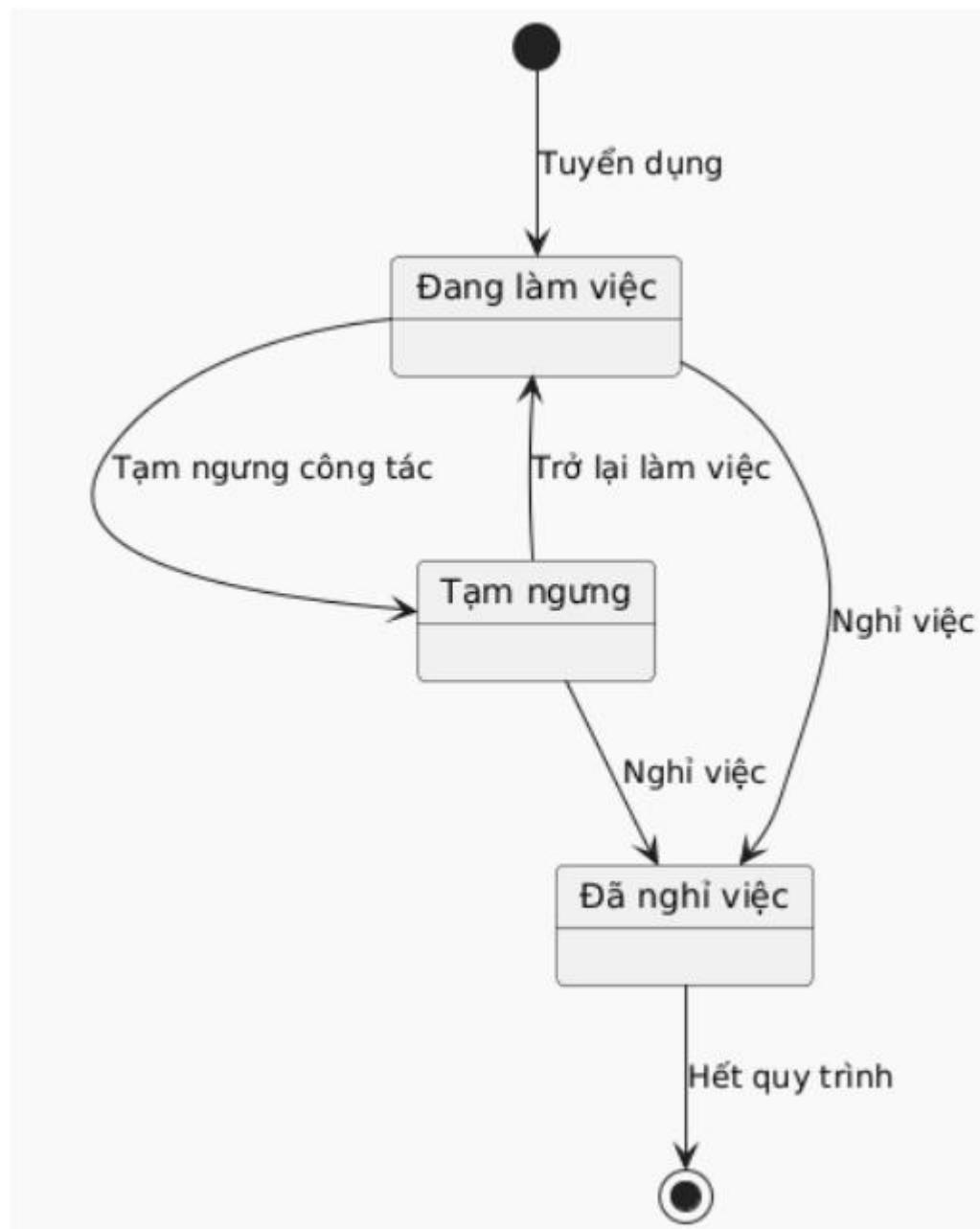


Figure 91 chuyển đổi trạng thái BQL

### Bảng mô tả chuyển đổi trạng thái

Table 37: chuyển đổi trạng thái BQL

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Tuyển dụng	Đang làm việc

## Báo cáo Thực tập Tốt nghiệp Đại học

Đang làm việc	Tạm ngưng công tác	Tạm ngưng
Tạm ngưng	Trở lại làm việc	Đang làm việc
Đang làm việc	Nghỉ việc	Đã nghỉ việc
Tạm ngưng	Nghỉ việc	Đã nghỉ việc
Đã nghỉ việc	Hết quy trình	[Kết thúc]

Bảng CuDan

Sơ đồ này mô tả các trạng thái cư trú của một cư dân.

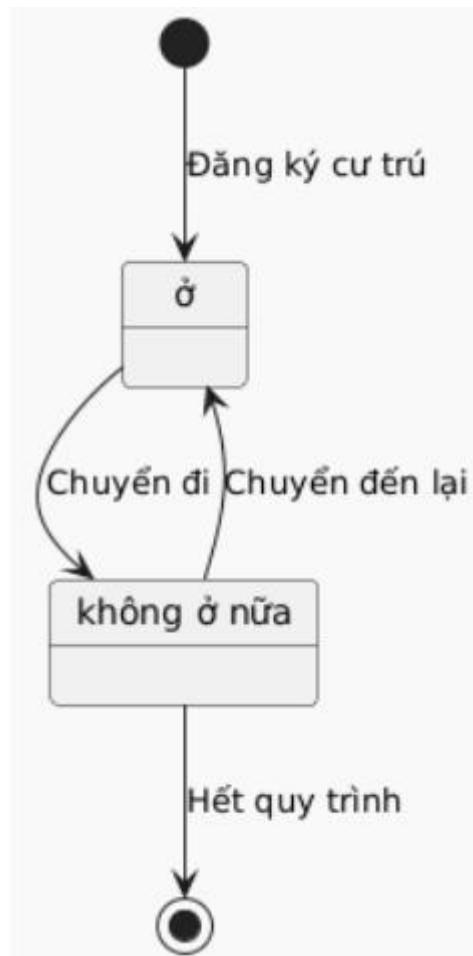


Figure 92 chuyển đổi trạng thái cư dân

Bảng mô tả chuyển đổi trạng thái

# Báo cáo Thực tập Tốt nghiệp Đại học

Table 38: chuyen doi trang thai cu dan

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Đăng ký cư trú	ở
ở	Chuyển đi	không ở nữa
không ở nữa	Chuyển đến lại	ở
không ở nữa	Hết quy trình	[Kết thúc]

## 2. Dịch vụ Quản lý Hợp đồng (Contract Service)

Bảng HopDong

Sơ đồ này minh họa các trạng thái của một hợp đồng dịch vụ.

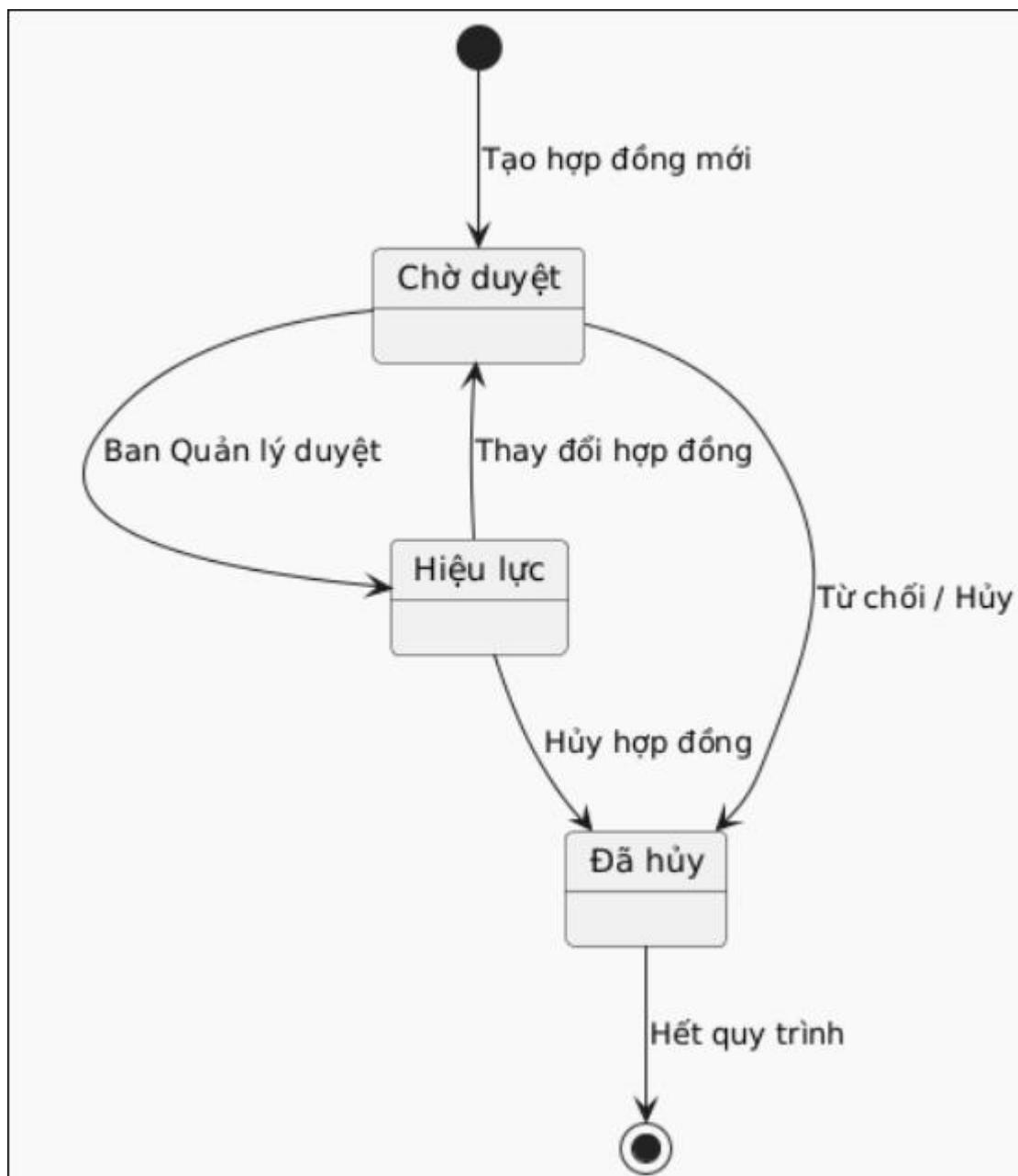


Figure 93 chuyển đổi trạng thái hợp đồng

### Bảng mô tả chuyển đổi trạng thái

Table 39 chuyển đổi trạng thái hợp đồng

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Tạo hợp đồng mới	Chờ duyệt
Chờ duyệt	Ban Quản lý duyệt	Hiệu lực

## Báo cáo Thực tập Tốt nghiệp Đại học

Chờ duyệt	Từ chối / Hủy	Đã hủy
Hiệu lực	Hủy hợp đồng	Đã hủy
Đã hủy	Hết quy trình	[Kết thúc]

Bảng YeuCauThayDoiHopDong

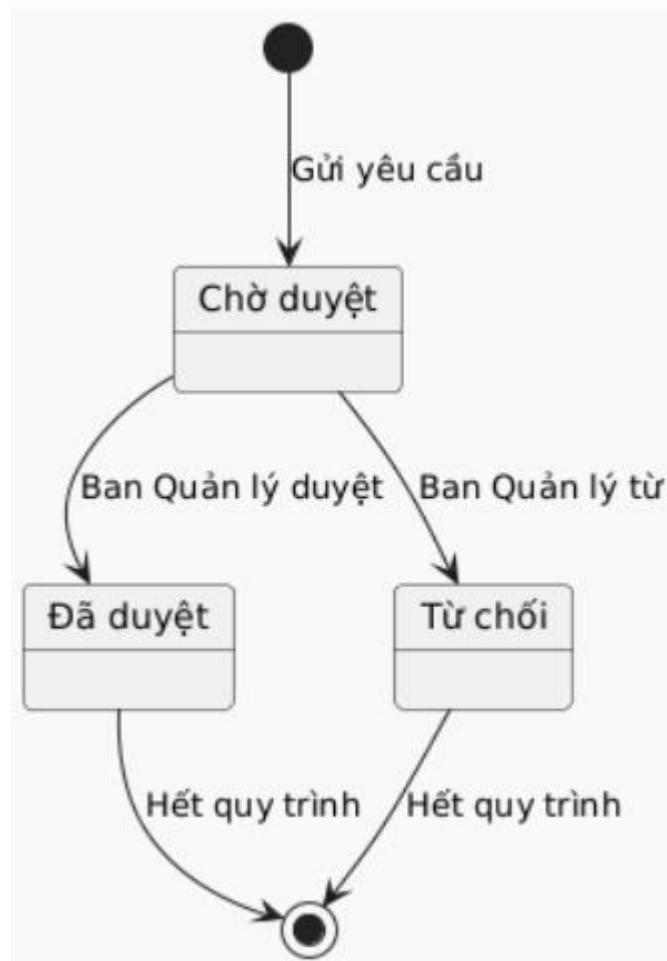


Figure 94 chuyển đổi trạng thái yêu cầu thay đổi

### Bảng mô tả chuyển đổi trạng thái

Table 40 : chuyển đổi trạng thái yêu cầu thay đổi

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Gửi yêu cầu	Chờ duyệt

# Báo cáo Thực tập Tốt nghiệp Đại học

Chờ duyệt	Ban Quản lý duyệt	Đã duyệt
Chờ duyệt	Ban Quản lý từ chối	Từ chối
Đã duyệt	Hết quy trình	[Kết thúc]
Từ chối	Hết quy trình	[Kết thúc]

## 3. Dịch vụ Quản lý Phụ lục Dịch vụ (Service Addendum Service)

### Bảng DichVu

Sơ đồ này thể hiện trạng thái hoạt động của một dịch vụ.

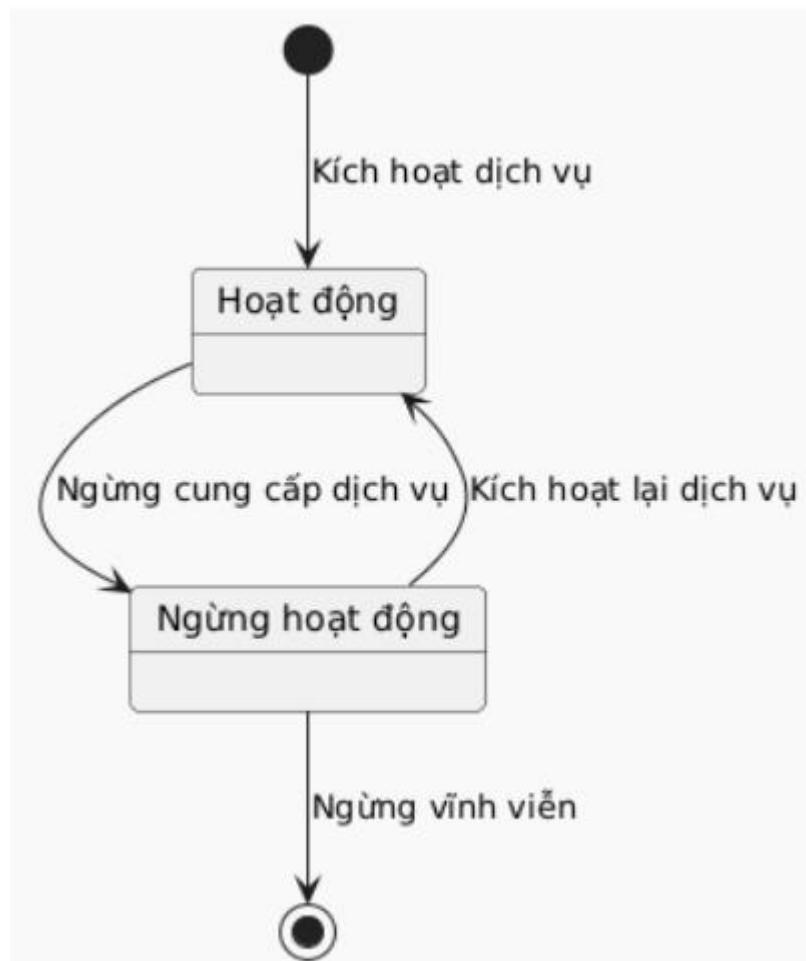


Figure 95 chuyển đổi trạng thái dịch vụ

### Bảng mô tả chuyển đổi trạng thái

# Báo cáo Thực tập Tốt nghiệp Đại học

Table 41 chuyển đổi trạng thái dịch vụ

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Kích hoạt dịch vụ	Hoạt động
Hoạt động	Ngừng cung cấp dịch vụ	Ngừng hoạt động
Ngừng hoạt động	Kích hoạt lại dịch vụ	Hoạt động
Ngừng hoạt động	Ngừng vĩnh viễn	[Kết thúc]

Bảng GoiCuocDichVu

Sơ đồ này mô tả trạng thái của một gói cước dịch vụ cụ thể.

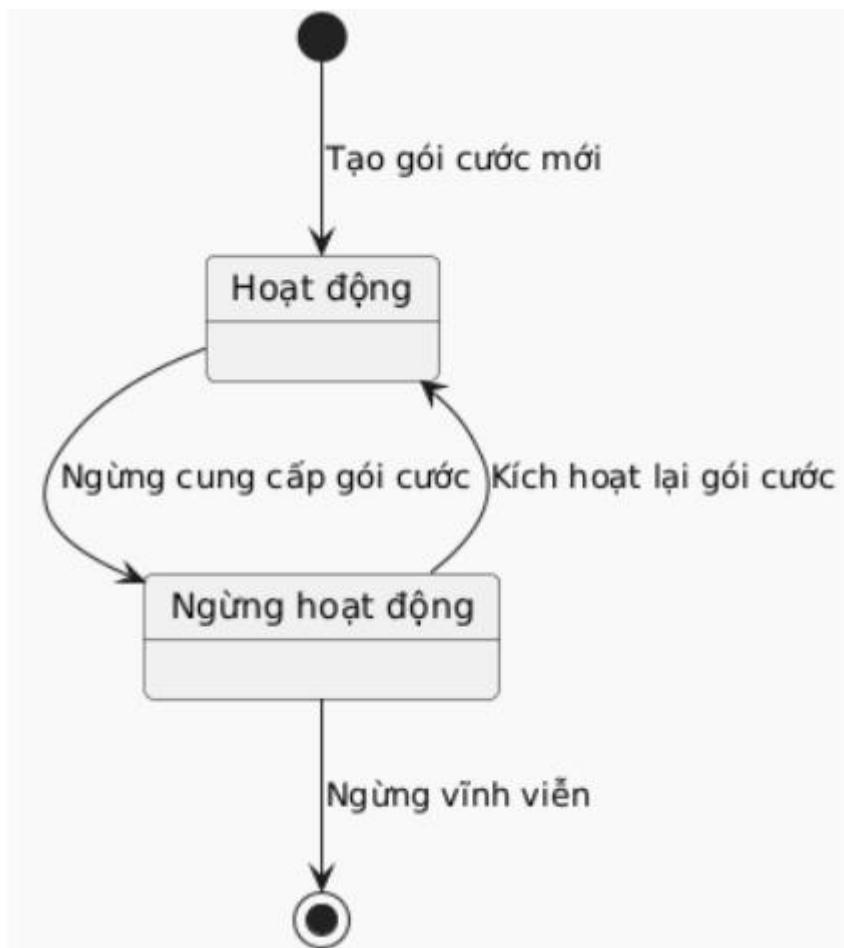


Figure 96 chuyển đổi trạng thái gói cước

Bảng mô tả chuyển đổi trạng thái

# Báo cáo Thực tập Tốt nghiệp Đại học

Table 42 chuyển đổi trạng thái gói cước

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Tạo gói cước mới	Hoạt động
Hoạt động	Ngừng cung cấp gói cước	Ngừng hoạt động
Ngừng hoạt động	Kích hoạt lại gói cước	Hoạt động
Ngừng hoạt động	Ngừng vĩnh viễn	[Kết thúc]

Bảng PhuLucDichVu

Sơ đồ này thể hiện trạng thái của một dịch vụ trong phụ lục hợp đồng của một cư dân.

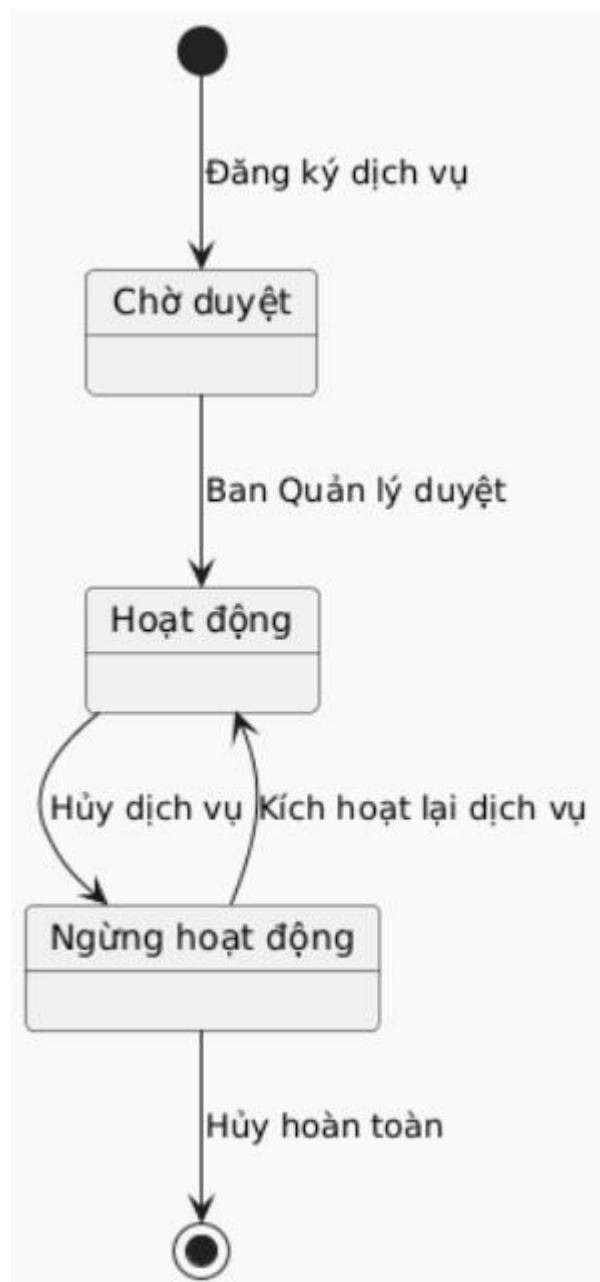


Figure 97 chuyển đổi trạng thái phụ lục dịch vụ

#### Bảng mô tả chuyển đổi trạng thái

Table 43 chuyển đổi trạng thái phụ lục dịch vụ

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Đăng ký dịch vụ	Chờ duyệt
Chờ duyệt	Ban Quản lý duyệt	Hoạt động
Hoạt động	Hủy dịch vụ	Ngừng hoạt động

# Báo cáo Thực tập Tốt nghiệp Đại học

Ngừng hoạt động	Kích hoạt lại dịch vụ	Hoạt động
Ngừng hoạt động	Hủy hoàn toàn	[Kết thúc]

## 4. Dịch vụ Lập hóa đơn & Thanh toán (Billing & Payment Service)

### Bảng HoaDon

Sơ đồ này mô tả trạng thái thanh toán của một hóa đơn.

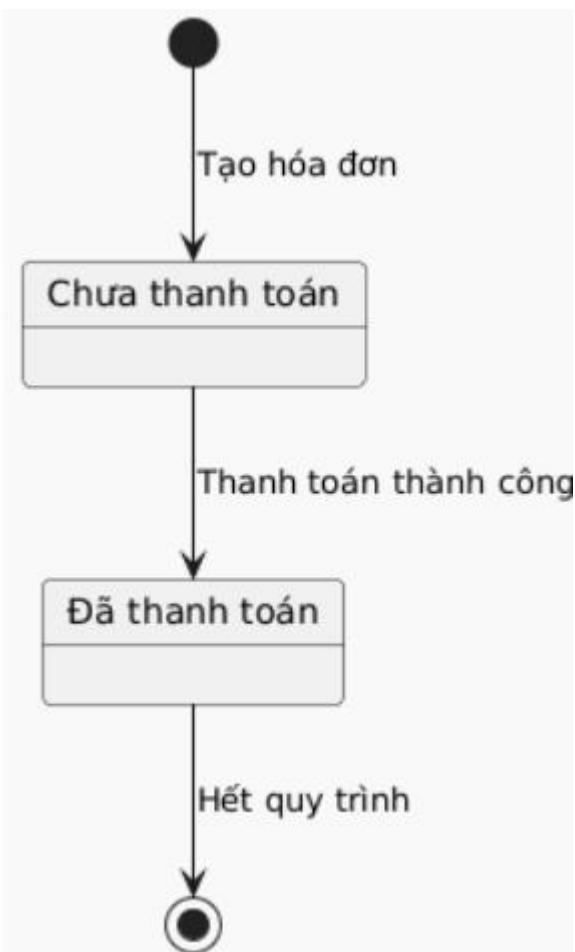


Figure 98 chuyển đổi trạng thái hóa đơn

### Bảng mô tả chuyển đổi trạng thái

Table 44 chuyển đổi trạng thái hóa đơn

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Tạo hóa đơn	Chưa thanh toán
Chưa thanh toán	Thanh toán thành công	Đã thanh toán

# Báo cáo Thực tập Tốt nghiệp Đại học

Đã thanh toán	Hết quy trình	[Kết thúc]
---------------	---------------	------------

## Bảng Lịch sử thanh toán

Sơ đồ này thể hiện trạng thái của một giao dịch thanh toán.

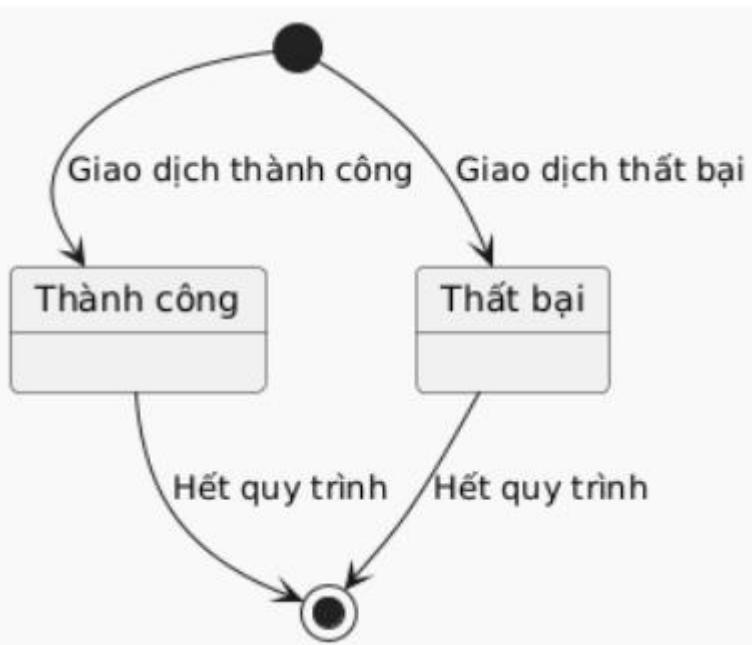


Figure 99 chuyển đổi trạng thái lịch sử thanh toán

## Bảng mô tả chuyển đổi trạng thái

Table 45 chuyển đổi trạng thái lịch sử thanh toán

Trạng thái	Sự kiện	Trạng thái mới
[Bắt đầu]	Giao dịch thành công	Thành công
Thành công	Hết quy trình	[Kết thúc]
[Bắt đầu]	Giao dịch thất bại	Thất bại
Thất bại	Hết quy trình	[Kết thúc]

## 5.3 Thiết kế cơ sở dữ liệu

### Dịch vụ Quản lý Người dùng (User Management Service)

Bảng: TaiKhoan

Báo cáo Thực tập Tốt nghiệp Đại học

Mục đích: Lưu trữ thông tin đăng nhập và xác thực của người dùng.

Table 46 Bảng: TaiKhoan

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	BIGINT	PK, AUTO_INCREMENT	Mã định danh duy nhất của tài khoản.
tenDangNhap	VARCHAR(255)	UNIQUE, NOT NULL	Tên đăng nhập của người dùng.
matKhauHash	VARCHAR(255)	NOT NULL	Mật khẩu đã được mã hóa.
loaiTaiKhoan	ENUM('CuDan', 'BQL')	NOT NULL	Loại tài khoản người dùng (Cư dân hoặc Ban Quản lý).
trangThai	ENUM('Đang hoạt động', 'Đã khóa', 'Chờ duyệt')	NOT NULL	Trạng thái hoạt động của tài khoản.
ngayDangKy	DATETIME(6)	NOT NULL	Thời điểm tài khoản được đăng ký.
ngayCapNhat	DATE		Thời điểm cập nhật thông tin gần nhất.

Bảng: CanHo

Mục đích: Lưu trữ thông tin chi tiết về từng căn hộ trong tòa nhà.

Table 47 Bảng: CanHo

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
maCanHo	VARCHAR(255)	PK	Mã định danh duy nhất của căn hộ.

Báo cáo Thực tập Tốt nghiệp Đại học

toaNha	VARCHAR(255)	NOT NULL	Tên tòa nhà.
tang	VARCHAR(255)	NOT NULL	Số tầng của căn hộ.
dienTich	FLOAT	NOT NULL, CHECK(>0)	Diện tích của căn hộ (đơn vị m <sup>2</sup> ).

Bảng: CuDan

Mục đích: Lưu trữ thông tin cá nhân chi tiết của cư dân, liên kết với một tài khoản và một căn hộ.

Table 48 Bảng: CuDan

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	BIGINT	PK, FK (TaiKhoan.id)	Mã định danh của cư dân, là khóa ngoại từ bảng TaiKhoan.
hoTen	VARCHAR(255)	NOT NULL	Họ và tên của cư dân.
email	VARCHAR(255)		Địa chỉ email của cư dân.
soDienThoai	VARCHAR(255)		Số điện thoại liên hệ.
ngaySinh	DATE		Ngày sinh của cư dân.
CCCD	VARCHAR(255)		Số căn cước công dân.
Diachi	VARCHAR(255)		Địa chỉ thường trú.

Báo cáo Thực tập Tốt nghiệp Đại học

maCanHo	VARCHAR(255)	NOT NULL, FK (CanHo.maCanHo)	Mã căn hộ mà cư dân đang ở.
trangThai	ENUM('ở', 'không ở nữa')	NOT NULL	Trạng thái cư trú của cư dân.

Bảng: NhanVienBQL

Mục đích: Lưu trữ thông tin cá nhân chi tiết của nhân viên Ban Quản lý, liên kết với một tài khoản.

Table 49 Bảng: NhanVienBQL

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	BIGINT	PK, FK (TaiKhoan.id)	Mã định danh của nhân viên, là khóa ngoại từ bảng TaiKhoan.
hoTen	VARCHAR(255)	NOT NULL	Họ và tên của nhân viên.
email	VARCHAR(255)		Địa chỉ email của nhân viên.
soDienThoai	VARCHAR(255)		Số điện thoại liên hệ.
CCCD	VARCHAR(255)		Số căn cước công dân.
ngaySinh	DATE		Ngày sinh của nhân viên.
trangThai	ENUM('Đang làm việc', 'Đã nghỉ việc', 'Tạm ngưng')	NOT NULL	Trạng thái làm việc của nhân viên.

**Dịch vụ Quản lý Hợp đồng (Contract Service)**

Bảng: HopDong

Mục đích: Lưu trữ thông tin tổng quan về hợp đồng, bao gồm mã cư dân, người ký, căn hộ, và trạng thái.

**Table 50** Bảng: HopDong

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
maHopDong	VARCHAR(255)	PK	Mã định danh duy nhất của hợp đồng.
maCuDan	INT	NOT NULL, FK	Mã của cư dân đã ký hợp đồng.
maNguoiKyBQL	INT		Mã của nhân viên BQL đã ký duyệt hợp đồng.
maCanHo	VARCHAR(255)		Mã căn hộ liên quan đến hợp đồng.
mauHopDongId	INT	NOT NULL, FK (MauHopDong.id)	Mã của mẫu hợp đồng được sử dụng.
ngayKy	DATE	NOT NULL	Ngày ký hợp đồng.
trangThai	ENUM('Chờ_duyệt','Hiệu_lực','Đã_hủy')		Trạng thái hiện tại của hợp đồng.
ngayKetThuc	DATE		Ngày hợp đồng hết hiệu lực.

Bảng: YeuCauThayDoiHopDong

Mục đích: Lưu trữ các yêu cầu thay đổi hợp đồng, cho phép BQL và cư dân theo dõi tiến trình của các yêu cầu này.

Báo cáo Thực tập Tốt nghiệp Đại học

Table 51 Bảng: YeuCauThayDoiHopDong

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của yêu cầu.
maHopDong	VARCHAR(255)	NOT NULL, FK (HopDong.maHopDong)	Mã hợp đồng bị ảnh hưởng bởi yêu cầu.
nguoiGuId	INT	NOT NULL	Mã người dùng đã gửi yêu cầu.
loaiThayDoi	ENUM('Nâng_cấp_gói_cước','Hủy_dịch_vụ','Đăng_ký_mới')	NOT NULL	Loại thay đổi được yêu cầu.
trangThai	ENUM('Chờ_duyệt','Đã_duyệt','Tù_chờ')	NOT NULL	Trạng thái xử lý của yêu cầu.
ngayTao	DATETIME	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Thời điểm yêu cầu được tạo.
noiDungChiTiet	TEXT		Nội dung chi tiết của yêu cầu.

**Dịch vụ Quản lý Phụ lục Dịch vụ (Service Addendum Service)**

**Bảng: MauHopDong**

Mục đích: Lưu trữ các mẫu hợp đồng chuẩn.

Table 52 Bảng: MauHopDong

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa

Báo cáo Thực tập Tốt nghiệp Đại học

id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của mẫu hợp đồng.
tenMau	VARCHAR(255)	NOT NULL	Tên của mẫu hợp đồng.
moTa	TEXT		Mô tả chi tiết về mẫu hợp đồng.
dieuKhoanChinh	TEXT		Các điều khoản chính trong mẫu hợp đồng.

**Bảng: MauHopDongDichVu**

Mục đích: Liên kết mẫu hợp đồng với các dịch vụ mặc định.

Table 53 Bảng: MauHopDongDichVu

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
maulId	INT	PK, FK (MauHopDong.id)	Mã mẫu hợp đồng.
maDichVu	VARCHAR(255)	PK, FK (DichVu.maDichVu )	Mã dịch vụ mặc định trong mẫu hợp đồng.

Bảng: DichVu

Mục đích: Danh mục các loại dịch vụ được cung cấp.

Table 54 Bảng: DichVu

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa

Báo cáo Thực tập Tốt nghiệp Đại học

maDichVu	VARCHAR(255)	PK	Mã định danh duy nhất của dịch vụ.
tenDichVu	VARCHAR(255)	NOT NULL	Tên của dịch vụ.
donViTinh	VARCHAR(50)		Đơn vị tính của dịch vụ (ví dụ: kWh, m <sup>3</sup> ).
moTa	TEXT		Mô tả chi tiết về dịch vụ.
loaiTinhPhi	ENUM('TheoChiSo', 'CoDinh', 'GoiCuoc')	NOT NULL	Loại tính phí của dịch vụ.
batBuoc	TINYINT(1)	NOT NULL, DEFAULT '0'	Dịch vụ có bắt buộc không.
trangThai	ENUM('HoatDong', 'NgungHoatDong')	NOT NULL	Trạng thái hoạt động của dịch vụ.

Bảng: CauHinhDichVu

Mục đích: Lưu trữ các phiên bản cấu hình giá phí theo bậc thang.

Table 55 Bảng: CauHinhDichVu

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của cấu hình.
tenCauHinh	VARCHAR(255)	NOT NULL	Tên của cấu hình.
ngayHieuLuc	DATE	NOT NULL	Ngày cấu hình bắt đầu có hiệu lực.

## Báo cáo Thực tập Tốt nghiệp Đại học

maDichVu	VARCHAR(255)	NOT NULL, FK (DichVu.maDichVu )	Mã dịch vụ mà cấu hình này áp dụng.
----------	--------------	---------------------------------------	--

Bảng: ThamSoPhi

Mục đích: Chi tiết các bậc thang giá phí trong một cấu hình dịch vụ.

Table 56 Bảng: ThamSoPhi

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của tham số phi.
cauHinhId	INT	NOT NULL, FK (CauHinhDichVu.id )	Mã cấu hình dịch vụ mà tham số này thuộc về.
ten	VARCHAR(255)	NOT NULL	Tên của bậc thang phi.
giaTriTu	DECIMAL(18,4)	NOT NULL, CHECK(>=0)	Giá trị bắt đầu của bậc thang phí.
giaTriDen	DECIMAL(18,4)	NOT NULL, CHECK(>=0)	Giá trị kết thúc của bậc thang phí.
donGia	DECIMAL(18,2)	NOT NULL, CHECK(>=0)	Đơn giá áp dụng cho bậc thang này.

Bảng: GoiCuocDichVu

Mục đích: Danh mục các gói cước dịch vụ.

Table 57 Bảng: GoiCuocDichVu

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
------------	--------------	-----------	---------

Báo cáo Thực tập Tốt nghiệp Đại học

id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của gói cước.
tenGoi	VARCHAR(255)	NOT NULL	Tên của gói cước.
donGia	DECIMAL(18,2)	NOT NULL, CHECK(>=0)	Đơn giá của gói cước.
moTa	TEXT		Mô tả chi tiết về gói cước.
maDichVu	VARCHAR(255)	NOT NULL, FK (DichVu.maDichVu )	Mã dịch vụ mà gói cước này thuộc về.
ngayHieuLuc	DATE	NOT NULL	Ngày gói cước bắt đầu có hiệu lực.
trangThai	ENUM('HoatDong','NgungHoatDong')	NOT NULL	Trạng thái hoạt động của gói cước.

Bảng: PhuLucDichVu

Mục đích: Lưu trữ các dịch vụ được đăng ký trong hợp đồng.

Table 58 Bảng: PhuLucDichVu

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của phụ lục dịch vụ.
maHopDong	VARCHAR(255)	NOT NULL	Mã hợp đồng mà phụ lục này thuộc về.

Báo cáo Thực tập Tốt nghiệp Đại học

maDichVu	VARCHAR(255)	NOT NULL, FK (DichVu.maDichVu )	Mã dịch vụ được đăng ký.
cauHinhId	INT	FK (CauHinhDichVu.id )	Mã cấu hình dịch vụ áp dụng.
goiCuocId	INT	FK (GoiCuocDichVu.id )	Mã gói cước dịch vụ áp dụng.
donGiaCoDinh	DECIMAL(18,2)	CHECK(>=0)	Đơn giá cố định (nếu có).
trangThai	ENUM('ChoDuyet',' HoatDong','NgungH oatDong')	NOT NULL	Trạng thái của dịch vụ trong phụ lục.
ngayBatDau	DATE		Ngày bắt đầu sử dụng dịch vụ.
ngayKetThuc	DATE		Ngày kết thúc sử dụng dịch vụ.
thongtinthem	TEXT		Thông tin bổ sung về dịch vụ.

**Dịch vụ Lập hóa đơn & Thanh toán (Billing & Payment Service)**

Bảng: HoaDon

Mục đích: Lưu trữ thông tin tổng quan về hóa đơn dịch vụ hàng tháng.

Table 59 Bảng: HoaDon

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của hóa đơn.

Báo cáo Thực tập Tốt nghiệp Đại học

maHopDong	VARCHAR(255)	NOT NULL	Mã hợp đồng liên quan đến hóa đơn.
ngay	DATE	NOT NULL	Ngày tạo hóa đơn.
tongTien	DECIMAL(18,2)	NOT NULL, CHECK(>=0)	Tổng số tiền cần thanh toán trên hóa đơn.
trangThai	ENUM('Chưa thanh toán', 'Đã thanh toán')	NOT NULL	Trạng thái thanh toán của hóa đơn.
loaiThanhToan	ENUM('thủ công', 'vnpay')	NULL	Lưu lại cách thanh toán của hóa đơn

Bảng: ChiTietHoaDon

Mục đích: Lưu trữ chi tiết từng khoản phí trong một hóa đơn.

Table 60 Bảng: ChiTietHoaDon

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của chi tiết hóa đơn.
hoaDonId	INT	NOT NULL, FK (HoaDon.id)	Mã hóa đơn cha.
maDichVu	VARCHAR(255)	NOT NULL	Mã dịch vụ.
tenKhoanMuc	VARCHAR(255)	NOT NULL	Tên của khoản mục phí.
soLuong	FLOAT	NOT NULL, CHECK(>=0)	Số lượng sử dụng dịch vụ.

Báo cáo Thực tập Tốt nghiệp Đại học

donGia	DECIMAL(18,2)	NOT NULL, CHECK( $>=0$ )	Đơn giá của khoản mục.
thanhTien	DECIMAL(18,2)	NOT NULL, CHECK( $>=0$ )	Tổng tiền cho khoản mục này.

Bảng: ChiSoDichVu

Mục đích: Ghi nhận chỉ số tiêu thụ hàng tháng của các dịch vụ.

Table 61 Bảng: ChiSoDichVu

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của chỉ số dịch vụ.
maCanHo	VARCHAR(255)	NOT NULL	Mã căn hộ.
maDichVu	VARCHAR(255)	NOT NULL	Mã dịch vụ.
thang	VARCHAR(7)	NOT NULL	Tháng ghi nhận chỉ số (định dạng YYYY-MM).
chiSoCu	FLOAT	NOT NULL, CHECK( $>=0$ )	Chỉ số cũ của dịch vụ.
chiSoMoi	FLOAT	NOT NULL, CHECK( $>=chiSoCu$ )	Chỉ số mới của dịch vụ.

Bảng: LichSuThanhToan

Mục đích: Ghi nhận lịch sử các giao dịch thanh toán.

Báo cáo Thực tập Tốt nghiệp Đại học

Table 62 Bảng: LichSuThanhToan

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của giao dịch.
hoaDonId	INT	NOT NULL, FK (HoaDon.id)	Mã hóa đơn đã được thanh toán.
soTien	DECIMAL(18,2)	NOT NULL, CHECK(>=0)	Số tiền đã thanh toán.
hinhThuc	ENUM('Chuyển khoản','Tiền mặt')	NOT NULL	Hình thức thanh toán.
thoiGian	DATETIME	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Thời điểm giao dịch.
trangThai	ENUM('Thành công','Thất bại')	NOT NULL	Trạng thái của giao dịch.
nguoiXacNhanId	INT	NOT NULL	Mã nhân viên BQL đã xác nhận giao dịch.

**Dịch vụ Thông báo (Notification Service)**

Bảng: ThongBao

Mục đích: Lưu trữ thông tin chi tiết về các thông báo đã được gửi.

Table 63 Bảng: ThongBao

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Ý nghĩa
id	INT	PK, AUTO_INCREMENT	Mã định danh duy nhất của thông báo.

Báo cáo Thực tập Tốt nghiệp Đại học

tieuDe	VARCHAR(255)	NOT NULL	Tiêu đề của thông báo.
noiDung	TEXT	NOT NULL	Nội dung chi tiết của thông báo.
thoiGianGui	DATETIME	NOT NULL, DEFAULT CURRENT_TIMESTAMP	Thời điểm thông báo được gửi.
nguoiguiId	INT	NOT NULL	Mã người dùng đã gửi thông báo.
maCanHo	VARCHAR(255)		Mã căn hộ được nhận thông báo (nếu là thông báo riêng).
daXem	TINYINT(1)	NOT NULL, DEFAULT '0'	Trạng thái đã xem của thông báo.

## Chương 4: Cài đặt và triển khai hệ thống

### 1 Mô tả chi tiết các Microservices

#### 1.1. Tổng quan về kiến trúc Microservices

Kiến trúc Microservices (Microservices Architecture) là một kiểu kiến trúc phần mềm trong đó ứng dụng được chia thành nhiều vi dịch vụ nhỏ, mỗi vi dịch vụ đảm nhận một chức năng cụ thể và có thể hoạt động độc lập với nhau. Các vi dịch vụ này thường được triển khai và phát triển riêng biệt, có thể sử dụng các công nghệ khác nhau, và giao tiếp với nhau thông qua các giao thức mạng (thường là HTTP REST hoặc message brokers như RabbitMQ, Kafka). Kiến trúc này giúp tăng tính linh hoạt, dễ dàng mở rộng và bảo trì hệ thống.

Các lợi ích chính của kiến trúc Microservices bao gồm:

- **Scalability (Khả năng mở rộng):** Mỗi vi dịch vụ có thể được mở rộng độc lập.
- **Flexibility (Linh hoạt):** Các nhóm có thể phát triển và triển khai các vi dịch vụ khác nhau một cách độc lập.
- **Fault isolation (Cách ly lỗi):** Lỗi trong một vi dịch vụ không ảnh hưởng đến các vi dịch vụ khác.
- **Technology diversity (Đa dạng công nghệ):** Các vi dịch vụ có thể sử dụng các công nghệ khác nhau (Spring Boot, Node.js, .NET, v.v.).

#### 1.2. Các thành phần trong hệ thống Microservices

Hệ thống Microservices mà bạn xây dựng sẽ bao gồm các vi dịch vụ khác nhau cho các chức năng riêng biệt của hệ thống, bao gồm:

- **Vi dịch vụ quản lý người dùng:** Cấp, duyệt, hủy tài khoản người dùng, phân quyền.
- **Vi dịch vụ quản lý phụ lục dịch vụ:** Quản lý các dịch vụ, cấu hình và gói cước bao gồm tạo, cập nhật, xóa. Quản lý các phụ lục dịch vụ hợp đồng, bao gồm tạo, cập nhật, xóa.
- **Vi dịch vụ lập hóa đơn và thanh toán:** Quản lý hóa đơn, thanh toán hóa đơn bao gồm lập hóa đơn tự động hàng tháng, ghi nhận thanh toán hóa đơn (thủ công và online), thanh toán online bằng vnpay.
- **Vi dịch vụ quản lý hợp đồng:** Quản lý hợp đồng như tạo, cập nhật.
- **Vi dịch vụ thông báo:** Gửi cảnh báo và thông báo về các sự kiện quan trọng (hợp đồng hết hạn, thanh toán sắp đến hạn, v.v.).

Mỗi vi dịch vụ sẽ có cơ sở dữ liệu riêng biệt, đảm bảo tính độc lập và khả năng mở rộng của hệ thống.

#### 1.3. Spring Boot và RESTful APIs cho mỗi vi dịch vụ

- **Spring Boot:** Là một framework của Java giúp phát triển các ứng dụng microservices một cách nhanh chóng và dễ dàng. Spring Boot cung cấp nhiều tính năng hữu ích như:
  - **Auto Configuration:** Giúp tự động cấu hình các thành phần trong ứng dụng.
  - **Embedded Servers:** Cho phép triển khai ứng dụng mà không cần cài đặt server bên ngoài.
  - **Spring Data JPA:** Cung cấp tích hợp dễ dàng với cơ sở dữ liệu, giúp làm việc với MySQL và các cơ sở dữ liệu quan hệ khác.

- **Spring Security:** Cung cấp bảo mật cho các vi dịch vụ thông qua xác thực và phân quyền người dùng.
- **RESTful APIs:** Mỗi vi dịch vụ trong hệ thống sẽ triển khai các API theo chuẩn REST (Representational State Transfer). RESTful APIs là cách phổ biến để các vi dịch vụ giao tiếp với nhau. Các API này sẽ sử dụng các phương thức HTTP cơ bản như GET, POST, PUT, DELETE để thực hiện các thao tác trên dữ liệu.

#### 1.4. MySQL cho mỗi vi dịch vụ

Mỗi vi dịch vụ trong kiến trúc microservices sẽ có cơ sở dữ liệu riêng biệt, trong trường hợp này là MySQL, để đảm bảo tính độc lập và khả năng mở rộng của các vi dịch vụ. MySQL là một hệ quản trị cơ sở dữ liệu phổ biến, hỗ trợ SQL, giúp lưu trữ và quản lý dữ liệu một cách hiệu quả.

#### 1.5. Eureka cho Discovery Service

- **Eureka** là một công cụ của **Spring Cloud** giúp các vi dịch vụ tìm thấy nhau và giao tiếp mà không cần cấu hình tĩnh. Eureka Server sẽ là một dịch vụ cung cấp thông tin về các vi dịch vụ trong hệ thống (Service Discovery). Mỗi vi dịch vụ sẽ đăng ký với Eureka Server và có thể tìm thấy các vi dịch vụ khác thông qua Eureka Client.

Eureka giúp giải quyết các vấn đề liên quan đến cấu hình và đồng bộ hóa giữa các vi dịch vụ khi hệ thống có sự thay đổi (ví dụ: khi có vi dịch vụ mới được thêm vào).

#### 1.6. Spring Cloud Gateway cho điểm truy cập duy nhất

- **Spring Cloud Gateway** là một API Gateway cung cấp điểm truy cập duy nhất cho người dùng. Gateway này sẽ nhận tất cả các yêu cầu từ người dùng và chuyển tiếp chúng đến các vi dịch vụ tương ứng.
- Spring Cloud Gateway giúp thực hiện các chức năng như:
  - **Load Balancing:** Phân phối yêu cầu giữa các vi dịch vụ để đảm bảo tính sẵn sàng và hiệu suất của hệ thống.
  - **Authentication & Authorization:** Xác thực và phân quyền người dùng trước khi chuyển yêu cầu đến các vi dịch vụ.
  - **Routing:** Chuyển tiếp yêu cầu đến đúng vi dịch vụ dựa trên các tiêu chí cấu hình.

### 2. Triển khai Backend (Spring Boot, Cấu trúc Code, Các API Chính)

#### 2.1. Môi trường phát triển và công cụ sử dụng

- **Spring Boot:** Spring Boot là framework phát triển ứng dụng Java giúp đơn giản hóa quá trình cấu hình và triển khai. Spring Boot cung cấp một môi trường phát triển mạnh mẽ và dễ dàng để xây dựng các vi dịch vụ trong kiến trúc microservices.
- **VSCode:** VSCode là một công cụ phát triển nhẹ và linh hoạt, giúp lập trình viên dễ dàng triển khai và phát triển các ứng dụng backend bằng Java.
- **MySQL:** Là hệ quản trị cơ sở dữ liệu quan hệ được sử dụng để lưu trữ thông tin của người dùng, hợp đồng và các dịch vụ.
- **Postman:** Là công cụ kiểm thử API, cho phép bạn gửi yêu cầu HTTP tới backend và kiểm tra các phản hồi từ server.

#### 2.2. Cấu trúc dự án Spring Boot

# Báo cáo Thực tập Tốt nghiệp Đại học

Cấu trúc của một dự án Spring Boot cơ bản có thể chia thành các thư mục như sau:

```
src
  |
  +-- main
    |
    +-- java
      |
      +-- com
        |
        +-- QLDV
          |
          +-- controller  # Chứa các controller để xử lý các request
          |
          +-- service    # Chứa các dịch vụ logic của ứng dụng
          |
          +-- repository # Chứa các lớp repository để tương tác với cơ sở dữ liệu
          |
          +-- model      # Chứa các lớp mô hình dữ liệu (Entity)
          |
          +-- Application.java # Lớp khởi tạo Spring Boot Application
          |
          +-- resources
            |
            +-- application.properties # Cấu hình các thông số cho ứng dụng (CSDL, port, v
            |
            +-- static           # Chứa các tài nguyên tĩnh (css, js, v.v.)
            |
            +-- templates        # Chứa các file HTML nếu sử dụng Thymeleaf
          |
          +-- pom.xml          # Các dependencies của ứng dụng
```

## 2.3. Cấu hình cơ sở dữ liệu MySQL trong Spring Boot

- **Cấu hình MySQL trong application.properties:**

```
spring.datasource.url=jdbc:mysql://localhost:3306/your_db_name
spring.datasource.username=root
spring.datasource.password=your_password
spring.jpa.hibernate.ddl-auto=update # Tự động cập nhật cấu trúc cơ sở dữ liệu
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect
```

- **Entity Classes:** Trong Spring Boot, các bảng cơ sở dữ liệu sẽ được đại diện bởi các lớp Entity. Ví dụ, lớp User sẽ tương ứng với bảng người dùng trong cơ sở dữ liệu.

```
public class User {  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
    private String username;  
    private String email;  
    private String password;  
    // Getters and Setters  
}
```

- **Repository Layer:** Để thực hiện các thao tác CRUD với cơ sở dữ liệu, bạn sẽ sử dụng Spring Data JPA để tạo các repository.

```
@Repository  
public interface UserRepository extends JpaRepository<User, Long> {  
    Optional<User> findByUsername(String username);  
}
```

## 2.4. Các API chính

### API của User Service

Tài liệu này mô tả các API được cung cấp bởi **User Service** để quản lý người dùng, cư dân và quá trình đăng ký.

#### 1. Quản lý Xác thực (AuthController)

Endpoint gốc: /api/auth

##### 1.1. Đăng nhập

- **Mô tả:** Xác thực người dùng và cấp token truy cập.
- **Endpoint:** POST /login
- **Input:**
  - username: String (tên đăng nhập)
  - password: String (mật khẩu)
- **Output:**
  - **Thành công (200 OK):**

```
{  
    "message": "Đăng nhập thành công",  
    "id": "Long",  
    "loaiTaiKhoan": "String",  
    "status": "String"  
}
```

- **Thất bại:**
  - **401 Unauthorized:** "Tên đăng nhập/mật khẩu sai"
  - **403 Forbidden:** "Tài khoản bị khóa"
- **Luồng hoạt động:**
  1. Nhận request chứa username và password.
  2. Tìm kiếm tài khoản theo username trong database.
  3. Nếu không tìm thấy tài khoản, trả về lỗi 401.
  4. Nếu tìm thấy, kiểm tra mật khẩu bằng PasswordUtil.matches().
  5. Nếu mật khẩu không khớp, trả về lỗi 401.
  6. Kiểm tra trạng thái khóa của tài khoản.
  7. Nếu tài khoản bị khóa, trả về lỗi 403.
  8. Trả về thông tin đăng nhập thành công.

## 2. Quản lý Cư dân (CuDanController)

Endpoint gốc: /api/cudan

### 2.1. Lấy thông tin cư dân và căn hộ

- **Mô tả:** Lấy thông tin chi tiết của một cư dân và căn hộ của họ bằng ID.
- **Endpoint:** GET /{id}
- **Output:**
  - **Thành công (200 OK):**

```
{  
    "message": "Lấy thông tin thành công",  
    "cuDan": "Object",  
    "canHo": "Object"  
}
```
  - **Thất bại (404 Not Found):** "Không tìm thấy cư dân với ID: {id}"
- **Luồng hoạt động:**
  1. Nhận ID cư dân từ path parameter.
  2. Tìm kiếm cư dân theo ID trong database.
  3. Nếu không tìm thấy, trả về lỗi 404.
  4. Lấy thông tin căn hộ từ quan hệ @ManyToOne.
  5. Đóng gói thông tin cư dân và căn hộ vào response.
  6. Trả về kết quả thành công.

### 2.2. Cập nhật thông tin cư dân

- **Mô tả:** Cập nhật thông tin của một cư dân bằng ID.
- **Endpoint:** PUT /{id}
- **Input (CuDanDTO):**

## Báo cáo Thực tập Tốt nghiệp Đại học

- hoTen: String
- email: String
- soDienThoai: String
- cccd: String
- diaChi: String
- trangThai: String (giá trị hợp lệ: ở hoặc không\_ở\_nữa)
- ngaySinh: String (định dạng yyyy-MM-dd)
- maCanHo: String
- **Output:**
  - **Thành công (200 OK):** "Cập nhật thông tin cư dân thành công"
  - **Thất bại:**
    - **404 Not Found:** "Không tìm thấy cư dân với id: {id}"
    - **400 Bad Request:** Các lỗi liên quan đến validate dữ liệu đầu vào.
- **Luồng hoạt động:**
  1. Nhận ID cư dân và dữ liệu cập nhật (CuDanDTO).
  2. Tìm kiếm cư dân theo ID.
  3. Nếu không tìm thấy, trả về lỗi 404.
  4. Cập nhật thông tin cơ bản (họ tên, email, số điện thoại, CCCD, địa chỉ).
  5. Kiểm tra và chuyển đổi trạng thái cư dân.
  6. Chuyển đổi và validate ngày sinh.
  7. Kiểm tra mã căn hộ mới.
  8. So sánh mã căn hộ mới với mã căn hộ hiện tại.
  9. Nếu mã căn hộ thay đổi, cập nhật trạng thái tài khoản thành "Chờ duyệt".
  10. Lưu thay đổi vào database.

### 2.3. Lấy danh sách cư dân

- **Mô tả:** Lấy danh sách thông tin của tất cả cư dân.
- **Endpoint:** GET /thong-tin
- **Output:**
  - **Thành công (200 OK):** Danh sách ThongTinCuDanDTO.
- **Luồng hoạt động:**
  1. Lấy toàn bộ danh sách cư dân từ database.
  2. Chuyển đổi từng cư dân sang ThongTinCuDanDTO.
  3. Đóng gói thông tin: ID, họ tên, mã căn hộ, email, số điện thoại, trạng thái tài khoản, ngày đăng ký.
  4. Trả về danh sách DTO.

## 3. Đăng ký (DangKyController)

Endpoint gốc: /api/dangky

### 3.1. Đăng ký tài khoản cư dân

- **Mô tả:** Đăng ký tài khoản mới cho cư dân.
- **Endpoint:** POST /
- **Input (DangKyCuDanRequest):**
  - tenDangNhap: String
  - matKhau: String

## Báo cáo Thực tập Tốt nghiệp Đại học

- hoTen: String
- email: String (yêu cầu định dạng @gmail.com)
- soDienThoai: String
- ngaySinh: LocalDate
- cccd: String
- diaChi: String
- canHoid: String
- **Output:**
  - **Thành công (200 OK):** "Đăng ký thành công. Vui lòng chờ duyệt."
  - **Thất bại (400 Bad Request):**
    - "Tên đăng nhập đã tồn tại"
    - "CCCD đã được đăng ký"
    - "Email không hợp lệ"
- **Luồng hoạt động:**
  1. Kiểm tra tên đăng nhập đã tồn tại.
  2. Tạo tài khoản mới với trạng thái "Chờ duyệt".
  3. Mã hóa mật khẩu và set các thông tin tài khoản.
  4. Lưu tài khoản để lấy ID.
  5. Kiểm tra mã căn hộ tồn tại.
  6. Kiểm tra CCCD đã được đăng ký.
  7. Validate định dạng email (@gmail.com).
  8. Tạo thông tin cư dân và liên kết với tài khoản.
  9. Lưu thông tin cư dân.

### 3.2. Lấy danh sách căn hộ

- **Mô tả:** Lấy danh sách tất cả căn hộ có trong hệ thống.
- **Endpoint:** GET /canho
- **Output:**
  - **Thành công (200 OK):** Danh sách CanHo.
- **Luồng hoạt động:**
  1. Lấy toàn bộ danh sách căn hộ từ database.
  2. Trả về danh sách.

## 4. Quản lý Tài khoản (TaiKhoanController)

Endpoint gốc: /api/taikhoan

### 4.1. Duyệt/Hủy duyệt tài khoản

- **Mô tả:** Duyệt hoặc hủy duyệt một tài khoản đang ở trạng thái chờ.
- **Endpoint:** PUT /duyet/{id}
- **Input:**
  - action: String (giá trị hợp lệ: duyet hoặc huy)
- **Output:**
  - **Thành công (200 OK):**
    - "Duyệt tài khoản thành công"
    - "Hủy duyệt tài khoản thành công"
  - **Thất bại:**

- **404 Not Found:** "Không tìm thấy tài khoản"
- **400 Bad Request:** "Tài khoản không ở trạng thái 'Chờ duyệt'"

- **Luồng hoạt động:**

1. Nhận ID tài khoản và action (duyet/huy).
2. Tìm kiếm tài khoản theo ID.
3. Kiểm tra tài khoản có ở trạng thái "Chờ duyệt".
4. Nếu action là "duyet", set trạng thái "Đang hoạt động".
5. Nếu action là "huy", set trạng thái "Đã khóa".
6. Lưu thay đổi vào database.

#### 4.2. Khóa/Mở khóa tài khoản

- **Mô tả:** Chuyển đổi trạng thái khóa/mở khóa của một tài khoản.
- **Endpoint:** PUT /toggle-khoa/{id}
- **Output:**

- **Thành công (200 OK):**

```
{  
  "message": "Cập nhật trạng thái tài khoản thành công",  
  "trangThaiMoi": "String"  
}
```

- **Thất bại (404 Not Found):** "Không tìm thấy tài khoản"

- **Luồng hoạt động:**

1. Nhận ID tài khoản.
2. Tìm kiếm tài khoản theo ID.
3. Kiểm tra trạng thái hiện tại.
4. Nếu đang hoạt động, chuyển sang trạng thái khóa.
5. Nếu đang khóa, chuyển sang trạng thái hoạt động.
6. Lưu thay đổi và trả về trạng thái mới.

### API của *Service Addendum* Service

#### 1. MauHopDongDichVuController

**Chức năng:** Quản lý các liên kết giữa mẫu hợp đồng và dịch vụ.

- **API 1: Lấy tất cả liên kết mẫu - dịch vụ**
  - **Endpoint:** GET /api/mauhopdongdichvu
  - **Mô tả:** Lấy danh sách tất cả các liên kết giữa mẫu hợp đồng và dịch vụ.
  - **Input:** Không có.
  - **Output:** Danh sách các liên kết mẫu - dịch vụ dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 500 Internal Server Error
- **API 2: Lấy liên kết mẫu - dịch vụ theo ID**
  - **Endpoint:** GET /api/mauhopdongdichvu/{maulId}/{maDichVu}
  - **Mô tả:** Lấy liên kết mẫu hợp đồng với dịch vụ theo maulId và maDichVu.
  - **Input:** maulId, maDichVu (tham số đường dẫn).
  - **Output:** Dữ liệu của liên kết mẫu - dịch vụ dưới dạng JSON.
  - **Mã thành công:** 200 OK

- **Mã lỗi:** 404 Not Found (nếu không tìm thấy liên kết).
- **API 3: Thêm liên kết mẫu - dịch vụ**
  - **Endpoint:** POST /api/mauhopdongdichvu
  - **Mô tả:** Thêm mới liên kết mẫu hợp đồng và dịch vụ.
  - **Input:** Đối tượng MauHopDongDichVu (bao gồm mã mẫu hợp đồng và mã dịch vụ).
  - **Output:** Liên kết mẫu - dịch vụ mới đã được thêm.
  - **Mã thành công:** 201 Created
  - **Mã lỗi:** 400 Bad Request (nếu dữ liệu không đầy đủ hoặc mẫu hợp đồng/dịch vụ không tồn tại), 409 Conflict (nếu liên kết đã tồn tại).
- **API 4: Xóa liên kết mẫu - dịch vụ**
  - **Endpoint:** DELETE /api/mauhopdongdichvu/{maulId}/{maDichVu}
  - **Mô tả:** Xóa liên kết mẫu hợp đồng và dịch vụ.
  - **Input:** maulId, maDichVu (tham số đường dẫn).
  - **Output:** Thông báo thành công nếu xóa thành công.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy liên kết để xóa).

**Luồng hoạt động:** Các API này cho phép bạn quản lý các liên kết giữa mẫu hợp đồng và dịch vụ. Khi thêm hoặc xóa một liên kết, hệ thống kiểm tra tính hợp lệ của dữ liệu đầu vào và đảm bảo rằng mẫu hợp đồng và dịch vụ tồn tại trong cơ sở dữ liệu. Các phản hồi từ server sẽ cung cấp thông tin về trạng thái thực hiện yêu cầu.

## 2. GoiCuocDichVuController

### Chức năng: Quản lý các gói cước dịch vụ.

- **API 1: Lấy tất cả gói cước dịch vụ**
  - **Endpoint:** GET /api/goicuocdichvu
  - **Mô tả:** Lấy danh sách tất cả các gói cước dịch vụ.
  - **Input:** Không có.
  - **Output:** Danh sách các gói cước dịch vụ dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 500 Internal Server Error
- **API 2: Lấy gói cước dịch vụ theo ID**
  - **Endpoint:** GET /api/goicuocdichvu/{id}
  - **Mô tả:** Lấy thông tin gói cước dịch vụ theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Dữ liệu gói cước dịch vụ dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy gói cước).
- **API 3: Thêm gói cước dịch vụ mới**
  - **Endpoint:** POST /api/goicuocdichvu
  - **Mô tả:** Thêm một gói cước dịch vụ mới.
  - **Input:** Đối tượng GoiCuocDichVu (bao gồm tên gói, đơn giá, mô tả, ngày hiệu lực, trạng thái).
  - **Output:** Gói cước dịch vụ mới đã được thêm.
  - **Mã thành công:** 201 Created
  - **Mã lỗi:** 400 Bad Request (nếu có lỗi validation).
- **API 4: Cập nhật gói cước dịch vụ**
  - **Endpoint:** PUT /api/goicuocdichvu/{id}
  - **Mô tả:** Cập nhật thông tin gói cước dịch vụ theo ID.

- **Input:** id (tham số đường dẫn), đối tượng GoiCuocDichVu (thông tin mới).
- **Output:** Gói cước dịch vụ đã được cập nhật.
- **Mã thành công:** 200 OK
- **Mã lỗi:** 404 Not Found (nếu không tìm thấy gói cước), 400 Bad Request (nếu có lỗi validation).
- **API 5: Xóa gói cước dịch vụ**
  - **Endpoint:** DELETE /api/goicuocdichvu/{id}
  - **Mô tả:** Xóa gói cước dịch vụ theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Thông báo thành công nếu xóa thành công.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy gói cước).

**Luồng hoạt động:** Các API này giúp bạn quản lý gói cước dịch vụ, cho phép thêm mới, sửa đổi, và xóa gói cước. Trong quá trình thêm mới hoặc sửa đổi, hệ thống kiểm tra tính hợp lệ của các trường dữ liệu (như tên gói, đơn giá, ngày hiệu lực).

### 3. CauHinhDichVuController

#### Chức năng: Quản lý cấu hình dịch vụ.

- **API 1: Lấy tất cả cấu hình dịch vụ**
  - **Endpoint:** GET /api/cauhinhdichvu
  - **Mô tả:** Lấy danh sách tất cả các cấu hình dịch vụ.
  - **Input:** Không có.
  - **Output:** Danh sách các cấu hình dịch vụ dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 500 Internal Server Error
- **API 2: Lấy cấu hình dịch vụ theo ID**
  - **Endpoint:** GET /api/cauhinhdichvu/{id}
  - **Mô tả:** Lấy thông tin cấu hình dịch vụ theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Dữ liệu cấu hình dịch vụ dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy cấu hình).
- **API 3: Thêm cấu hình dịch vụ mới**
  - **Endpoint:** POST /api/cauhinhdichvu
  - **Mô tả:** Thêm một cấu hình dịch vụ mới.
  - **Input:** Đối tượng CauHinhDichVu (bao gồm tên cấu hình, ngày hiệu lực, mã dịch vụ).
  - **Output:** Cấu hình dịch vụ mới đã được thêm.
  - **Mã thành công:** 201 Created
  - **Mã lỗi:** 400 Bad Request (nếu có lỗi validation).
- **API 4: Cập nhật cấu hình dịch vụ**
  - **Endpoint:** PUT /api/cauhinhdichvu/{id}
  - **Mô tả:** Cập nhật thông tin cấu hình dịch vụ theo ID.
  - **Input:** id (tham số đường dẫn), đối tượng CauHinhDichVu (thông tin mới).
  - **Output:** Cấu hình dịch vụ đã được cập nhật.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy cấu hình), 400 Bad Request (nếu có lỗi validation).
- **API 5: Xóa cấu hình dịch vụ**

- **Endpoint:** DELETE /api/cauhinhdichvu/{id}
- **Mô tả:** Xóa cấu hình dịch vụ theo ID.
- **Input:** id (tham số đường dẫn).
- **Output:** Thông báo thành công nếu xóa thành công.
- **Mã thành công:** 200 OK
- **Mã lỗi:** 404 Not Found (nếu không tìm thấy cấu hình).

**Luồng hoạt động:** Các API này cho phép bạn quản lý các cấu hình dịch vụ, bao gồm thêm mới, sửa đổi và xóa cấu hình dịch vụ. Hệ thống sẽ kiểm tra tính hợp lệ của các trường dữ liệu (như tên cấu hình, ngày hiệu lực) và kiểm tra sự tồn tại của dịch vụ trước khi thực hiện các thao tác.

#### 4. MauHopDongController

**Chức năng:** Quản lý các mẫu hợp đồng.

- **API 1: Lấy tất cả mẫu hợp đồng**
  - **Endpoint:** GET /api/mauhopdong
  - **Mô tả:** Lấy danh sách tất cả các mẫu hợp đồng.
  - **Input:** Không có.
  - **Output:** Danh sách các mẫu hợp đồng dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 500 Internal Server Error
- **API 2: Lấy mẫu hợp đồng theo ID**
  - **Endpoint:** GET /api/mauhopdong/{id}
  - **Mô tả:** Lấy thông tin mẫu hợp đồng theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Dữ liệu mẫu hợp đồng dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy mẫu hợp đồng).
- **API 3: Thêm mẫu hợp đồng mới**
  - **Endpoint:** POST /api/mauhopdong
  - **Mô tả:** Thêm một mẫu hợp đồng mới.
  - **Input:** Đối tượng MauHopDong (bao gồm tên mẫu hợp đồng, các điều khoản, ngày hiệu lực).
  - **Output:** Mẫu hợp đồng mới đã được thêm.
  - **Mã thành công:** 201 Created
  - **Mã lỗi:** 400 Bad Request (nếu có lỗi validation).
- **API 4: Cập nhật mẫu hợp đồng**
  - **Endpoint:** PUT /api/mauhopdong/{id}
  - **Mô tả:** Cập nhật thông tin mẫu hợp đồng theo ID.
  - **Input:** id (tham số đường dẫn), đối tượng MauHopDong (thông tin mới).
  - **Output:** Mẫu hợp đồng đã được cập nhật.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy mẫu hợp đồng), 400 Bad Request (nếu có lỗi validation).
- **API 5: Xóa mẫu hợp đồng**
  - **Endpoint:** DELETE /api/mauhopdong/{id}
  - **Mô tả:** Xóa mẫu hợp đồng theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Thông báo thành công nếu xóa thành công.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy mẫu hợp đồng).

**Luồng hoạt động:** Các API này giúp quản lý mẫu hợp đồng, cho phép thêm mới, sửa đổi, và xóa mẫu hợp đồng. Hệ thống kiểm tra tính hợp lệ của các trường dữ liệu và đảm bảo rằng các mẫu hợp đồng tồn tại trong cơ sở dữ liệu trước khi thực hiện các thao tác.

## 5. PhuLucDichVuController

**Chức năng:** Quản lý phụ lục dịch vụ.

- **API 1: Lấy tất cả phụ lục dịch vụ**
  - **Endpoint:** GET /api/phulucdichvu
  - **Mô tả:** Lấy danh sách tất cả các phụ lục dịch vụ.
  - **Input:** Không có.
  - **Output:** Danh sách các phụ lục dịch vụ dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 500 Internal Server Error
- **API 2: Lấy phụ lục dịch vụ theo ID**
  - **Endpoint:** GET /api/phulucdichvu/{id}
  - **Mô tả:** Lấy thông tin phụ lục dịch vụ theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Dữ liệu phụ lục dịch vụ dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy phụ lục dịch vụ).
- **API 3: Thêm phụ lục dịch vụ mới**
  - **Endpoint:** POST /api/phulucdichvu
  - **Mô tả:** Thêm một phụ lục dịch vụ mới.
  - **Input:** Đối tượng PhuLucDichVu (bao gồm tên phụ lục, mã dịch vụ, mô tả).
  - **Output:** Phụ lục dịch vụ mới đã được thêm.
  - **Mã thành công:** 201 Created
  - **Mã lỗi:** 400 Bad Request (nếu có lỗi validation).
- **API 4: Cập nhật phụ lục dịch vụ**
  - **Endpoint:** PUT /api/phulucdichvu/{id}
  - **Mô tả:** Cập nhật thông tin phụ lục dịch vụ theo ID.
  - **Input:** id (tham số đường dẫn), đối tượng PhuLucDichVu (thông tin mới).
  - **Output:** Phụ lục dịch vụ đã được cập nhật.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy phụ lục dịch vụ), 400 Bad Request (nếu có lỗi validation).
- **API 5: Xóa phụ lục dịch vụ**
  - **Endpoint:** DELETE /api/phulucdichvu/{id}
  - **Mô tả:** Xóa phụ lục dịch vụ theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Thông báo thành công nếu xóa thành công.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy phụ lục dịch vụ).

**Luồng hoạt động:** Các API này cho phép bạn quản lý phụ lục dịch vụ, bao gồm thêm mới, sửa đổi và xóa phụ lục. Hệ thống sẽ kiểm tra tính hợp lệ của dữ liệu và xác minh sự tồn tại của phụ lục dịch vụ trước khi thực hiện các thao tác.

## 6. ThamSoPhiController

**Chức năng:** Quản lý các tham số phí.

- **API 1: Lấy tất cả tham số phí**
  - **Endpoint:** GET /api/thamsophi
  - **Mô tả:** Lấy danh sách tất cả các tham số phí.
  - **Input:** Không có.
  - **Output:** Danh sách các tham số phí dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 500 Internal Server Error
- **API 2: Lấy tham số phí theo ID**
  - **Endpoint:** GET /api/thamsophi/{id}
  - **Mô tả:** Lấy thông tin tham số phí theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Dữ liệu tham số phí dưới dạng JSON.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy tham số phí).
- **API 3: Thêm tham số phí mới**
  - **Endpoint:** POST /api/thamsophi
  - **Mô tả:** Thêm một tham số phí mới.
  - **Input:** Đối tượng ThamSoPhi (bao gồm mã phí, tên phí, giá trị phí).
  - **Output:** Tham số phí mới đã được thêm.
  - **Mã thành công:** 201 Created
  - **Mã lỗi:** 400 Bad Request (nếu có lỗi validation).
- **API 4: Cập nhật tham số phí**
  - **Endpoint:** PUT /api/thamsophi/{id}
  - **Mô tả:** Cập nhật thông tin tham số phí theo ID.
  - **Input:** id (tham số đường dẫn), đối tượng ThamSoPhi (thông tin mới).
  - **Output:** Tham số phí đã được cập nhật.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy tham số phí), 400 Bad Request (nếu có lỗi validation).
- **API 5: Xóa tham số phí**
  - **Endpoint:** DELETE /api/thamsophi/{id}
  - **Mô tả:** Xóa tham số phí theo ID.
  - **Input:** id (tham số đường dẫn).
  - **Output:** Thông báo thành công nếu xóa thành công.
  - **Mã thành công:** 200 OK
  - **Mã lỗi:** 404 Not Found (nếu không tìm thấy tham số phí).

**Luồng hoạt động:** API này giúp bạn quản lý các tham số phí, bao gồm thêm mới, sửa đổi và xóa các tham số phí. Hệ thống sẽ kiểm tra tính hợp lệ của các tham số phí và đảm bảo rằng các tham số tồn tại trong cơ sở dữ liệu.

## Chương 5: Kết Luận

### 5.1 Kết quả đạt được

Trong quá trình triển khai hệ thống quản lý cho cư dân và ban quản lý, các mục tiêu ban đầu của dự án đã được hoàn thành với các kết quả đáng khích lệ:

- Cấu trúc hệ thống hoàn thiện:** Hệ thống được xây dựng theo kiến trúc microservices, mỗi dịch vụ đảm nhận một chức năng riêng biệt, giúp tối ưu hóa hiệu suất và bảo trì hệ thống. Các vi dịch vụ chính (quản lý người dùng, hợp đồng, thanh toán, thông báo, và quản lý dịch vụ) đã được triển khai thành công và tích hợp với nhau qua các API RESTful.
- Ứng dụng web thân thiện với người dùng:** Giao diện web dành cho cả ban quản lý và cư dân đã được xây dựng dễ sử dụng và phù hợp với nhu cầu thực tế. Cư dân có thể đăng ký tài khoản, quản lý hợp đồng, và thực hiện thanh toán trực tuyến một cách nhanh chóng. Ban quản lý có thể duyệt tài khoản, quản lý hợp đồng và xử lý thanh toán mà không gặp phải khó khăn nào.
- Khả năng mở rộng và bảo trì dễ dàng:** Kiến trúc microservices giúp hệ thống dễ dàng mở rộng và bảo trì. Mỗi dịch vụ có thể được nâng cấp hoặc thay đổi mà không ảnh hưởng đến các dịch vụ khác.

### 5.2 Lợi ích đối với ban quản lý và cư dân

#### 1. Đối với ban quản lý:

- Quản lý dễ dàng:** Ban quản lý có thể dễ dàng cấp/duyệt/hủy tài khoản cư dân, quản lý hợp đồng và dịch vụ, và thanh toán hóa đơn.
- Tối ưu hóa công việc:** Các công việc như tạo hóa đơn tự động và quyết toán được tự động hóa, giúp tiết kiệm thời gian và giảm sai sót.
- Theo dõi hiệu quả dịch vụ:** Ban quản lý có thể dễ dàng theo dõi và báo cáo các dịch vụ đang hoạt động, các hợp đồng và tình hình thanh toán.

#### 2. Đối với cư dân:

- Tạo hợp đồng dễ dàng:** Cư dân có thể dễ dàng đăng ký tài khoản, tạo hợp đồng và lựa chọn dịch vụ theo nhu cầu.
- Quản lý thông tin nhanh chóng:** Cư dân có thể cập nhật thông tin hợp đồng và tra cứu nội dung hợp đồng bất kỳ lúc nào.
- Thanh toán online:** Cư dân có thể thanh toán trực tuyến cho các dịch vụ một cách nhanh chóng và an toàn.
- Nhận thông báo:** Hệ thống gửi cảnh báo khi hợp đồng sắp hết hạn hoặc đến kỳ thanh toán, giúp cư dân không bỏ sót các thông tin quan trọng.

### 5.3 Hạn chế còn tồn tại

- Khả năng mở rộng thêm tính năng:** Mặc dù hệ thống đã có tính linh hoạt cao nhờ kiến trúc microservices, nhưng việc thêm các tính năng mới như tích hợp các cổng thanh toán khác hoặc các dịch vụ phụ trợ có thể yêu cầu phải tái cấu trúc một phần hệ thống.
- Tính năng báo cáo:** Hiện tại hệ thống chưa có các tính năng báo cáo chi tiết và tùy chỉnh cho ban quản lý. Việc thêm các báo cáo trực quan có thể là một yêu cầu trong tương lai.
- Quản lý và bảo trì các dịch vụ:** Dù hệ thống hoạt động ổn định, nhưng nếu có nhiều vi dịch vụ được triển khai trong tương lai, việc quản lý và giám sát tất cả các dịch vụ có thể gặp phải một số thách thức.

4. **Tốc độ xử lý khi người dùng tăng:** Mặc dù hệ thống hiện tại hoạt động tốt, nhưng khi số lượng người dùng gia tăng đáng kể, việc tối ưu hóa hiệu suất của hệ thống cần được chú trọng thêm.
5. **Chưa hoàn thiện được tất cả các tính năng:**

#### 5.4 Hướng phát triển, nâng cấp trong tương lai

1. **Hoàn thiện các tính năng còn lại**
2. **Mở rộng các tính năng báo cáo:** Cung cấp cho ban quản lý khả năng tạo báo cáo chi tiết và tùy chỉnh về các dịch vụ, hợp đồng và tình hình thanh toán. Các báo cáo trực quan sẽ giúp quản lý dễ dàng hơn trong việc phân tích và đưa ra quyết định.
3. **Tích hợp thêm công thanh toán:** Mở rộng khả năng thanh toán trực tuyến bằng cách tích hợp thêm các công thanh toán khác như PayPal, Stripe, hoặc các phương thức thanh toán quốc tế.
4. **Cải thiện giao diện người dùng:** Cải thiện thêm giao diện người dùng cho cả ban quản lý và cư dân, tối ưu hóa trải nghiệm người dùng trên cả desktop và mobile. Cập nhật thêm các tính năng như giao diện động, lọc và tìm kiếm thông tin dễ dàng hơn.
5. **Xử lý và phân tích dữ liệu lớn:** Hệ thống có thể được phát triển thêm để hỗ trợ phân tích dữ liệu lớn (Big Data), giúp ban quản lý đưa ra các quyết định chính xác hơn dựa trên dữ liệu hành vi người dùng và các thống kê về dịch vụ.
6. **Cải tiến bảo mật:** Tiếp tục cải tiến các biện pháp bảo mật, bao gồm mã hóa dữ liệu, xác thực đa yếu tố, và bảo vệ các dịch vụ chống lại các cuộc tấn công bảo mật.
7. **Tích hợp với các dịch vụ bên ngoài:** Xem xét tích hợp với các dịch vụ bên ngoài như hệ thống quản lý tài sản, quản lý bảo trì, hoặc các dịch vụ tiện ích khác để mở rộng tính năng của hệ thống.

## Tài liệu tham khảo

1. R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture," *ACM Transactions on Internet Technology (TOIT)*, vol. 2, no. 2, pp. 115-150, May 2002.
2. M. Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley, 2002.
3. S. Newman, *Building Microservices: Designing Fine-Grained Systems*, O'Reilly Media, 2015.
4. A. W. Brown, *Spring in Action*, 5th ed. New York: Manning Publications, 2018.
5. S. K. Dubey, "A survey of microservices architecture," *International Journal of Computer Applications*, vol. 157, no. 5, pp. 40-45, Jan. 2017. doi: 10.5120/ijca2017913237.
6. P. Choudhury, "Integrating payment gateway into ecommerce system using API," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 4, pp. 10-18, 2018.
7. R. C. Seibel, "Spring Boot: A microservice-based architecture for modern applications," *Journal of Software Engineering and Applications*, vol. 7, pp. 235-242, Dec. 2014.
8. M. Lewis and M. L. Fowler, "Microservices patterns: With examples in Java," *Springer*, 2019.
9. J. A. McKinsey & Company, "The future of work: Reshaping the work environment," McKinsey Global Institute Report, 2017. [Online]. Available: <https://www.mckinsey.com/insights/future-of-work>
10. "MySQL 8.0 Reference Manual," Oracle, 2019. [Online]. Available: <https://dev.mysql.com/doc/>

## Bảng phân công nhiệm vụ

Nguyễn Thành Long - N21DCCN143	<ul style="list-style-type: none"><li>-Phân tích và thiết kế các chức năng:<ul style="list-style-type: none"><li>Xác nhận thanh toán (ban quản lý)</li></ul></li><li>Chức năng dành cho cư dân:<ul style="list-style-type: none"><li>• Quản lý tài khoản cá nhân:<ul style="list-style-type: none"><li>+ Đăng ký tài khoản.</li><li>+ Đăng nhập, cập nhật thông tin cá nhân.</li></ul></li><li>• Đăng ký hợp đồng dịch vụ:<ul style="list-style-type: none"><li>+ Tạo hợp đồng mới, chọn dịch vụ cần dùng.</li><li>+ Xem điều khoản dịch vụ.</li><li>+ Chính sửa dịch vụ trong hợp đồng.</li></ul></li><li>-Viết Backend và Frontend cho các chức năng</li><li>Chức năng dành cho ban quản lý</li><li>• Quản lý dịch vụ và hợp đồng:<ul style="list-style-type: none"><li>+ Khai báo danh mục dịch vụ: thuê nhà, điện, nước, bảo trì, vệ sinh, bảo vệ, gửi xe...</li><li>+ Định nghĩa điều khoản áp dụng cho từng dịch vụ.</li><li>+ Tạo mẫu hợp đồng để cư dân chọn.</li></ul></li><li>Chức năng dành cho cư dân:<ul style="list-style-type: none"><li>• Đăng ký hợp đồng dịch vụ:<ul style="list-style-type: none"><li>+ Tạo hợp đồng mới, chọn dịch vụ cần dùng.</li><li>+ Xem điều khoản dịch vụ.</li><li>+ Chính sửa dịch vụ trong hợp đồng.</li></ul></li><li>• Tra cứu hợp đồng:<ul style="list-style-type: none"><li>+ Xem danh sách hợp đồng đã ký.</li><li>+ Xem trạng thái thanh toán, thời hạn hợp đồng.</li></ul></li><li>• Thanh toán:</li></ul></li></ul></li></ul>
--------------------------------	--

Báo cáo Thực tập Tốt nghiệp Đại học

	<ul style="list-style-type: none"> <li>+ Thanh toán thủ công (chuyển khoản, tiền mặt).</li> <li>+ Thanh toán online qua VNPay.</li> <li>- Chính sửa và hoàn thiện word</li> </ul>
Vũ Đức Trọng - N21DCCN190	<p>-Phân tích và thiết kế các chức năng:</p> <p>Chức năng dành cho ban quản lý:</p> <ul style="list-style-type: none"> <li>• Quản lý tài khoản cư dân: <ul style="list-style-type: none"> <li>+ Xem danh sách cư dân đã đăng ký.</li> <li>+ Duyệt hoặc hủy tài khoản.</li> <li>+ Khóa/mở khóa tài khoản khi cần.</li> </ul> </li> <li>• Quản lý dịch vụ và hợp đồng: <ul style="list-style-type: none"> <li>+ Khai báo danh mục dịch vụ: thuê nhà, điện, nước, bảo trì, vệ sinh, bảo vệ, gửi xe...</li> <li>+ Định nghĩa điều khoản áp dụng cho từng dịch vụ.</li> <li>+ Tạo mẫu hợp đồng để cư dân chọn.</li> </ul> </li> <li>• Quản lý hóa đơn và ghi chi số: <ul style="list-style-type: none"> <li>+ Lập hóa đơn tự động hằng tháng dựa trên hợp đồng.</li> <li>+ Ghi chỉ số điện/nước thủ công</li> </ul> </li> </ul> <p>-Viết backend và frontend cho các chức năng:</p> <p>Chức năng dành cho ban quản lý</p> <ul style="list-style-type: none"> <li>• Quản lý tài khoản cư dân: <ul style="list-style-type: none"> <li>+ Xem danh sách cư dân đã đăng ký.</li> <li>+ Duyệt hoặc hủy tài khoản.</li> <li>+ Khóa/mở khóa tài khoản khi cần.</li> </ul> </li> </ul> <p>Chức năng dành cho cư dân</p> <ul style="list-style-type: none"> <li>• Quản lý tài khoản cá nhân: <ul style="list-style-type: none"> <li>+ Đăng ký tài khoản.</li> <li>+ Đăng nhập, cập nhật thông tin cá nhân.</li> </ul> </li> <li>• Tra cứu hợp đồng:</li> </ul>

Báo cáo Thực tập Tốt nghiệp Đại học

	<ul style="list-style-type: none"> <li>+ Xem danh sách hợp đồng đã ký.</li> <li>+ Xem trạng thái thanh toán, thời hạn hợp đồng.</li> <li>• Thanh toán: <ul style="list-style-type: none"> <li>+ Thanh toán thủ công (chuyển khoản, tiền mặt).</li> <li>+ Thanh toán online qua VNPay.</li> </ul> </li> </ul>
Thạch Thị Nhanh – N21DCCN159	<p>-Phân tích và thiết kế các chức năng:</p> <p>Chức năng dành cho cư dân:</p> <ul style="list-style-type: none"> <li>• Tra cứu hợp đồng: <ul style="list-style-type: none"> <li>+ Xem danh sách hợp đồng đã ký.</li> <li>+ Xem trạng thái thanh toán, thời hạn hợp đồng.</li> </ul> </li> <li>• Thanh toán: <ul style="list-style-type: none"> <li>+ Thanh toán thủ công (chuyển khoản, tiền mặt).</li> <li>+ Thanh toán online qua VNPay.</li> </ul> </li> <li>• Nhận thông báo: <ul style="list-style-type: none"> <li>+ Nhắc nhở thời hạn hợp đồng.</li> <li>+ Thông báo khi có hóa đơn mới hoặc sắp hết hạn dịch vụ.</li> <li>+ Gửi qua email và hiển thị trên giao diện web.</li> </ul> </li> </ul> <p>-Viết backend và frontend cho các chức năng:</p> <p>Chức năng dành cho ban quản lý:</p> <ul style="list-style-type: none"> <li>• Quản lý hóa đơn và ghi chỉ số: <ul style="list-style-type: none"> <li>+ Lập hóa đơn tự động hàng tháng dựa trên hợp đồng.</li> <li>+ Ghi chỉ số điện/nước thủ công</li> <li>+ Xác nhận thanh toán (thủ công hoặc online).</li> </ul> </li> </ul> <p>Chức năng dành cho cư dân:</p> <ul style="list-style-type: none"> <li>• Thanh toán:</li> </ul>

	<ul style="list-style-type: none"><li>+ Thanh toán thủ công (chuyển khoản, tiền mặt).</li><li>+ Thanh toán online qua VNPAY.</li><li>• Nhận thông báo:<ul style="list-style-type: none"><li>+ Nhắc nhở thời hạn hợp đồng.</li><li>+ Thông báo khi có hóa đơn mới hoặc sắp hết hạn dịch vụ.</li><li>+ Gửi qua email và hiển thị trên giao diện web.</li></ul></li></ul>
--	--