**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN**

****

**BÀI TẬP LỚN MÔN CÔNG NGHỆ PHẦN MỀM**

**PHÂN TÍCH THIẾT KẾ HỆ THỐNG ỨNG DỤNG QUẢN LÝ BÁN MẢNH ĐẤT Ở TỈNH HƯNG YÊN**

NGÀNH: KỸ THUẬT PHẦN MỀM

SINH VIÊN: **VŨ PHONG PHÚ**

MÃ LỚP: **125217**

HƯỚNG DẪN: **TS. ĐÀO ANH HIỂN**

**HƯNG YÊN – 2023**

**NHẬN XÉT**

**Nhận xét của giáo viên hướng dẫn**

**GIÁO VIÊN HƯỚNG DẪN**

**TS. ĐÀO ANH HIỂN**

**LỜI CAM ĐOAN**

Em xin cam đoan bài tập lớn “Phân tích thiết kế hệ thống ứng dụng quản lý bán mảnh đất ở tỉnh Hưng yên” là kết quả thực hiện của bản thân em dưới sự hướng dẫn của thầy Đào Anh Hiển.

Những phần sử dụng tài liệu tham khảo trong bài tập lớn đã được nêu rõ trong phần tài liệu tham khảo. Các kết quả trình bày trong bài tập lớn và chương trình xây dựng được hoàn toàn là kết quả do bản thân em thực hiện.

Nếu vi phạm lời cam đoan này, em xin chịu hoàn toàn trách nhiệm trước khoa và nhà trường.

*Hưng Yên, ngày … tháng 6 năm 2023*

Sinh viên

Vũ Phong Phú

**LỜI CẢM ƠN**

Để có thể hoàn thành bài tập lớn này, lời đầu tiên em xin phép gửi lời cảm ơn tới bộ môn Công nghệ phần mềm, Khoa Công nghệ thông tin – Trường Đại học Sư phạm Kỹ thuật Hưng Yên đã tạo điều kiện thuận lợi cho em thực hiện bài tập lớn môn học này.

Đặc biệt em xin chân thành cảm ơn thầy Đào Anh Hiển đã rất tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện bài tập lớn vừa qua.

Em cũng xin chân thành cảm ơn tất cả các Thầy, các Cô trong Trường đã tận tình giảng dạy, trang bị cho em những kiến thức cần thiết, quý báu để giúp em thực hiện được bài tập lớn này.

Mặc dù em đã có cố gắng, nhưng với trình độ còn hạn chế, trong quá trình thực hiện đề tài không tránh khỏi những thiếu sót. Em hi vọng sẽ nhận được những ý kiến nhận xét, góp ý của các Thầy giáo, Cô giáo về những kết quả triển khai trong bài tập lớn.

Em xin trân trọng cảm ơn!

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 8](#_Toc137015372)

[1.1 Lý do chọn đề tài 8](#_Toc137015373)

[1.2 Mục tiêu của đề tài 9](#_Toc137015374)

[1.2.1 Mục tiêu tổng quát 9](#_Toc137015375)

[1.2.2 Mục tiêu cụ thể 9](#_Toc137015376)

[1.3 Nội dung thực hiện 9](#_Toc137015377)

[1.4 Phương pháp tiếp cận 10](#_Toc137015378)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 11](#_Toc137015379)

[2.1 Phân tích thiết kế hệ thống [1] 11](#_Toc137015380)

[2.1.1 Giới thiệu 11](#_Toc137015381)

[2.1.2 Các mô hình phát triển phần mềm 13](#_Toc137015382)

[2.2 Ngôn ngữ UML [1] 18](#_Toc137015383)

[2.2.1 Giới thiệu 18](#_Toc137015384)

[2.2.2 Đặc điểm 18](#_Toc137015385)

[2.2.3 Các phần tử mô hình trong UML 19](#_Toc137015386)

[2.2.4 Các quan hệ trong UML 21](#_Toc137015387)

[2.2.5 Các biểu đồ trong UML 22](#_Toc137015388)

[2.2.6 UML và quy trình phát triển phần mềm 23](#_Toc137015389)

[2.3 Đặc tả yêu cầu phần mềm [1] 25](#_Toc137015390)

[2.3.1 Mục đích của đặc tả yêu cầu 25](#_Toc137015391)

[2.3.2 Các chế tác mô hình yêu cầu 26](#_Toc137015392)

[2.3.3 Biểu đồ ca sử dụng 28](#_Toc137015393)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ PHẦN MỀM 30](#_Toc137015394)

[3.1 Thông tin khảo sát 30](#_Toc137015395)

[3.1.1 Thông tin chung 30](#_Toc137015396)

[3.1.2 Nội dung khảo sát 30](#_Toc137015397)

[3.2 Mô tả yêu cầu hệ thống 31](#_Toc137015398)

[3.2.1 Yêu cầu chức năng 31](#_Toc137015399)

[3.2.2 Yêu cầu phi chức năng 32](#_Toc137015400)

[3.3 Biểu đồ ca sử dụng [1] 33](#_Toc137015401)

[3.3.1 Đặc tả ca sử dụng 34](#_Toc137015402)

[3.4 Biểu đồ lớp thực thể [1] [2] 47](#_Toc137015403)

[3.4.1 Biểu đồ lớp 47](#_Toc137015404)

[3.4.2 Đặc tả thực thể trong biểu đồ lớp thực thể 47](#_Toc137015405)

[3.5 Biểu đồ tuân tự [1] 50](#_Toc137015406)

[3.5.1 Biểu đồ tuần tự ca sử dụng Đăng nhập 50](#_Toc137015407)

[3.5.2 Biểu đồ tuần tự ca sử dụng Quản lý hợp đồng 51](#_Toc137015408)

[3.5.3 Biểu đồ tuần tự ca sử dụng Quản lý chi tiết hợp đồng 52](#_Toc137015409)

[3.5.4 Biểu đồ tuần tự ca sử dụng Quản lý khách hàng 53](#_Toc137015410)

[3.5.5 Biểu đồ tuần tự ca sử dụng Quản lý mảnh đất 55](#_Toc137015411)

[3.5.6 Biểu đồ tuần tự ca sử dụng Quản lý mảnh đất(sản phẩm của văn phòng) 57](#_Toc137015412)

[3.5.7 Biểu đồ tuần tự ca sử dụng Quản lý tài khoản 59](#_Toc137015413)

[3.5.8 Biểu đồ tuần tự ca sử dụng Thống kê doanh thu 61](#_Toc137015414)

[3.6 Thiết kế CSDL [3] [1] 61](#_Toc137015415)

[3.6.1 Lược đồ CSDL 61](#_Toc137015416)

[3.6.2 Mô tả chi tiết từng bảng trong CSDL 62](#_Toc137015417)

[● KẾT LUẬN 65](#_Toc137015418)

[● TÀI LIỆU THAM KHẢO 67](#_Toc137015419)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Cụm từ tiếng anh** | **Diễn giải** |
| 1 | CSDL |  | Cơ sở dữ liệu |
| 2 | CNPM |  | Công nghệ phần mềm |
| 3 | Winform | Windows Form |  |
| 4 | HD |  | Hợp đồng |
| 5 | CTHD |  | Chi tiết hợp đồng |
| 6 | QL |  | Quản lý |

**DANH MỤC HÌNH ẢNH**

[Hình 3.1 Sơ đồ Use case tổng quát 33](#_Toc137015459)

[Hình 3.2 Sơ đồ use case Quản lý thông tin mảnh đất 34](#_Toc137015460)

[Hình 3.3 Sơ đồ use case Quản lý thông tin mảnh đất 35](#_Toc137015461)

[Hình 3.4 Sơ đồ use case Quản lý khách hàng 38](#_Toc137015462)

[Hình 3.6 Sơ đồ use case Quản lý nhân viên 40](#_Toc137015463)

[Hình 3.9 Sơ đồ use case Quản lý tài khoản 43](#_Toc137015464)

[Hình 3.10 Sơ đồ use case Thống kê doanh thu 45](#_Toc137015465)

[Hình 3.11 Biểu đồ lớp thực thể 47](#_Toc137015466)

[Hình 3.12 Biểu đồ tuần tự tác vụ Đăng nhập 50](#_Toc137015467)

[Hình 3.13 Biểu đồ tuần tự tác vụ hiển thị danh sách hợp đồng 51](#_Toc137015468)

[Hình 3.14 Biểu đồ tuần tự thêm hợp đồng 51](#_Toc137015469)

[Hình 3.15 Biểu đồ tuần tự sửa hợp đồng 51](#_Toc137015470)

[Hình 3.16 Biểu đồ tuần tự xoá hợp đồng 52](#_Toc137015471)

[Hình 3.17 Biểu đồ tuần tự tác vụ hiển thị danh sách chi tiết hợp đồng 52](#_Toc137015472)

[Hình 3.18 Biểu đồ tuần tự thêm chi tiết hợp đồng 52](#_Toc137015473)

[Hình 3.19 Biểu đồ tuần tự sửa chi tiết hợp đồng 53](#_Toc137015474)

[Hình 3.20 Biểu đồ tuần tự xoá chi tiết hợp đồng 53](#_Toc137015475)

[Hình 3.21 Biểu đồ tuần tự tác vụ hiển thị khách hàng 53](#_Toc137015476)

[Hình 3.22 Biểu đồ tuần tự thêm khách hàng 54](#_Toc137015477)

[Hình 3.23 Biểu đồ tuần tự sửa khách hàng 54](#_Toc137015478)

[Hình 3.24 Biểu đồ tuần tự xoá khách hàng 54](#_Toc137015479)

[Hình 3.25 Biểu đồ tuần tự tìm khách hàng theo tên 55](#_Toc137015480)

[Hình 3.27 Biểu đồ tuần tự thêm thông tin mảnh đất 55](#_Toc137015481)

[Hình 3.28 Biểu đồ tuần tự sửa thông tin mảnh đất 56](#_Toc137015482)

[Hình 3.29 Biểu đồ tuần tự xoá thông tin mảnh đất 56](#_Toc137015483)

[Hình 3.30 Biểu đồ tuần tự tìm mảnh đất theo tên 57](#_Toc137015484)

[Hình 3.36 Biểu đồ tuần tự tác vụ hiển thị thông tin mảnh đất 57](#_Toc137015485)

[Hình 3.37 Biểu đồ tuần tự thêm thông tin mảnh đất 57](#_Toc137015486)

[Hình 3.38 Biểu đồ tuần tự sửa thông tin mảnh đất 58](#_Toc137015487)

[Hình 3.39 Biểu đồ tuần tự xoá thông tin mảnh đất 58](#_Toc137015488)

[Hình 3.40 Biểu đồ tuần tự tìm mảnh đất theo tên 58](#_Toc137015489)

[Hình 3.41 Biểu đồ tuần tự tìm mảnh đất theo danh mục mảnh đất 59](#_Toc137015490)

[Hình 3.42 Biểu đồ tuần tự tác vụ hiển thị thông tin tài khoản 59](#_Toc137015491)

[Hình 3.43 Biểu đồ tuần tự thêm thông tin tài khoản 59](#_Toc137015492)

[Hình 3.44 Biểu đồ tuần tự sửa thông tin tài khoản 60](#_Toc137015493)

[Hình 3.45 Biểu đồ tuần tự xoá thông tin tài khoản 60](#_Toc137015494)

[Hình 3.46 Biểu đồ tuần tự tìm tài khoản theo tên đăng nhập 60](#_Toc137015495)

[Hình 3.47 Biểu đồ tuần tự thống kê doanh thu 61](#_Toc137015496)

[Hình 3.48 Diagram CSDL hệ thống 61](#_Toc137015497)

**DANH MỤC BẢNG BIỂU**

[Bảng 3.1 Yêu cầu của khách hàng 30](#_Toc137015498)

[Bảng 3.2 Bảng yêu cầu chức năng của hệ thống 31](#_Toc137015499)

[Bảng 3.3 Bảng yêu cầu phi chức năng của hệ thống 32](#_Toc137015500)

[Bảng 3.4 Bảng danh sách Actor 33](#_Toc137015501)

[Bảng 3.5 Bảng danh sách Use case 33](#_Toc137015502)

[Bảng 3.6 Dòng sự kiện chính Đăng nhập 34](#_Toc137015503)

[Bảng 3.7 Dòng sự kiện chính Use case Quản lý thông tin mảnh đất 35](#_Toc137015504)

[Bảng 3.8 Dòng sự kiện thêm hợp đồng Use case Quản lý thông tin mảnh đất 36](#_Toc137015505)

[Bảng 3.9 Dòng sự kiện sửa hợp đồng Use case Quản lý thông tin mảnh đất 36](#_Toc137015506)

[Bảng 3.10 Dòng sự kiện xoá hợp đồng Use case Quản lý thông tin mảnh đất 36](#_Toc137015507)

[Bảng 3.11 Dòng sự kiện thêm chi tiết hợp đồng Use case Quản lý thông tin mảnh đất 37](#_Toc137015508)

[Bảng 3.12 Dòng sự kiện sửa chi tiết hợp đồng Use case Quản lý thông tin mảnh đất 37](#_Toc137015509)

[Bảng 3.13 Dòng sự kiện xoá chi tiết hợp đồng Use case Quản lý thông tin mảnh đất 37](#_Toc137015510)

[Bảng 3.14 Dòng sự kiện chính khách hàng Use case quản lý khách hàng 38](#_Toc137015511)

[Bảng 3.15 Dòng sự kiện thêm khách hàng Use case quản lý khách hàng 39](#_Toc137015512)

[Bảng 3.16 Dòng sự kiện sửa khách hàng Use case quản lý khách hàng 39](#_Toc137015513)

[Bảng 3.17 Dòng sự kiện xoá khách hàng Use case quản lý khách hàng 39](#_Toc137015514)

[Bảng 3.18 Dòng sự kiện tìm kiếm khách hàng Use case quản lý khách hàng 39](#_Toc137015515)

[Bảng 3.24 Dòng sự kiện chính Use case quản lý nhân viên 40](#_Toc137015516)

[Bảng 3.25 Dòng sự kiện thêm nhân viên 41](#_Toc137015517)

[Bảng 3.26 Dòng sự kiện sửa nhân viên 41](#_Toc137015518)

[Bảng 3.27 Dòng sự kiện xoá nhân viên 42](#_Toc137015519)

[Bảng 3.40 Dòng sự kiện chính Use case Quản lý tài khoản 43](#_Toc137015520)

[Bảng 3.41 Dòng sự kiện thêm tài khoản 43](#_Toc137015521)

[Bảng 3.42 Dòng sự kiện sửa tài khoản 44](#_Toc137015522)

[Bảng 3.43 Dòng sự kiện xoá tài khoản 44](#_Toc137015523)

[Bảng 3.44 Dòng sự kiện tìm tài khoản theo tên tài khoản 44](#_Toc137015524)

[Bảng 3.45 Dòng sự kiện thống kê doanh thu 45](#_Toc137015525)

[Bảng 3.46 Danh sách các thuộc tính lớp NhanVien 47](#_Toc137015526)

[Bảng 3.47 Danh sách các phương thức lớp NhanVien 47](#_Toc137015527)

[Bảng 3.48 Danh sách các thuộc tính lớp TaiKhoan 48](#_Toc137015528)

[Bảng 3.49 Danh sách các phương thức lớp TaiKhoan 48](#_Toc137015529)

[Bảng 3.52 Danh sách các thuộc tính lớp Dat 48](#_Toc137015530)

[Bảng 3.53 Danh sách các phương thức lớp Dat 49](#_Toc137015531)

[Bảng 3.58 Danh sách các thuộc tính lớp KhachHang 49](#_Toc137015532)

[Bảng 3.60 Danh sách các thuộc tính lớp HopDong 49](#_Toc137015533)

[Bảng 3.61 Danh sách các phương thức lớp HopDong 50](#_Toc137015534)

# TỔNG QUAN VỀ ĐỀ TÀI

## Lý do chọn đề tài

* Nhu cầu thực tiễn: Bất động sản, đặc biệt là bán đất thổ cư, là một lĩnh vực có nhu cầu lớn và luôn thu hút sự quan tâm của người dân và doanh nghiệp. Quản lý thông tin và giao dịch bán đất một cách hiệu quả là rất quan trọng để đáp ứng nhu cầu của thị trường.
* Tiềm năng thị trường: Tỉnh Hưng Yên, như nhiều tỉnh thành khác ở Việt Nam, đang trở thành một điểm đến thu hút đầu tư bất động sản. Việc phát triển phần mềm quản lý bán đất thổ cư ở địa phương này có thể giúp tăng cường sự chuyên nghiệp và hiệu quả trong quản lý bất động sản và giao dịch mua bán đất.
* Tính ứng dụng và tiện ích: Phần mềm quản lý bán đất thổ cư cung cấp các tính năng quản lý thông tin khách hàng, đất thổ cư, giao dịch bán đất, tính toán giá bán và tạo hợp đồng. Điều này giúp cho việc quản lý và thực hiện giao dịch trở nên tiện lợi, nhanh chóng và chính xác hơn, đồng thời giảm thiểu các rủi ro liên quan đến thông tin và giao dịch không chính xác.
* Thách thức và cơ hội: Việc xây dựng phần mềm quản lý bán đất thổ cư đòi hỏi kiến thức về quản lý dự án phần mềm, phát triển cơ sở dữ liệu, giao diện người dùng và các công nghệ lập trình hiện đại. Điều này mang đến cơ hội để nghiên cứu và áp dụng các công nghệ mới, từ đó nâng cao kỹ năng và khả năng làm việc của nhóm phát triển phần mềm.
* Đóng góp cho cộng đồng: Phần mềm quản lý bán đất thổ cư có thể giúp cải thiện quy trình quản lý và thực hiện giao dịch bất động sản tại tỉnh Hưng Yên. Điều này góp phần vào sự phát triển kinh tế và xã hội của địa phương, tạo điều kiện thuận lợi cho người dân và doanh nghiệp trong việc giao dịch bất động sản.
* Tóm lại, việc lựa chọn đề tài "Phần mềm quản lý bán đất thổ cư ở tỉnh Hưng Yên" được đưa ra dựa trên nhu cầu thực tế, tiềm năng thị trường, tính ứng dụng và tiện ích của phần mềm, cũng như thách thức và cơ hội mà nó mang lại. Đồng thời, việc phát triển phần mềm này còn đóng góp vào sự phát triển cộng đồng và nâng cao kỹ năng của nhóm phát triển.

## Mục tiêu của đề tài

### Mục tiêu tổng quát

Mục tiêu tổng quát của dự án phần mềm quản lý bán đất thổ cư ở tỉnh Hưng Yên là xây dựng một hệ thống phần mềm chuyên biệt, có khả năng quản lý và giao dịch bán đất thổ cư một cách hiệu quả. Dự án nhằm cung cấp các công cụ và tính năng hỗ trợ cho quá trình quản lý thông tin khách hàng, đất thổ cư, giao dịch bán đất và tạo hợp đồng, từ đó tạo ra một quy trình làm việc chuyên nghiệp và tiện lợi cho người dùng.

### Mục tiêu cụ thể

* Các mục tiêu cụ thể trong dự án có thể bao gồm:
* Quản lý thông tin khách hàng: Xây dựng hệ thống quản lý thông tin khách hàng, bao gồm thông tin cá nhân, thông tin liên hệ và lịch sử giao dịch.
* Quản lý thông tin đất thổ cư: Xây dựng cơ sở dữ liệu chứa thông tin về các lô đất thổ cư, bao gồm vị trí, diện tích, mô tả, giá bán và trạng thái sở hữu.
* Giao dịch bán đất: Cung cấp giao diện và công cụ cho việc thực hiện giao dịch mua bán đất thổ cư, bao gồm tìm kiếm đất, đặt cọc, tạo hợp đồng và thanh toán.
* Tính toán giá bán đất: Xây dựng chức năng tính toán giá bán đất dựa trên các yếu tố như vị trí, diện tích, tiện ích xung quanh và thị trường.
* Quản lý hợp đồng: Tạo hệ thống quản lý hợp đồng mua bán đất, ghi nhận thông tin chi tiết về các hợp đồng, quản lý các giai đoạn thanh toán và thông báo các hạn mức thời gian quan trọng.
* Báo cáo và thống kê: Cung cấp chức năng tạo báo cáo và thống kê về tình hình giao dịch, doanh thu, diện tích đất còn lại và các thông tin liên quan khác.
* Đảm bảo tính bảo mật và phân quyền: Xây dựng hệ thống có tính năng bảo mật cao, bao gồm phân quyền truy cập dữ liệu và các biện pháp bảo vệ thông tin cá nhân của khách hàng.

## Nội dung thực hiện

* Xác định yêu cầu: Tiến hành phân tích và thu thập yêu cầu từ khách hàng, gồm các chức năng và tính năng cần có trong hệ thống quản lý bán đất thổ cư. Xác định yêu cầu về cơ sở dữ liệu, giao diện người dùng, quy trình làm việc và các yêu cầu bảo mật.
* Thiết kế hệ thống: Dựa trên yêu cầu đã xác định, thiết kế cấu trúc cơ sở dữ liệu phù hợp, bao gồm các bảng, quan hệ, và các ràng buộc. Thiết kế giao diện người dùng dễ sử dụng và thân thiện. Xác định quy trình làm việc logic và các chức năng cần có trong hệ thống.
* Xây dựng cơ sở dữ liệu: Triển khai cấu trúc cơ sở dữ liệu dựa trên thiết kế đã hoàn thiện. Tạo các bảng, quan hệ, ràng buộc và chỉ mục cần thiết. Xác định các khóa chính và khóa ngoại để quan hệ giữa các bảng.
* Xây dựng lớp Logic: Xây dựng các lớp và phương thức để thực hiện các chức năng và quy trình logic của hệ thống. Đảm bảo việc truy xuất và xử lý dữ liệu được thực hiện đúng và an toàn.
* Xây dựng giao diện người dùng: Thiết kế và xây dựng giao diện người dùng cho phần mềm quản lý bán đất thổ cư. Đảm bảo giao diện dễ sử dụng, thân thiện và có trải nghiệm người dùng tốt.
* Phát triển chức năng: Xây dựng và triển khai các chức năng và tính năng cần có trong hệ thống quản lý bán đất thổ cư, bao gồm quản lý thông tin khách hàng, quản lý đất thổ cư, giao dịch bán đất, tính toán giá bán, quản lý hợp đồng, báo cáo và thống kê.
* Kiểm thử và sửa lỗi: Tiến hành kiểm thử hệ thống để đảm bảo tính đúng đắn và hoạt động ổn định. Xử lý các lỗi và sự cố phát sinh trong quá trình kiểm thử.
* Triển khai và triển khai: Triển khai hệ thống phần mềm quản lý bán đất thổ cư trên môi trường sản xuất. Đảm bảo hệ thống hoạt động ổn định và đáp ứng yêu cầu của khách hàng.
* Đào tạo và hỗ trợ: Cung cấp đào tạo cho người dùng cuối và nhân viên liên quan về cách sử dụng hệ thống. Đảm bảo sẵn sàng hỗ trợ và giải đáp các thắc mắc, vấn đề sau triển khai.
* Bảo trì và nâng cấp: Thực hiện bảo trì định kỳ và nâng cấp hệ thống để đảm bảo tính bảo mật, hiệu suất và tính khả dụng của phần mềm quản lý bán đất thổ cư.

## Phương pháp tiếp cận

* Phân tích yêu cầu: Tiếp cận từ góc độ khách hàng và các bên liên quan để hiểu rõ yêu cầu và nhu cầu của họ. Thực hiện cuộc trò chuyện, phỏng vấn và khảo sát để xác định các chức năng, tính năng và yêu cầu cần có trong hệ thống.
* Thiết kế hệ thống: Dựa trên yêu cầu đã xác định, tiến hành thiết kế cấu trúc hệ thống, bao gồm cơ sở dữ liệu, giao diện người dùng, logic xử lý và quy trình làm việc. Sử dụng các phương pháp và công cụ thiết kế để tạo ra mô hình và tài liệu thiết kế chi tiết.
* Phát triển phần mềm: Sử dụng mô hình phát triển phần mềm như Agile, Waterfall hoặc Spiral để thực hiện quá trình phát triển. Bắt đầu từ việc xác định các yêu cầu cần thiết, phân chia công việc thành các giai đoạn và tuần tự hoàn thiện từng giai đoạn cho đến khi hoàn thành toàn bộ hệ thống.
* Kiểm thử và sửa lỗi: Tiến hành kiểm thử toàn diện trên hệ thống để đảm bảo tính chính xác, hiệu suất và tính năng của phần mềm. Sử dụng các kỹ thuật kiểm thử như kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử chấp nhận và kiểm thử tải.
* Triển khai và triển khai: Chuẩn bị và triển khai hệ thống phần mềm trên môi trường sản xuất. Đảm bảo rằng hệ thống hoạt động một cách ổn định và đáp ứng được yêu cầu của người dùng.
* Đào tạo và hỗ trợ: Cung cấp đào tạo và hướng dẫn cho người dùng cuối và nhân viên liên quan về cách sử dụng hệ thống. Đảm bảo sẵn sàng hỗ trợ và giải đáp các thắc mắc, vấn đề sau triển khai.
* Bảo trì và nâng cấp: Thực hiện các hoạt động bảo trì định kỳ để đảm bảo tính bảo mật, hiệu suất và tính khả dụng của hệ thống. Tiếp tục nâng cấp và phát triển hệ thống để đáp ứng các yêu cầu và thay đổi mới trong tương lai.

# CƠ SỞ LÝ THUYẾT

## Phân tích thiết kế hệ thống [1]

### Giới thiệu

1. Phần mềm
   * Phần mềm (Software) được coi là một tập hợp gồm các chương trình máy tính thực thi được, các cấu trúc dữ liệu và các tài liệu.
   * Các mảnh đất phần mềm (Software Products) được phát triển có thể được bán cho khác hàng
   * Các mảnh đất có thể được chia làm 2 loại:
     + Các mảnh đất được phát triển cho các thị trường chung và được bán chi nhiều khách hàng khác nhau sử dụng.
     + Các mảnh đất được phát triển theo yêu cầu cụ thể của khách hàng.
2. Kỹ nghệ phần mềm
   * Kỹ nghệ phần mềm (Software Engineering) là việc áp dụng một cách có hệ thống, có kỷ luật, định lượng được để phát triển, vận hàng và bảo trì phần mềm một cách có hiệu quả với các giới hạn cho trước (giới hạn về chi phí, thời gian, công cụ, con người)
   * Kỹ nghệ phần mềm được phân thành các lớp:
     + Quản lý chất lượng (Quality). Bất kỳ cách tiếp cận kỹ nghệ nào (bao gồm kỹ nghệ phần mềm) phải dựa trên cam kết của 1 tổ chức. Ví dụ chất lượng theo tiêu chuẩn ISO
     + Quy trình (Process) kỹ nghệ phần mềm là nền tảng của kỹ nghệ phần mềm, là chất kế dính để tổ chức các tầng kỹ nghệ lại với nhau và cho phép phát triển các phần mềm hợp lý và kịp thời
     + Các phương pháp (Methods) kỹ nghệ phần mềm. Các phương pháp kỹ nghệ cung cấp kỹ nghệ làm thế nào để xây dựng phần mềm. Các phương pháp bao gồm một loạt các nhiệm vụ như giao tiếp, phân tích các yêu cầu, mô hình hoá thiết kế, xây dựng chương chình, thử nghiệm và hỗ trợ
     + Các công cụ (Tools) kỹ nghệ phần mềm cung cấp các hỗ trợ tự động hoặc bán tự động hoá cho các quy trình và các phương pháp kỹ nghệ phần mềm
   * Trong công nghiệp sản xuất phần mềm, kỹ nghệ phần mềm đóng một vai trò rất quan trọng vì:
     + Ngày càng nhiều cá nhân và các tổ chức xã hội hoạt động dựa vào các hệ thống phần mềm.
     + Về lâu dài, khi áp dụng các phương pháp kỹ nghệ phần mềm cho các hệ thống phần mềm chuyên nghiệp sẽ rẻ hơn là chỉ viết chương trình như là một dự án lập trình cá nhân.
3. Quy trình phần mềm
   * Quy trình phần mềm (Software Process) là một tập hợp gồm các hoạt động có cấu trúc nhằm phát triển hoặc tiến hoá một mảnh đất phần mềm.
   * Mặc dù tồn tại nhiều quy trình sản xuất phần mềm khác nhau, nhưng tất cả các quy trình này phải bao gồm 4 hoạt động kỹ nghệ phần mềm cơ bản:
     + Đặc tả phần mềm (Software Specification): hoạt động này được thực hiện bởi các kỹ sư phần mềm làm việc với khách hàng để xác định được chức năng của phần mềm, các nghiệp vụ và các ràng buộc trên các chức năng của phần mềm
     + Phát triển phần mềm (Software Development): kỹ sư phần mềm thực hiện thiết kế và lập trình phần mềm theo bản đặc tả của phần mềm
     + Thẩm định phần mềm (Software Validation): hoạt động kiểm tra phần mềm để đảm bảo rằng phần mềm được phát triển đáp ứng đúng yêu cầu của khách hàng
     + Tiến hoá phần mềm (Software Evolution): khi phần mềm được sửa đổi để đáp ứng sự thay đổi yêu cầu của khách hàng và yêu cầu của thị trường

### Các mô hình phát triển phần mềm

* Các mô hình (quy trình) phát triển phần mềm còn được gọi là mô hình vòng đời phát triển phần mềm (Software Development Life Cycle – SDLLC model) là mô tả trừu tượng, mức cao của các quy trình phần mềm có thể được sử dụng để giải thích các cách tiếp cận khách nhau để phát triển phần mềm.

1. Mô hình thác nước
   * Mô hình thác nước (Waterfall Model) gồm 5 pha được thực hiện một cách tuần tự như là dòng chảy thác nước. Đây là mô hình đầu tiên thể hiện quy trình phát triển phần mềm có kế hoạch
   * Các giai đoạn mô hình thác nước phản ánh trực tiếp các hoạt động phát triển phần mềm cơ bản gồm :
     + Định nghĩa và phần tích yêu cầu (Requirements analysis and definition): xác định những “yêu cầu”(“What”) liên quan đến chức năng và phi chức năng mà phần mềm cần có
     + Thiết kế hệ thống và phần mềm (System and software design): phân bổ các yêu cầu phần cứng hoặc phần mềm. Thiết kế phần mềm sẽ xác định “làm thế nào”(“How”) để hẹ thống phần mềm đáp ứng nhúng “yêu cầu”(“What”)
     + Triển khai cài đặt và kiểm thử đơn vị (Implementation and unit testing): hiện thực “làm thế nào”(“How”) thành mã nguồn triển khai và thực hiện kiểm thử mức đơn vị để xác minh rằng mỗi đơn vị mã nguồn đáp ứng được đặc tả của nó
     + Kiểm thử tích hợp và kiểm thử hệ thống(Integration and system testing): kiểm thử mã nguồn mở được triển khai. Đầu tiên là kiểm thử đơn vị → kiểm thử tích hợp → kiểm thử → kiểm thử chấp nhận. Sau khi kiểm thử toàn diện hệ thống, phần mềm được chuyển giao cho khách hàng
     + Vận hành và bảo trì (Operation and maintenance) : sửa chữa lỗi không được phát hiện ở các giai đoạn trước của vòng đời và phát triển những thay đổi mới được khách hàng yêu cầu
   * Đặc điểm của mô hình:
     + Các pha tách biệt với nhau và được thực hiện 1 cách tuần tự
     + Chậm có phiên bản thực hiện được
   * Mô hình thác nước phù hợp với các dự án nhỏ và được yêu cầu rất rõ ràng, chi tiết và hầu như không thay đổi, thường xuất phát từ mảnh đất đã đạt mức ổn định và hầu như không có rủi ro
2. Mô hình chữ V
   * Mô hình chữ V (V model) là một mô hình cải tiến từ mô hình thác nước. Trong đó, toàn bộ quy trình được chia thành 2 nhóm tương ứng nhau: phát triển và kiểm thử
   * Ở mọi pha, kế hoạch kiểm thử và các ca kiểm thử được tạo để xác minh và thẩm định mảnh đất phần mềm theo yêu cầu của các pha tương ứng. Hoạt động xác minh và thẩm định tiến hành song song nhau và song song từ đầu chu trình cùng với các hoạt động phát triển phần mềm.
   * Mô hình chữ V còn được gọi là mô hình xác minh và thẩm định (Verification and Validation model).
3. Mô hình tăng trưởng
   * Mô hình tăng trưởng (Incremental Model) chia nhỏ mảnh đất phần mềm thành từng phần tăng theo yêu cầu chức năng, hệ thống phần mềm được chia thành từng bản khác nhau, từ một chu kỳ lớn sẽ được phân chia thành nhiều chu kỳ nhỏ ứng với từng bản, và do đó ta có được một đa chu kỳ phát triển.
   * Phát triển tăng trưởng có ba lợi thế lớn so với mô hình thác nước:
     + Chi phí thực hiện thay đổi yêu cầu được giảm. Diện tích tài liệu phân tích ít hơn đáng kể so với mô hình thác nước
     + Dễ dàng hơn để lấy phản hồi của khách hàng về các công việc đã được thực hiện. Khách hàng có thể cho ý kiến trên các thuyết minh của phần mềm
     + Có thể chuyển giao và triển khai phần mềm hữu ích tới khách hàng, ngay cả khi tất cả các chức năng chưa được đưa vào. Khách hàng có thể sửa phần mềm sớm hơn so với mô hình thác nước
   * Nhược điểm:
     + Quy trình không nhìn thấy được. Các nhà quản lý cần chuyển giao đều đặn để đo lường tiến độ. Nếu các hệ thống được phát triển nhanh, việc cung cấp các tài liệu mà phản ánh mọi phiên bản của hệ thống sẽ không hiệu quả về mặt chi phí.
     + Cấu trúc hệ thống có xu hướng giảm khi các tăng trưởng mới được thêm vào.
     + Việc thay đổi thường xuyên dẫn đến mã nguồn lộn xộn vì các chức năng mới được thêm vào bằng mọi cách có thể.
     + Việc thêm các tính năng mới vào hệ thống ngày càng khó khăn và tốn kém.
4. Mô hình xoắn ốc
   * Mô hình xoắn ốc là sự kết hợp giữa mô hình lặp với một trong các mô hình vòng đời phát triển phần mềm. Mô hình xoắc ốc với các giai đoạn lặp theo chu kỳ xoay vòng, trong đó mỗi chu kỳ bao gồm 4 pha:
     + Objective identification- Thiết lập mục tiêu: xác định mục tiêu, đối tượng cho từng pha của dự án.
     + Alternate evaluation- Đánh giá và giảm thiểu rủi ro: đánh giá rủi ro và thực hiện các hành động để giảm thiểu rủi ro.
     + Product development- Phát triển mảnh đất: Lựa chọn mô hình phù hợp để phát triển hệ thống.
     + Next phase planning- Lập kế hoạch: đánh giá dự án và lập kế hoạch cho pha tiếp theo.
   * Ưu điểm:
     + Trong mô hình xoắn ốc, quản lý rủi ro được đẩy lên như một phần thiết yếu trong mỗi “spiral” để tăng độ tin cậy của dự án
     + Mô hình này kết hợp những tính chất tốt nhất của mô hình thác nước và mô hình tiến hóa
     + Mô hình cho phép thay đổi tùy theo điều kiện thực tế dự án tại mỗi "spiral"
     + Mô hình được coi là mô hình tổng quát nhất
   * Nhược điểm:
     + Phức tạp và không phù hợp cho dự án nhỏ với ít rủi ro
     + Cần có kỹ năng tốt về phân tích rủi ro
   * Ứng dụng:
     + Dự án lớn có nhiều rủi ro, sự thành công của dự án không có sự đảm bảo nhất định
     + Những dự án đòi hỏi nhiều tính toán
     + Đội ngũ thực hiện dự án phải có năng lực phân tích rủi ro
5. Mô hình Agile
   * Mô hình Agile SDLC là sự kết hợp của các mô hình quy trình lập và tăng trưởng tập trung vào khả năng thích ứng của quy trình và sự hài lòng của khách hàng bằng cách phân phối nhanh chóng các mảnh đất phần mềm đang hoạt động
   * Agile là một triết lý hoặc là một khung tư duy để thích ứng và phản hồi nhanh chóng với các thay đổi để đạt được thành công trong một môi trường yêu cầu liên tục thay đổi và không chắc chắn.
   * Phát triển phần mềm Agile (Agile Software Development) là một thuật ngữ chung chỉ tất cả các kỹ thuật và phương pháp phát triển phần mềm theo triết ly Agile. Triết lý Agile được mô tả trong bản tuyên ngôn của Agile thông qua những giá trị cốt lỗi và các nguyên tắc phổ biến rộng rãi
   * 12 nguyên lý trong tuyên ngôn Agile:

* Sự hài lòng của khách hàng
* Chào mừng thay đổi
* Chuyển giao phần mềm hoạt động được
* Hợp tác
* Động lực
* Đối thoại trực tiếp
* Đo lường tiến độ theo phần mềm hoạt động
* Duy trì nhịp độ không đổi
* Giám sát
* Sự đơn giản
* Các nhóm tự tổ chức
* Đánh giá công việc thường xuyên
  + Các phương pháp Agile :
    - Scrum
    - Kanban
    - Scrumban
    - Lean Software Development
    - XP
  + Ưu điểm :
    - Agile rất phù hợp với những dự án nhỏ, các yêu cầu thường không được xác định rõ ràng hay thay đổi thường xuyên
    - Khách hàng có thể xem trước được từng phần dự án trong suốt quá trình phát triển vì Agile phát triển phần mềm theo hướng tăng dần.
    - Agile chia dự án thành những phần nhỏ và giao cho mỗi thành viên.
    - Tỉ lệ thành công của các dự án sử dụng Agile thưởng cao hơn các quy trình khác.
  + Nhược điểm :
    - Thiếu sự nhấn mạnh về thiết kế và tài liệu cần thiết
    - Quy mô nhân lực thưởng giới hạn từ 7 đến 10 người, sẽ có trở ngại lớn nếu nguồn nhân lực yêu cầu vượt qua con số này ví dụ trong các cuộc họp trao đổi.
    - Diện tích yêu cầu có thể nhiều và khó quản lý nếu như nó bao gồm nhiều khía cạnh khác nhau về dự án.
    - Diện tích nhân lực càng tăng, chất lượng càng khó kiểm soát hơn. Việc kiểm tra mã thường xuyên và thiết lập các chỉ tiêu đánh giá năng lực của lập trình viên cho phép giảm thiểu nhược điểm này.
  + Khi nào sử dụng mô hình Agile :
    - Khi yêu cầu thường xuyên thay đổi.
    - Chi phi cho việc thực hiện các thay đổi là rất ít.
    - Dự án mà các yêu cầu mới không mất quá nhiều thời gian.
    - Khi khách hàng luôn sẵn sàng họp với nhóm phần mềm mọi lúc.
    - Khi hoạt động phát triển phần mềm có sự linh động, thoải mái

## Ngôn ngữ UML [1]

### Giới thiệu

* Ngôn ngữ mô hình hoá thống nhất UML(Unified Modeling Language) là mô hình hóa đồ họa mục đích chung với mục đích cung cấp một ngôn ngữ đồ họa đơn giản dễ hiểu và sử dụng nó. Thực tế, UML không chỉ được dùng để mô hình hóa hệ thống phần mềm mà nó còn được dùng để mô hình các hệ thống không phải là phần mềm như luồng công việc hay cho các hoạt động thiết kế phần cứng
* Lịch sử phát triển: UML được phát triển từ nỗ lực đơn giản hóa và hợp nhất của phương pháp OTM của James Rumbaugh. Phiên bản đề xuất về UML được gửi lên tổ chức OMG vào năm 1997. Vào tháng 11 năm 1997, OMG đã thông qua đánh dấu sự ra đời của ngôn ngữ UML 1.0 vào tháng 11 năm 1997. Năm 2004, UML phiên bản 2.0 đã ra đời. Năm 2015 là phiên bản 2.5, và cho đến nay UML vẫn đang được phát triển

### Đặc điểm

1. UML là ngôn ngữ mô hình hoá

* UML là ngôn ngữ mô hình hóa đồ họa để diễn đạt các mô hình hệ thống. Nó bao gồm một tập các từ vừng (concepts) được đại diện bởi các ký hiệu (notation), những ký hiệu này được sử dụng trong các mô hình của UML và một tập các quy tắc chỉ cách sử dụng và kết hợp các ký hiệu.
* Các quy tắc này bao gồm:
  + Cú pháp (syntactic) bao gồm có cú pháp trừu tượng và cú pháp cụ thể cho biết hình dạng các biểu tượng và cách kết hợp chúng trong ngôn ngữ.
  + Ngữ nghĩa (semantic) cho biết ý nghĩa của mỗi biểu tượng, chúng được hiểu thế nào khi nằm trong hoặc không nằm trong ngữ cảnh của các biểu tượng khác.
  + Pragmatic định nghĩa ý nghĩa của biểu tượng để sao cho mục đích của mô hình được thể hiện và mọi người có thể hiểu được.

1. UML là ngôn ngữ trực quan để hiển thị

UML sẽ cung cấp một ngôn ngữ mô hình hóa các ý tưởng, giải pháp của họ bằng các mô hình đồ họa. Các mô hình này giúp diễn đạt các ý tưởng, các giải pháp, các vấn đề của người dùng cũng như nhà phát triển để cho phép tất cả các bên liên quan có thể hiểu thống nhất một vấn đề mà không gặp rào cản về mặt ngôn ngữ tự nhiên trình bày. UML là ngôn ngữ để đặc tả

1. UML là ngôn ngữ để đặc tả

Mục đích của UML là để hỗ trợ mô tả các vấn đề trong phát triển phần mềm một cách chính xác không nhập nhằng và hoàn thiện. UML phục vụ cho việc đặc tả các mô hình trong quá trình phân tích, thiết kế và các giải pháp cài đặt trong vòng đời phát triển của hệ thống.

1. UML là ngôn ngữ để xây dựng

Các biểu đồ UML không chỉ được sử dụng như tài liệu để phát triển và các bên liên quan có thể đọc và hiểu được mà chúng còn được sử dụng như là đầu vào để sinh tự động sang cá dạng biểu diễn khác phục vụ cho quá trình phát triển đặc biệt là chuyển sang mã chương trình.

1. UML là ngôn ngữ để làm tài liệu

UML được sử dụng để hỗ trợ xây dựng các mô hình đặc tả yêu cầu, thiết kế kiến trúc, thiết kế hệ thống, thiết kế đối tượng , yêu cầu kiểm thử,…Cuối cùng UML cung cấp ngôn ngữ để mô hình hóa các hoạt động lập kế hoạch dự án cũng như quản lý chuyển giao các phiên bản

### Các phần tử mô hình trong UML

1. Phần tử cấu trúc
   * Phần tử cấu trúc là các danh từ trong mô hình UML. Chúng là các bộ phận tĩnh của mô hình để nắm bắt các thực thể khái niệm hay vật lý.
   * Có 7 loại phần tử cấu trúc:

* Lớp (Class): Là phần tử mô tả tập các đối tượng cùng chung thuộc tính, thao tác, quan hệ và ngữ nghĩa .
* Giao diện (Interface): Giao diện là tập hợp các thao tác đặc tả dịch vụ của một lớp hay thành phần làm dịch vụ của lớp hay thành phần. Vì vậy giao diện mô tả các hành vi của một lớp hoặc một thành phần có thể quan sát được bên ngoài.
* Phần tử cộng tác (Collaboration): Phần tử cộng tác đặc tả sự tương tác các đối tượng, biểu diễn một ngữ cảnh bao gồm các vai trò và các phần tương tác với nhau để cung cấp hành vi cộng tác phức tạp lớn hơn.
* Ca sử dụng (Use case): Ca sử dụng biểu diễn chuỗi các hành động mà hệ thống thực hiện để đạt được một kết quả có thể quan sát được cho tác nhân nào đó.
* Tác nhân (Actor): Một tác nhận được định nghĩa là một thực thể bên trong hoặc bên ngoài hệ thống. Tác nhân có thể là người, là thiết bị hoặc là một hệ thống đứng bên ngoài hệ thống tương tác với hệ thống.
* Lớp tích cực (Active class): Là lớp mà các đối tượng thể hiện của nó quản lý một hay nhiều tiến trình hay luồng xử lý.
* Thành phần (Component): Thành phần biểu diễn một phần vật lý của hệ thống. Các thành phần như các tệp mã nguồn, các tệp nhị phân là những thành phần hay gặp trong quá trình phát triển hệ thống.
* Nút (Node): Nút thể hiện phần tử vật lý, tồn tại khi chương trình chạy và biểu diễn các tài nguyên tính toàn cùng với các năng lực lưu trữ và xử lý. Phần tử hành vi
* Phần tử hành vi là bộ phận của mô hình UML. Các phần tử này các động từ của mô hình biểu diễn các hành vi theo thời gian và không gian.
* Có hai loại phần tử hành vi chính là tương tác và trạng thái:
  + Tương tác: Phần tử tương tác biểu diễn các hành vi bằng cách mô tả quá trình trai đổi các thông điệp giữa các đối tượng trong một ngữ cảnh cụ thể để thể hiện một mục đích cụ thể nào đó.
  + Trạng thái: Máy trạng thái là hành vi biểu diễn chuỗi các trạng thái của một đối tượng hay của một tương tác sẽ đi qua trong vòng đời của chúng để đáp ứng các sự kiện.

1. Phần tử nhóm

* Phần tử nhóm được định nghĩa như các cơ chế để nhóm các phần tử của một mô hình với nhau. Nó cung cấp cơ chế để các mô hình có thể được phân chia cũng như gộp nhóm. Chỉ có một phần tử thuộc nhóm này là gói (package)
* Gói là cơ chết đa năng để tổ chức các phần tử vào nhóm. Gói giúp chúng ta có cái nhìn khái quát về hệ thống, giúp chúng ta dễ quản lý các chế tác.

1. Phần tử chú thích

* Phần tử chú thích (note) được định nghĩa như một cơ chế để ghi lại các chú thích, các mô tả và các nhận xét của các phần tử mô hình UML ở dạng văn bản. Đó là lời giải thích áp dụng để mô tả các phần tử khác của mô hình.

### Các quan hệ trong UML

* UML có bốn loại quan hệ là quan hệ phụ thuộc, kết hợp, khái quát hóa và hiện thực hóa. Chúng là cơ sở để xây dựng mọi quan hệ trong UML.
  + Phụ thuộc (Dependency): Phụ thuộc là quan hệ ngữ nghĩa giữ hai phần tử trong đó khi thay đổi phần tử này (phần tử độc lập) có thể ảnh hưởng ngữ nghĩa của từ còn lại (phần từ độc lập) có thể ảnh hưởng đến ngữ nghĩa của phần tử còn lại (phần tử phụ thuộc). Ký pháp đồ họa cho quan hệ phụ thuộc là đoạn thẳng nét đứt, có thể có hướng từ phần tử phụ thuộc đến phần tử độc lập
  + Kết hợp (Association): Kết hợp là quan hệ cấu trúc để mô tả tập các liên kết (links), mỗi liên kết biểu diễn sự kết nối giữa hai đối tượng. Khi đối tượng của lớp này gửi/nhận thông điệp đến/từ đối tượng của lớp kia thì ta gọi là có quan hệ kết hợp. Ký pháp kết hợp là đoạn thẳng có thể có hướng thường có nhãn là tên quan hệ và lực lượng quan hệ (multiplicity) cũng như vai trò tham gia quan hệ (roles).
    - Tụ hợp (aggregation) là dạng đặc biệt của kết hợp, quan hệ này diễn đạt quan hệ cấu trúc giữa toán thể và các bộ phận của nó. Để biểu diễn quan hệ hợp thành, ký pháp kết hợp được bổ sung thêm ở phía tổng thể một hình tráng trắng
    - Một dạng đặc biệt của tụ hợp là quan hệ hợp thành (composition), trong đó nếu như đối tượng toàn thể bị hủy bỏ thì các đối tượng bộ phận của nó cũng bị hủy bỏ theo. Để biểu diễn quan hệ tụ hợp, ký pháp kết hợp được bổ sung thêm ở phía tổng thể một hình tráng đen
  + Tổng quát hóa (Generalization): Tổng quát hóa là loại quan hệ đặc biệt hóa/khái quát hóa, trong đó đối tượng đặc biệt hóa (con) sẽ kế thừa các thuộc tính và phương pháp của đối tượng tổng quát (cha).
  + Hiện thực hóa (Realization): Hiện thực hóa là quan hệ nghĩa giữa các phân lớp (classifier), trong đó một phân lớp sẽ đặc tả cam kết mà phân lớp kia phải đảm bảo đáp ứng thực thi.

### Các biểu đồ trong UML

Biểu đồ (diagram) là sự biểu diễn trực quan đồ họa của một tập các phân tử ở dạng đồ thị, trong đó các đỉnh ứng với các phân tử và các ứng với các quan hệ. Các biểu đồ được sử dụng để trực quan hóa hệ thống từ một khung nhìn khía cạnh nào đó, nó chính là một phép chiếu lên hệ thống. Trong các hệ thống, biểu đồ tái hiện một khung nhìn về các phần tử tạo nên hệ thống. Một phần tử thường xuất hiện trong một số biểu đồ, cũng có thể là tất cả các biểu đồ. Về mặt lý thuyết, một số biểu đồ có thể chứa bất kì sự kết hợp nào của phần tử và quan hệ. Tuy nhiên trong thực tế, chỉ một số kiểu kết hợp nào càn thiết cho việc phát triển hệ thống.

Trong UML có 14 kiểu biểu đồ là:

* Các biểu đồ mô hình hoá cấu trúc (Structural Modelling Diagrams):
  + Biểu đồ lớp (Class Diagram) biểu diễn các lớp và quan hệ giữa các lớp. Đây là biểu đồ mô phỏng khía cạnh tĩnh của hệ thống.
  + Biểu đồ thành phần (Component Diagrams) biểu diễn các thành phần mềm và quan hệ giữa các thành phần mềm của hệ thống.
  + Biểu đồ triển khai (Deployment Diagrams) biểu diễn các nút triển khai trong hệ thống và các kết nối giữa các nút triển.
  + Biểu đồ đối tượng (Object Diagrams) biểu diễn các đối tượng là thể hiện của các lớp và liên kết giữa các đối tượng.
  + Biểu đồ gói (Package Diagrams) biểu diễn các gói và quan hệ giữa các gói phần mềm trong hệ thống.
  + Biểu đồ hồ sơ (Profile Diagrams) cung cấp sơ chế mở rộng chung để tùy chỉnh mô hình UML cho các miền và nền tảng cụ thể.
* Các biểu đồ mô hình hoá hành vi (Behavioral Modeling Diagrams):
  + Biểu đồ ca sử dụng (Use Case Diagrams) biểu diễn các yêu cầu chức năng của hệ thống.
  + Biểu đồ hoạt động (Activity Diagrams) để mô tả các khía cạnh động của hệ thống. Biểu đồ hoạt động về cơ cản là một phiên bản nâng cao của sơ đồ luông từ hoạt động này sang hoạt động khác.
  + Biểu đồ máy trạng thái (State Machine Diagrams) để mô tả các trạng thái và các chuyển đổi giữa các trạng thái.
  + Biểu đồ tuần tự (Sequence Diagrams) biểu diễn các tương tác giữa các đối tượng theo trình tự thời gian.
  + Biểu đồ cộng tác (Communication Diagrams) biểu diễn các tương tác giữa các đối tượng không quan tâm tới trình tự thời gian.
  + Biểu đồ tổng quan tương tác (Integration Overview Diagrams) cung cấp một mô hình tương tác ở mức độ trừu tượng cao.
  + Biểu đồ cấu trúc tổng hợp (Composite Structural Diagrams) chứa các lớp, giao diện, gói và mối quan hệ của chúng và cung cấp một cái nhìn logic về tất cả hoặc một phần của hệ thống phần.
  + Biểu đồ thời giang (Timing Diagram) biểu diễn các tương tác khi mục đích chính của biểu đồ là lý do về thời gian

### UML và quy trình phát triển phần mềm

1. Giai đoạn nghiên cứu sơ bộ

Trong giai đoạn nghiên cứu sơ bộ, UML đưa ra khái niệm ca sử dụng(use case) để nắm bắt các yêu cầu của khách hàng và sử dụng biểu đồ Use case (Use Case Diagram) để nêu bật mối quan hệ cũng như sự giao tiếp với hệ thống.

Qua phương pháp mô hình hóa Use case, các tác nhân (Actor) bên ngoài quan tâm đến hệ thống sẽ được mô hình hóa song song với chức năng mà họ đòi hỏi từ phía hệ thống (tức là Use case).

Các tác nhân và các Use case được mô hình hóa cùng các mối quan hệ và được miêu tả trong biểu đồ Use case của UML

Mỗi một Use case được mô tả trong tài liệu, và nó sẽ đặc tả các yêu cầu của khách hàng: Anh ta hay chị ta chờ đợi điều gì ở phía hệ thống mà không hề để ý đến việc chức năng này sẽ được thực thi ra sao.

1. Giai đoạn phân tích

Giai đoạn phân tích quan tâm đến quá trình trừu tượng hóa đầu tiên (các lớp và các đối tượng) cũng như cơ chế hiện hữu trong phạm vi vấn đề.

Sau khi nhà phân tích đã nhận biết được các lớp thành phần của mô hình cũng như mối quan hệ giữa chúng với nhau, các lớp cùng các mối quan hệ đó sẽ được miêu tả bằng công cụ biểu đồ lớp (class diagram) của UML.

Trong giai đoạn phân tích, chỉ duy nhất các lớp có tồn tại trong phạm vi vấn đề (các khái niệm đời thực) là được mô hình hóa. Các lớp kỹ thuật định nghĩa chi tiết cũng như giải pháp trong hệ thống phần mềm, ví dụ như các lớp cho giao diện người dùng, cho ngân hàng dữ liệu, cho sự giao tiếp, trùng hợp, v.v…, chưa phải là mối quan tâm của giai đoạn này.

1. Giai đoạn thiết kế

Trong giai đoạn này, kết quả của giai đoạn phân tích sẽ được mở rộng thành một giải pháp kỹ thuật như mô hình thiết kế hệ thống, mô hình kiến trúc, các mô hình thiết kế chi tiết,… Các mô hình chi tiết này cung cấp hạ tầng kỹ thuật như giao diện, nền tảng cho xây dựng CSDL,..

Kết quả của giai đoạn thiết kế là các đặc tả đủ chi tiết để chuyển sang mã nguồn trong giai đoạn xây dựng phần mềm.

Hoạt động phân tích và thiết kế thường lấy kiến trúc làm trung tâm. Kiến trúc được thể hiện thông qua một số khung nhìn kiến trúc. Mỗi khung nhìn chứa các quyết định thiết kế chính.

Kiến trúc phần mềm bao gồm một tập các quyết định thiết yếu về tổ chức hệ thống phần mềm phản ánh các khía cạnh về cấu trúc, hành vi và cách tổ chức phân cấp trong hệ thống. Kiến trúc phần mềm được tạo từ công thức : Kiến trúc = Các phần tử + Các khuôn dạng + Cơ sở hợp lý. Trong đó cơ sở hợp lý được sử dụng để đánh giá một kiến trúc phần mềm tốt.

Rational đã đưa ra mô hình kiên trúc “*Khung nhìn 4+1*”. Một khung nhìn kiến trúc là mô tả được đơn giản hóa về hệ thống phản ánh mối quan tâm liên quan. Các khung nhìn là các lát cắt của các mô hình.

* *Khung nhìn ca sử dụng (use case view)* : Là khung nhìn đầu tiên và quan trong quá trình phân tích thiết kế. Khung nhìn này biểu diễn hệ thống dưới dạng các chức năng theo hướng nhìn của người dùng cuối, nó chi phối tất cả các khung nhìn khác.
* *Khung nhìn logic (Logical view)* : Biểu diễn tổ chức tổ chức và quan hệ của các lớp trong hệ thống để hình thành nên cấu trúc logic của hệ thống. Khung nhìn này bao gồm các biểu đồ như đối tương, lớp, gói, …
* *Khung nhìn thực thi (Implementation view)* : Phản ánh hệ thống như là một khối được lắp ghép từ các thành phần.
* *Khung nhìn triển khai (Deployment view)* : Tập trung phân bổ tài nguyên vật lý cùng với những trách nhiệm đi kèm.
* *Khung nhìn tiến trình (Process view)* : Biểu diễn phân tách các luồng thực hiện chương trình và sự đồng bộ hóa giữa các luồng, đối tượng tham gia và trách nhiệm thực hiện trong các luồng.

1. Giai đoạn thực thi

Trong giai đoạn thực thi (giai đoạn lập trình), các lớp của giai đoạn thiết kế sẽ được biến thành những dòng code cụ thể trong một ngôn ngữ lập trình hướng đối tượng cụ thể (không nên dùng một ngôn ngữ lập trình hướng chức năng).

Phụ thuộc vào khả năng của ngôn ngữ được sử dụng, đây có thể là một công việc khó khăn hay dễ dàng. Khi tạo ra các mô hình phân tích và thiết kế trong UML, tốt nhất nên cố gắng né tránh việc ngay lập tức biến đổi các mô hình này thành các dòng code.

Trong những giai đoạn trước, mô hình được sử dụng để dễ hiểu, dễ giao tiếp và tạo nên cấu trúc của hệ thống; vì vậy, vội vàng đưa ra những kết luận về việc viết code có thể sẽ thành một trở ngại cho việc tạo ra các mô hình chính xác và đơn giản. Giai đoạn xây dựng là một giai đoạn riêng biệt, nơi các mô hình được chuyển thành code.

1. Giai đoạn kiểm thử

Giai đoạn kiểm thử (Testing) sẽ sử dụng các biểu đô UML trong các giai đoạn trước là đầu vào để thực hiện các hoạt động kiểm thử phần mềm trong các mức kiểm thử khác nhau :

* *Kiểm thử đơn vị (unit testing)*
* *Kiểm thử tích hợp (integration testing)*
* *Kiểm thử hệ thống (system testing)*
* *Kiểm thử chấp nhận (acceptance testing)*

## Đặc tả yêu cầu phần mềm [1]

### Mục đích của đặc tả yêu cầu

Hoạt động đặc tả yêu cầu phần mềm là hoạt động chuyển từ yêu cầu người dùng thành yêu cầu của phần mềm. Hoạt động này được diễn ra ngay sau khi hoạt động mô hình hóa nghiệp vụ và trước hoạt động phân tích và thiết kế.

Hoạt động đặc tả yêu cầu giúp nắm bắt được các yêu cầu phần mềm với những mục đích chính như sau:

* Nhằm thiết lập và duy trì những thỏa thuận giữa khách hàng và các bên liên quan về những gì mà hệ thống sẽ được phát triển.
* Giúp nhà phát triển hiểu tốt hơn về các yêu cầu phần mềm.
* Xác định biên của hệ thống.
* Cung cấp một cơ sở cho việc lập kế hoạch cho những nội dung kỹ thuật trong các vòng lặp của quá trình phát triển.
* Là cơ sở ở cho việc đánh giá và ước lượng chi phí và thời gian phát triển hệ thống.
* Định hình giao diện tương tác giữa người dùng và hệ thống, qua đó làm rõ được nhu cầu và mục tiêu của người dùng với hệ thống cần phát triển.

Đầu ra của hoạt động đặc tả yêu cầu sẽ là đầu vào cho hoạt động phân tích và thiết kế. Các sai sót trong các chế tác đầu ra của giai đoạn này có thể được phát hiện trong giai đoạn phân tích và thiết kế, từ đó các chế tác yêu cầu được cập nhật thay đổi.

### Các chế tác mô hình yêu cầu

1. Phát biểu yêu cầu

Phát biểu vấn đề là một phần của tài liệu tổng thể hệ thống. Tài liệu này sẽ làm rõ hiện trạng của hệ thống hiện thời, đặc biệt là các vấn đề khó khăn đặt ra với hệ thống hiện thời. Từ đó, phát biểu nêu ra được những lý do và động cơ chính để phát triển hệ thống mới. Tài liệu phát biểu vấn đề cũng nêu ra các ràng buộc khác nhau cho việc phát triển hệ thống mới như các ràng buộc về kỹ thuật, thời gian phát triển hay nguồn lực.

1. Từ điển thuật ngữ

Tài liệu từ điển thuật ngữ xác định các thuật ngữ quan trọng được sử dụng trong dự án. Các thuật ngữ này được mô tả ở dạng văn bản và được dùng chung cho tất cả các mô hình của hệ thống.

Từ điển thuật ngữ được xây dựng trong giai đoạn đầu của dự án, giai đoạn đòi hỏi sự thống nhất về các thuật ngữ được sử dụng. Trong pha khởi đầu (inception) và xây dựng (elaboration), các chuyên gia miền sẽ sử dụng các thuật ngữ trong từ điển này để giải thích các thuật ngữ chuyên ngành lĩnh vực của họ.

1. Đặc tả bổ sung

Những yêu cầu không được đề cập đến trong mô hình ca sử dụng sẽ được đặc tả trong tài liệu đặc tả bổ sung. Những yêu cầu này là bổ sung quan trọng cho mô hình ca sử dụng.

Các yêu cầu chức năng và phi chức năng trong tài liệu bổ sung thường là những ràng buộc đối với việc thực hiện như:

* Ràng buộc chức năng (functionality): Danh sách các yêu cầu chức năng được dùng chung và tham chiếu đến từ ca sử dụng.
* Khả năng sử dụng (usability): Các yêu cầu liên quan hoặc ảnh hưởng đến khả năng sử dụng của hệ thống. Ví dụ các yêu cầu về tính dễ sử dụng hoặc các yêu cầu về đào tạo người dùng, trong đó chỉ rõ mức độ sẵn sàng của hệ thống được các tác nhân của nó sử dụng.
* Độ tin cậy (reliability): Các yêu cầu liên quan đến độ tin cậy của hệ thống. Yêu cầu này thường nêu các yêu cầu về độ đo, sự lượng hóa thời gian trung bình giữa các lần thất bại hay số lỗi trên mỗi nghìn dòng lệnh. Hiệu suất (performance): Các tính chất hiệu suất hoạt động của hệ thống như thời gian phản hồi cụ the.
* Khả năng hỗ trợ (supportability): Các yêu cầu nâng cao khả năng hỗ trợ hoặc khả năng bảo trì của hệ thống.
* Các ràng buộc thiết kế (design constraints): Các yêu cầu về ràng buộc thiết kế trên hệ thống đang được xây dựng.

1. Mô hình ca sử dụng

Khi bắt đầu xây dựng một mảnh đất phần mềm, nhóm phát triển phải xác định các chức năng mà hệ thống cần phải thực hiện là gì. Biểu đồ ca sử dụng trong UML được sử dụng để mô hình hóa các chức năng sẽ được xây dựng và các chức năng được cung cấp bởi hệ thống đó.

Một biểu đồ ca sử dụng UML là dạng chính của yêu cầu hệ thống/phần mềm của một chương trình phần mềm mới được phát triển. Các ca sử dụng xác định hành vi mong đợi (what). Các ca sử dụng sau khi được xác định có thể được biểu thị bằng cả văn bản và biểu diễn trực quan bằng biểu đồ ca sử dụng.

Trong đó mỗi ca sử dụng mô tả chi tiết về những gì mà hệ thống làm trong quá trình tương tác với tác nhân. Nó phản ánh chuỗi sự kiện diễn ra trong từng kịch bản ca sử dụng. Những mô tả này được xây dựng thành tài liệu đặc tả ca sử dụng (use case specification).

Mô hình ca sử dụng mô tả hướng nhìn chức năng của hệ thống. Hướng nhìn này là rất quan trọng, bởi nó ảnh hưởng đến tất cả các hướng nhìn khác của hệ thống. Cả cấu trúc logic lẫn cấu trúc vật lý đều chịu ảnh hưởng từ các ca sử dụng, bởi chức năng được đặc tả trong mô hình này chính là những chức năng được thực thi trong các cấu trúc kia.

### Biểu đồ ca sử dụng

1. Khái niệm
   * Biểu đồ ca sử dụng (Use Case) là kỹ thuật được dùng trong kỹ thuật phần mềm và hệ thống để nắm bắt yêu cầu chức năng của hệ thống. Use case mô tả sự tương tác đặc trưng giữa người dùng (Actor) và hệ thống.
2. Các ký pháp

* Tác nhân (Actor): có thể là một người, hệ thống, các kích hoạt thời gian…Tác nhân tương tác với hệ thống bằng cách gửi, nhận hoặc thay đổi các thông tin cùng với hệ thống. Đó là các ca sử dụng mà tác nhân thực hiện. Ký hiệu tác nhân là biểu tượng hình người. Tác nhân được đặt tên bằng danh từ và phản ánh vai trò của tác nhân trong hệ thống. Tác nhân gồm 2 loại: primary actor và secondary actor.
  + Ca sử dụng (Use Case): Ca sử dụng mô tả các chức năng sử dụng cụ thể của hệ thống bởi 1 tác nhân. Mỗi ca sử dụng mô tả các tương tác của tác nhân với hệ thống để hoàn thành nhiệm vụ cụ thể. Một tác nhân có thể liên kết tới một hoặc nhiều ca sử dụng.
  + Biên hệ thống (System): biên hệ thống có thể là toàn bộ hệ thống như được định nghĩa trong các tài liệu yêu cầu. Với các hệ thống lớn và phức tạp, mỗi phân hệ có thể là biên hệ thống. Toàn bộ hệ thống có thể bao toàn bộ các phân hệ này mô tả biên hệ thống tổng thể.
  + Mối quan hệ (Relationship): là các mỗi quan hệ giữa các tác nhân với tác nhân, tác nhân với ca sử dụng, ca sử dụng với ca sử dụng. Các quan hệ này gồm:
    - Liên kết (Association): là mối quan hệ giữa tác nhân và ca sử dụng, kí hiệu đường nét liền, có thể có mũi tên nếu là liên kết có hướng.
    - Kế thừa (Generalization): là mối quan hệ kế thừa của tác nhân với tác nhân, ca sử dụng với ca sử dụng, kí hiệu đường nét liền có mũi tên hình tam giác
    - Bao gồm (Include): là mối quan hệ giữa hai ca sử dụng, kí hiệu mũi tên nét đứt và có trường <<Include>> trên đường nét đứt.
    - Mở rộng (Extend): là mối quan hệ có hướng giữa hai ca sử dụng với nhau, kí hiệu mũi tên có hướng, đầu mũi là hình tam giác đậm và có trường <<Extend>> trên đường nét đứt.

1. Các bước xây dựng mô hình ca sử dụng

* Để xây dựng mô hình ca sử dụng hoàn chỉnh cần phải trải qua 10 bước được phân thành 3 giai đoạn chính: Giai đoạn mô hình hóa, giai đoạn cấu trúc và giai đoạn review.
* Giai đoạn mô hình hóa:
* Bước 1: Thực hiện thiết lập ngữ cảnh của hệ thống.
* Bước 2: Xác định các Actor.
* Bước 3: Xác định các Use Case.
* Bước 4: Định nghĩa các mối quan hệ giữa Actor và Use Case.
* Bước 5: Đánh giá các mối quan hệ đó để tìm cách chi tiết hóa.
* Giai đoạn cấu trúc:
* Bước 6: Đánh giá các Use Case cho quan hệ Include.
* Bước 7: Đánh giá các Use Case cho quan hệ Extend.
* Bước 8: Đánh giá các Use Case cho quan hệ Generalization .
* Giai đoạn review:
* Bước 9: Kiểm tra (verification): Đảm bảo hệ thống đúng với tài liệu đặc tả.
* Bước 10: Thẩm định (validation): Đảm bảo hệ thống sẽ được phát triển là thứ mà khách hàng cuối thực sự cần thiết.

# PHÂN TÍCH THIẾT KẾ PHẦN MỀM

## Thông tin khảo sát

### Thông tin chung

#### Thời gian

* + Kéo dài 1 tháng từ tháng 2 năm 2023 đến tháng 3 năm 2023

#### Địa điểm

* + Nghiên cứu sẽ được thực hiện tại các văn phòng Bất động sản trong tình Hưng yên. Để tập trung cho phạm vi nghiên cứu, tôi đã lựa chọn những văn phòng có quy mô vừa và nhỏ

#### Thành phần tham gia

* Đại diện từ khách hàng
* Nhân viên quản lý
* Nhân viên kế toán
* Cơ quan chức năng và các bên liên quan khác tham gia vào quá trình khảo sát thông tin

#### Thành phần cung cấp thông tin

* Tài liệu
* Hồ sơ
* Quy trình làm việc hiện tại
* Dữ liệu từ các bộ phận hoặc đơn vị liên quan
* Thông tin từ các bên liên quan khác

### Nội dung khảo sát

#### Yêu cầu của khách hàng

Bảng 3.1 Yêu cầu của khách hàng

|  |  |
| --- | --- |
| STT | Tên yêu cầu |
| **1** | Chức năng phân quyền cho các tài khoản sử dụng trong ứng dụng, quản lý thông tin của tài khoản của ứng dụng |
| **2** | Chức năng quản lý hợp đồng theo từng loại : hợp đồng tại văn phòng, hợp đồng mang về và hợp đồng ký kết giữa 2 bên |
| **3** | Chức năng quản lý thông tin khách hàng |
| **4** | Chức năng quản lý thông tin nhân viên |
| **5** | Chức năng quản lý thông tin mảnh đất |
| **6** | Chức năng thống kê báo cáo doanh thu của văn phòng theo ngày |

#### Quy trình nghiệp vụ

1. Quy trình nghiệp vụ quản lý hợp đồng
   * Khi khách hàng yêu cầu thông tin về 1 mảnh đất tại văn phòng. Nhân viên tiếp nhận yêu cầu và cung cấp thông tin và danh sách thông tin mảnh đất: vị trí mảnh đất, diện tích mảnh đất, giá tiền và ghi chú của mảnh đất.
   * Với khách hàng đặt hợp đồng thì nhân viên cần ghi thêm địa chỉ, số điện thoại và tên khách hàng
2. Quy trình nghiệp vụ quản lý thông tin khách hàng
   * Nhân viên ghi lại các thông tin về họ tên, số điện thoại, địa chỉ email… vào giấy và viết lại vào file excel.
3. Quy trình nghiệp vụ quản lý thông tin nhân viên
   * Quản lý ghi lại các thông tin nhân viên : họ tên, số điện thoại, địa chỉ email … vào file excel.
4. Quy trình nghiệp vụ quản lý thông tin mảnh đất
   * Các mảnh đất của văn phòng được lưu trữ trên cơ sở dữ liệu của chương trình
5. Quy trình nghiệp vụ quản lý thông tin mảnh đất của văn phòng
   * Các mảnh đất của văn phòng được phân loại theo từng danh mục mảnh đất và được lưu lại thông tin về đơn giá trong file excel và được in ra từ văn phòng.
6. Quy trình nghiệp vụ thống kê doanh thu
   * Quản lý tổng hợp doanh thu theo từng ngày và viết lại trong file excel.

## Mô tả yêu cầu hệ thống

### Yêu cầu chức năng

Bảng 3.2 Bảng yêu cầu chức năng của hệ thống

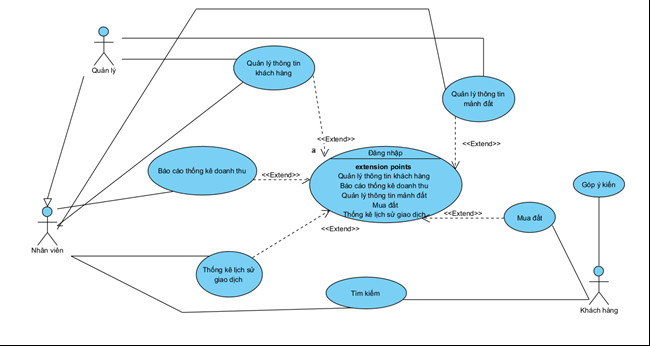
|  |  |  |
| --- | --- | --- |
| Mục | Tên yêu cầu | Mô tả yêu cầu |
| **I** | **Các yêu cầu chức năng nghiệp vụ** | Là các chức năng của phần mềm tương ứng với các công việc trong thế giới thực. |
| 1 | Quản lý khách hàng | Thêm sửa xoá, tìm kiếm thông tin khách hàng |
| 2 | Quản lý nhân viên | Thêm sửa xoá, tìm kiếm thông tin nhân viên |
| 3 | Quản lý mảnh đất | Thêm sửa xoá, tìm kiếm thông tin mảnh đất |
| 4 | Quản lý hợp đồng | Thêm sửa xoá thông tin hợp đồng và chi tiết hợp đồng |
| 5 | Thống kê báo cáo | Thống kê doanh thu văn phòng theo ngày, vẽ biểu đồ doanh thu |
| **II** | **Các yêu cầu chức năng hệ thống** | Là các chức năng của phần mềm được phát sinh thêm khi thực hiện một công việc trên máy tính thay vì trong thế giới thực hoặc không tương ứng với bất kỳ công việc nào trong thế giới thực. |
| 1 | Quản lý tài khoản | Thêm sửa xoá thông tin tài khoản |
| 2 | Đăng nhập | Người dùng nhập thông tin tên đăng nhập và mật khẩu, nếu chính xác thì mở trang chủ của ứng dụng và hiển thị các mục tùy theo quyền của tài khoản được đăng nhập |

### Yêu cầu phi chức năng

Bảng 3.3 Bảng yêu cầu phi chức năng của hệ thống

|  |  |  |
| --- | --- | --- |
| Mục | Tên yêu cầu | Mô tả yêu cầu |
| 1 | Hiệu xuất | Đảm bảo ứng dụng hoạt động một cách nhanh chóng và đáp ứng đúng thời gian thực, đồng thời đảm bảo tối ưu hóa tài nguyên hệ thống như bộ nhớ, CPU, và băng thông mạng |
| 2 | Bảo mật | Đảm bảo dữ liệu của văn phòng cà phê, như thông tin khách hàng, thông tin mảnh đất, và thông tin giao dịch, được bảo vệ chống lại việc truy cập trái phép, mất mát hoặc lộ ra ngoài |
| 3 | Tương thích | Đảm bảo ứng dụng tương thích với các thiết bị của khách hàng |
| 4 | Tính khả dụng | Yêu cầu về tính khả dụng đảm bảo rằng giao diện người dùng của ứng dụng dễ sử dụng, thân thiện, và đáp ứng được nhu cầu của người dùng, đồng thời đảm bảo tính nhất văn phòng giữa các chức năng khác nhau của ứng dụng |
| 5 | Độ tin cậy | Yêu cầu về độ tin cậy đảm bảo rằng ứng dụng hoạt động ổn định, ít gặp lỗi, và dễ dàng phục hồi sau các sự cố, đồng thời đảm bảo tính khả dụng của ứng dụng |

## Biểu đồ ca sử dụng [1]



Hình 3.1 Sơ đồ Use case tổng quát

Bảng 3.4 Bảng danh sách Actor

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Actor** | **Ý nghĩa/Ghi chú** |
| 1 | Nhân viên | Thao tác tạo hợp đồng, quản lý thông tin khách hàng |
| 2 | Quản lý | Quản lý tất cả thông tin của văn phòng, quản lý thông tin tài khoản trong ứng dụng |

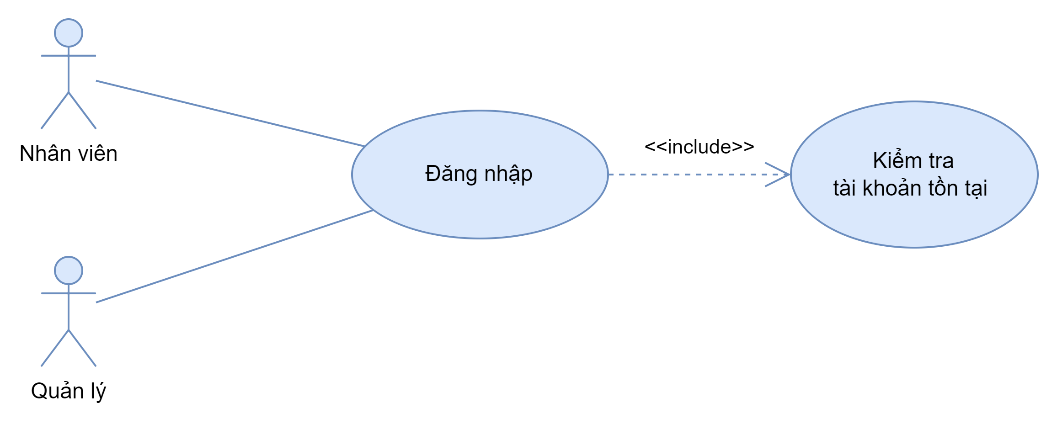
Bảng 3.5 Bảng danh sách Use case

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Use case** | **Ý nghĩa/Ghi chú** |
| 1 | Quản lý thông tin mảnh đất | Tạo các loai hợp đồng: hợp đồng tại văn phòng, hợp đồng từ xa. Chọn, chuyển mảnh đất. Xác nhận thanh toán hợp đồng |
| 2 | Quản lý khách hàng | Quản lý thông tin của khách hàng |
| 3 | Quản lý mảnh đất | Quản lý thông tin mảnh đất trong văn phòng |
| 4 | Quản lý nhân viên | Quản lý thông tin nhân viên của văn phòng |
| 5 | Quản lý danh mục mảnh đất | Quản lý các danh mục mảnh đất của văn phòng |
| 6 | Quản lý tài khoản | Quản lý tài khoản sử dụng trong ứng dụng |
| 7 | Quản lý hợp đồng | Quản lý hợp đồng của văn phòng |
| 8 | Thống kê doanh thu | Xem thông kê, báo cáo doanh thu |

### Đặc tả ca sử dụng

#### Đặc tả Use case Đăng nhập

* **Biểu đồ use case phân rã:**



Hình 3.2 Sơ đồ use case Quản lý thông tin mảnh đất

* **Mô tả tóm tắt:**
  + Tác nhân chính: Quản lý, nhân viên
  + Chức năng: Đăng nhập
  + Ý nghĩa: Giúp hệ thống xác định thông tin và quyền sử dụng của người dùng
* **Dòng sự kiện chính:**

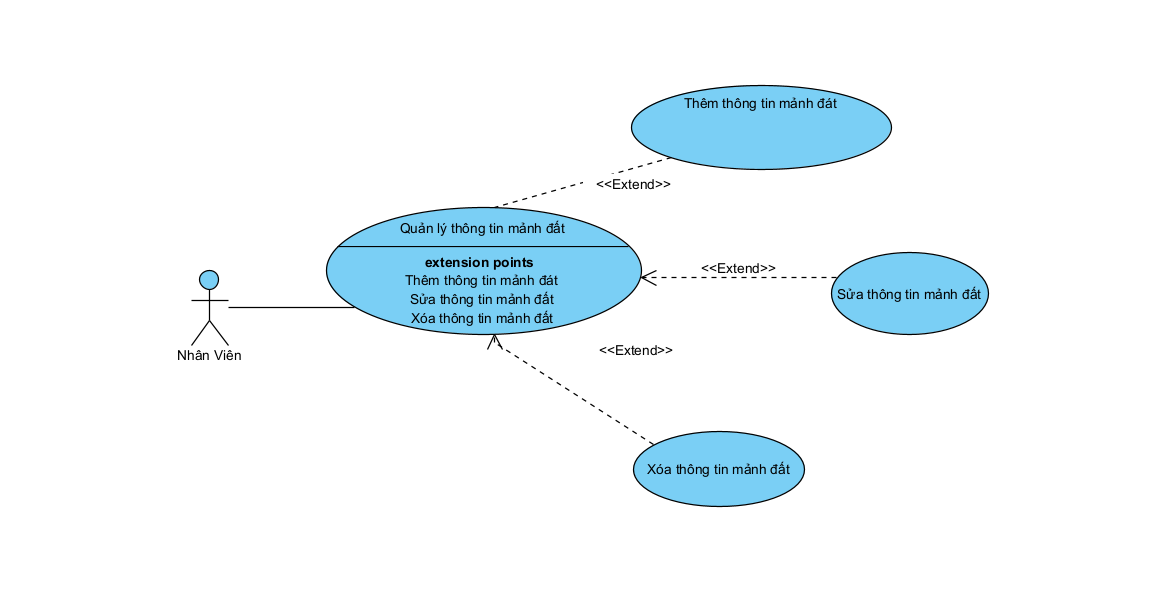
Bảng 3.6 Dòng sự kiện chính Đăng nhập

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Truy cập ứng dụng | 1. Hiển thị form “GUI\_Login” |
| 1. Nhập thông tin đăng nhập |  |
| 1. Nhấn nút đăng nhập | 1. Kiểm tra tính hợp lệ của dữ liệu |
|  | 1. Mở giao diện trang chủ của ứng dụng |

* *Dòng sự kiện phụ:*
* Không có
* *Dòng sự kiện rẽ nhánh:*
  + Thông tin được nhập không hợp lệ : Yêu cầu nhập lại thông tin
* *Dòng sự kiện ngoại lệ:*
  + Ứng dụng bị tắt trước khi mở giao diện: hủy toàn bộ thao tác
* *Trạng thái hệ thống khi bắt đầu thực hiện Use case:*
* Không có
* *Trạng thái hệ thống sau khi thực hiện Use case:*
* Đăng nhập thành công hoặc thất bại

#### Đặc tả use case Quản lý thông tin mảnh đất

* **Biểu đồ use case phân rã:**



Hình 3.3 Sơ đồ use case Quản lý thông tin mảnh đất

* **Mô tả tóm tắt:**
  + Tác nhân chính: Nhân viên
  + Chức năng: Tạo mới mảnh đất, cập nhật/xoá thông tin mảnh đất, tạo mới chi tiết mảnh đất, sửa/xoá chi tiết mảnh đất
  + Ý nghĩa: Thao tác với thông tin mảnh đất cơ bản của chương trình ứng dụng
* **Dòng sự kiện chính:**

Bảng 3.7 Dòng sự kiện chính Use case Quản lý thông tin mảnh đất

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn mục tác vụ tạo hợp đồng (tại văn phòng / giao hợp đồng) | 1. Hiển thị form “GUI\_NhanVien””, hiển thị danh sách thông tin mảnh đất |
| 1. Chọn các chức năng tuỳ chọn: thêm/sửa/xoá/tìm kiếm hợp đồng, thêm/sửa/xoá/tìm kiếm chi tiết mảnh đất | 1. Xác nhận yêu cầu lưu lại thông tin trong cơ sở dữ liệu |

* *Dòng sự kiện phụ:*
* Thêm mảnh đất*:*

Bảng 3.8 Dòng sự kiện thêm hợp đồng Use case Quản lý thông tin mảnh đất

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Nhập thông tin về mảnh đất, mảnh đất, diện tích mảnh đất, ghi chú, vị trí…. và nhấn nút thêm | 1. Kiểm tra tính hợp lệ của thông tin |
|  | 1. Thêm mới mảnh đất vào hệ thống. Hiển thị thông báo thông tin được thêm. |

* Sửa mảnh đất:

Bảng 3.9 Dòng sự kiện sửa hợp đồng Use case Quản lý thông tin mảnh đất

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn mảnh đất muốn sửa | 1. Hiển thị thông tin của mảnh đất lên form |
| 1. Chỉnh sửa thông tin mảnh đất trên form và nhấn nút sửa | 1. Kiểm tra tính hợp lệ của thông tin |
|  | 1. Thông báo sửa thông tin và lưu thông tin vào hệ thống |

* Xoá mảnh đất:

Bảng 3.10 Dòng sự kiện xoá hợp đồng Use case Quản lý thông tin mảnh đất

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn chi tiết mảnh đất muốn xoá và nhấn nút xoá | 1. Xoá chi tiết mảnh đất trong hệ thống |

* Thêm chi tiết mảnh đất*:*

Bảng 3.11 Dòng sự kiện thêm chi tiết hợp đồng Use case Quản lý thông tin mảnh đất

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Nhập các thông tin của hợp đồng lên form và nhấn nút Thêm chi tiết hợp đồng | 1. Kiểm tra tính hợp lệ của thông tin |
|  | 1. Lưu thông tin vào hệ thống |

* Sửa hợp đồng:

Bảng 3.12 Dòng sự kiện sửa chi tiết hợp đồng Use case Quản lý thông tin mảnh đất

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn chi tiết hợp đồng muốn sửa | 1. Hiển thị thông tin của chi tiết hợp đồng lên form |
| 1. Chỉnh sửa thông tin chi tiết hợp đồng trên form và nhấn nút sửa | 1. Kiểm tra tính hợp lệ của thông tin và thông báo xác nhận lưu thay đổi và lưu thông tin vào hệ thống |

* Xoá hợp đồng:

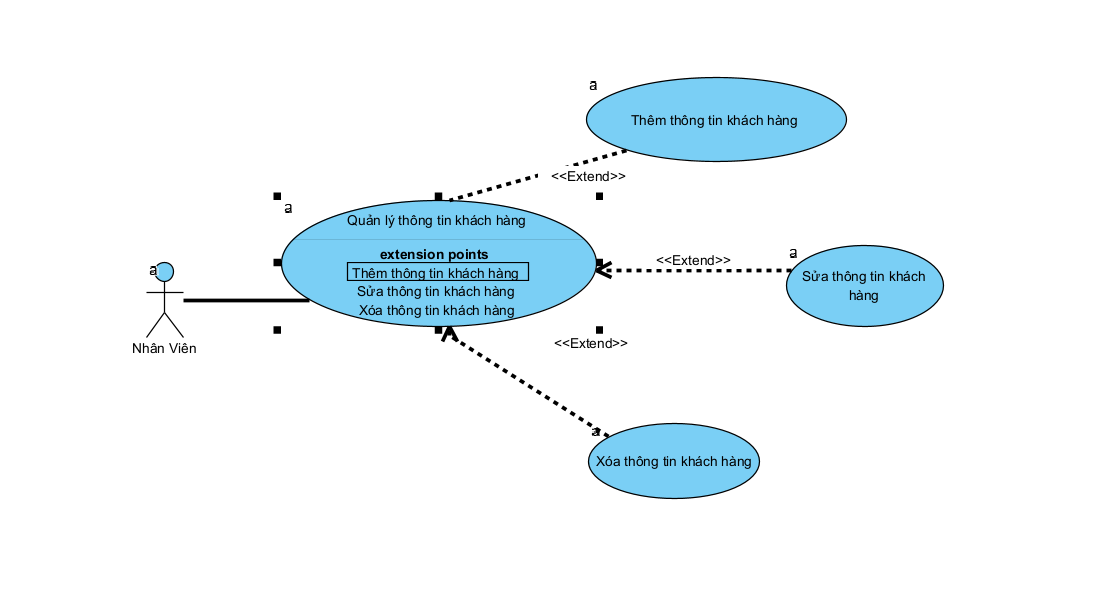
Bảng 3.13 Dòng sự kiện xoá chi tiết hợp đồng Use case Quản lý thông tin mảnh đất

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn chi tiết hợp đồng muốn xoá và nhấn nút xoá | 1. Xoá chi tiết hợp đồng trong hệ thống và thông báo xoá |

* *Dòng sự kiện rẽ nhánh:*
  + Thông tin được nhập không hợp lệ : Yêu cầu nhập lại thông tin
  + Khi thêm chi tiết hợp đồng: nếu đã tồn tại chi tiết hợp đồng có cùng thông tin về mã hợp đồng và mã mảnh đất thì cộng thêm diện tích mảnh đất và thông tin ghi chú vào chi tiết hợp đồng trước đó
* *Dòng sự kiện ngoại lệ:*
  + Ứng dụng bị tắt trước khi lưu thông tin vào hệ thống: huỷ toàn bộ thao tác
* *Trạng thái hệ thống khi bắt đầu thực hiện Use case:*
* Đăng nhập thành công
* *Trạng thái hệ thống sau khi thực hiện Use case:*
* Hiển thị danh sách chi tiết hợp đồng theo mảnh đất
* Thêm, sửa, xoá, chuyển mảnh đất thành công hoặc thất bại

#### Đặc tả use case Quản lý khách hàng

* **Biểu đồ use case phân rã:**



Hình 3.4 Sơ đồ use case Quản lý khách hàng

* **Mô tả tóm tắt:**
  + Tác nhân chính: Quản lý, nhân viên
  + Chức năng: Thêm, sửa, xoá, tìm kiếm thông tin khách hàng
  + Ý nghĩa: Quản lý thông tin khách hàng của văn phòng
* **Dòng sự kiện chính:**

Bảng 3.14 Dòng sự kiện chính khách hàng Use case quản lý khách hàng

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn mục quản lý khách hàng | 1. Hiển thị form “GUI\_KhachHang” hiển thị danh sách thông tin khách hàng |
| 1. Lựa chọn các chức năng thêm/sửa/xoá/tìm kiếm thông tin khách hàng | 1. Lưu thông tin vào hệ thống |

* *Dòng sự kiện phụ:*
* Thêm khách hàng

Bảng 3.15 Dòng sự kiện thêm khách hàng Use case quản lý khách hàng

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Nhập thông tin khách hàng và nhấn nút thêm | 1. Kiểm tra tính hợp lệ của thông tin |
|  | 1. Thêm mới thông tin khách hàng vào hệ thống. Hiển thị thông báo thông tin được thêm. |

* Sửa khách hàng:

Bảng 3.16 Dòng sự kiện sửa khách hàng Use case quản lý khách hàng

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn thông tin khách hàng muốn sửa | 1. Hiển thị thông tin của khách hàng lên form |
| 1. Chỉnh sửa thông tin khách hàng trên form và nhấn nút sửa | 1. Kiểm tra tính hợp lệ của thông tin và |
|  | 1. Thông báo sửa thông tin và lưu thông tin vào hệ thống |

* Xoá khách hàng:

Bảng 3.17 Dòng sự kiện xoá khách hàng Use case quản lý khách hàng

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn chi tiết khách hàng muốn xoá và nhấn nút xoá | 1. Xoá thông tin khách hàng trong hệ thống |

* + Tìm khách hàng:

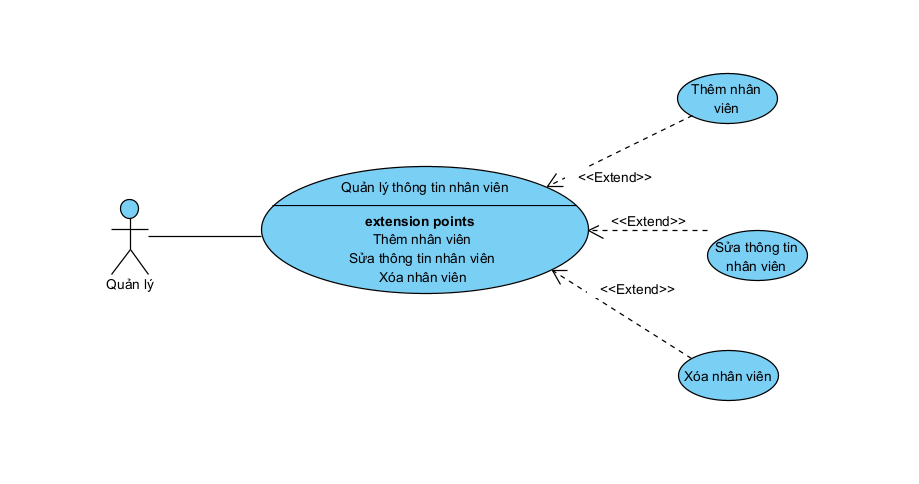
Bảng 3.18 Dòng sự kiện tìm kiếm khách hàng Use case quản lý khách hàng

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Nhập từ khoá vào ô tìm kiếm và nhấn nút tìm kiếm | 1. Hiển thị danh sách những khách hàng có tên chứa từ khoá được nhập |

* *Dòng sự kiện rẽ nhánh:*
  + Thông tin được nhập không hợp lệ : Yêu cầu nhập lại thông tin
* *Dòng sự kiện ngoại lệ:*
  + Ứng dụng bị tắt trước khi lưu thông tin vào hệ thống: huỷ toàn bộ thao tác
* *Trạng thái hệ thống khi bắt đầu thực hiện Use case:*
* Đăng nhập thành công
* *Trạng thái hệ thống sau khi thực hiện Use case:*
* Hiển thị danh sách thông tin khách hàng.
* Thêm, sửa, xoá thành công hoặc thất bại

#### Đặc tả use case quản lý nhân viên

* **Biểu đồ use case phân rã:**



Hình 3.6 Sơ đồ use case Quản lý nhân viên

* **Mô tả tóm tắt:**
  + Tác nhân chính: Quản lý
  + Chức năng: Thêm, sửa, xoá, tìm kiếm thông tin nhân viên
  + Ý nghĩa: Quản lý thông tin nhân viên của văn phòng
* **Dòng sự kiện chính:**

Bảng 3.24 Dòng sự kiện chính Use case quản lý nhân viên

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn mục quản lý nhân viên | 1. Hiển thị form “GUI\_QuanLy” hiển thị danh sách thông tin nhân viên |
| 1. Lựa chọn các chức năng thêm/sửa/xoá/tìm kiếm thông tin nhân viên | 1. Lưu thông tin vào hệ thống |

* *Dòng sự kiện phụ:*
* Thêm nhân viên

Bảng 3.25 Dòng sự kiện thêm nhân viên

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Nhập thông tin nhân viên và nhấn nút xoá | 1. Kiểm tra tính hợp lệ của thông tin |
|  | 1. Thêm mới thông tin nhân viên vào hệ thống. Hiển thị thông báo thông tin được thêm. |

* Sửa nhân viên:

Bảng 3.26 Dòng sự kiện sửa nhân viên

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn thông tin nhân viên muốn sửa | 1. Hiển thị thông tin của nhân viên lên form |
| 1. Chỉnh sửa thông tin nhân viên trên form và nhấn nút sửa | 1. Kiểm tra tính hợp lệ của thông tin và |
|  | 1. Thông báo sửa thông tin và lưu thông tin vào hệ thống |

* Xoá nhân viên:

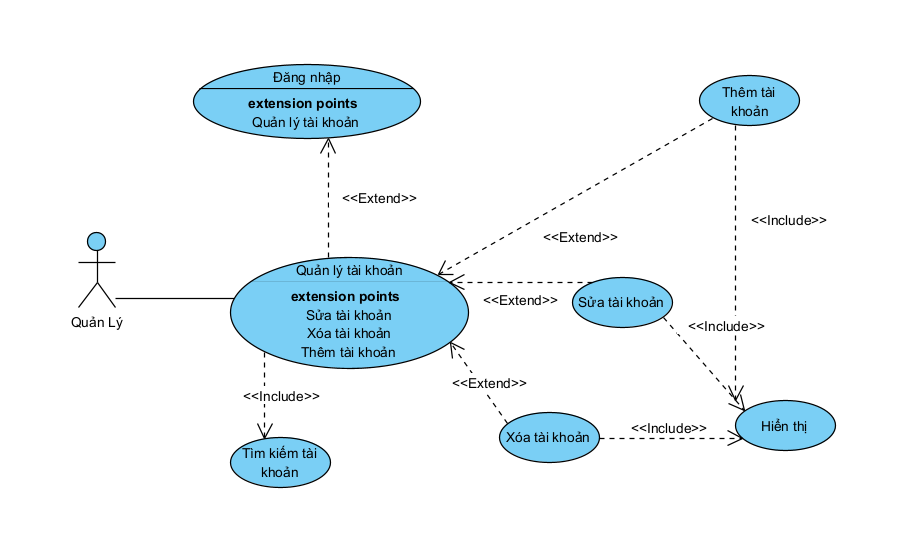
Bảng 3.27 Dòng sự kiện xoá nhân viên

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn chi tiết nhân viên muốn xoá và nhấn nút xoá | 1. Xoá thông tin nhân viên trong hệ thống |

* + Tìm nhân viên:
* *Dòng sự kiện rẽ nhánh:*
  + Thông tin được nhập không hợp lệ : Yêu cầu nhập lại thông tin
* *Dòng sự kiện ngoại lệ:*
  + Ứng dụng bị tắt trước khi lưu thông tin vào hệ thống: huỷ toàn bộ thao tác
* *Trạng thái hệ thống khi bắt đầu thực hiện Use case:*
* Đăng nhập thành công với quyền Quản lý
* *Trạng thái hệ thống sau khi thực hiện Use case:*
* Hiển thị danh sách thông tin nhân viên.
* Thêm, sửa, xoá thành công hoặc thất bại

#### Đặc tả use case quản lý tài khoản

* **Biểu đồ use case phân rã:**



Hình 3.9 Sơ đồ use case Quản lý tài khoản

* **Mô tả tóm tắt:**
  + Tác nhân chính: Quản lý
  + Chức năng: Thêm, sửa, xoá, tìm kiếm thông tin tài khoản
  + Ý nghĩa: Quản lý thông tin tài khoản
* **Dòng sự kiện chính:**

Bảng 3.40 Dòng sự kiện chính Use case Quản lý tài khoản

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn mục quản lý tài khoản | 1. Hiển thị form “GUI\_Account” hiển thị danh sách thông tin tài khoản |
| 1. Lựa chọn các chức năng thêm/sửa/xoá/tìm kiếm thông tin tài khoản | 1. Lưu thông tin vào hệ thống |

* *Dòng sự kiện phụ:*
* Thêm tài khoản

Bảng 3.41 Dòng sự kiện thêm tài khoản

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Nhập thông tin tài khoản và nhấn nút thêm | 1. Kiểm tra tính hợp lệ của thông tin |
|  | 1. Thêm mới thông tin tài khoản vào hệ thống. Hiển thị thông báo thông tin được thêm. |

* Sửa tài khoản:

Bảng 3.42 Dòng sự kiện sửa tài khoản

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn thông tin tài khoản muốn sửa | 1. Hiển thị thông tin của tài khoản lên form |
| 1. Chỉnh sửa thông tin tài khoản trên form và nhấn nút sửa | 1. Kiểm tra tính hợp lệ của thông tin và |
|  | 1. Thông báo sửa thông tin và lưu thông tin vào hệ thống |

* Xoá tài khoản:

Bảng 3.43 Dòng sự kiện xoá tài khoản

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Chọn chi tiết tài khoản muốn xoá và nhấn nút xoá | 1. Xoá thông tin tài khoản trong hệ thống |

* + Tìm tài khoản theo tên tài khoản:

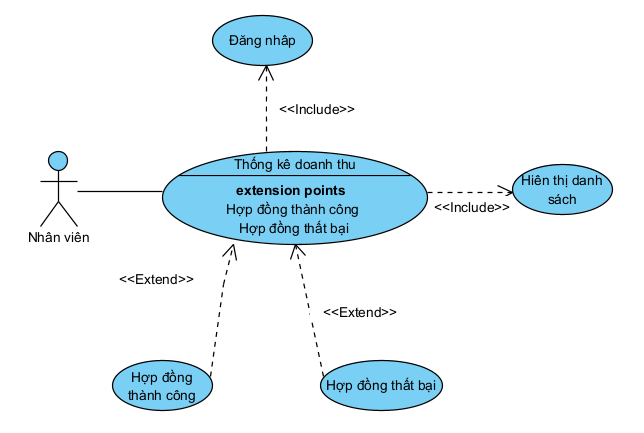
Bảng 3.44 Dòng sự kiện tìm tài khoản theo tên tài khoản

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Nhập từ khoá vào ô tìm kiếm và nhấn nút tìm kiếm | 1. Hiển thị danh sách những tài khoản có tên đăng nhập chứa từ khoá được nhập |

* *Dòng sự kiện rẽ nhánh:*
  + Thông tin được nhập không hợp lệ : Yêu cầu nhập lại thông tin
* *Dòng sự kiện ngoại lệ:*
  + Ứng dụng bị tắt trước khi lưu thông tin vào hệ thống: huỷ toàn bộ thao tác
* *Trạng thái hệ thống khi bắt đầu thực hiện Use case:*
* Đăng nhập thành công với quyền Quản lý
* *Trạng thái hệ thống sau khi thực hiện Use case:*
* Hiển thị danh sách thông tin tài khoản.
* Thêm, sửa, xoá thành công hoặc thất bại

#### Đặc tả use case thống kê doanh thu

* **Biểu đồ use case phân rã:**



Hình 3.10 Sơ đồ use case Thống kê doanh thu

* **Mô tả tóm tắt:**
  + Tác nhân chính: Nhân viên
  + Chức năng: Vẽ biểu đồ doanh thu của văn phòng trong khoảng thời gian
  + Ý nghĩa: Xem thống kê doanh thu của văn phòng
* **Dòng sự kiện chính:**

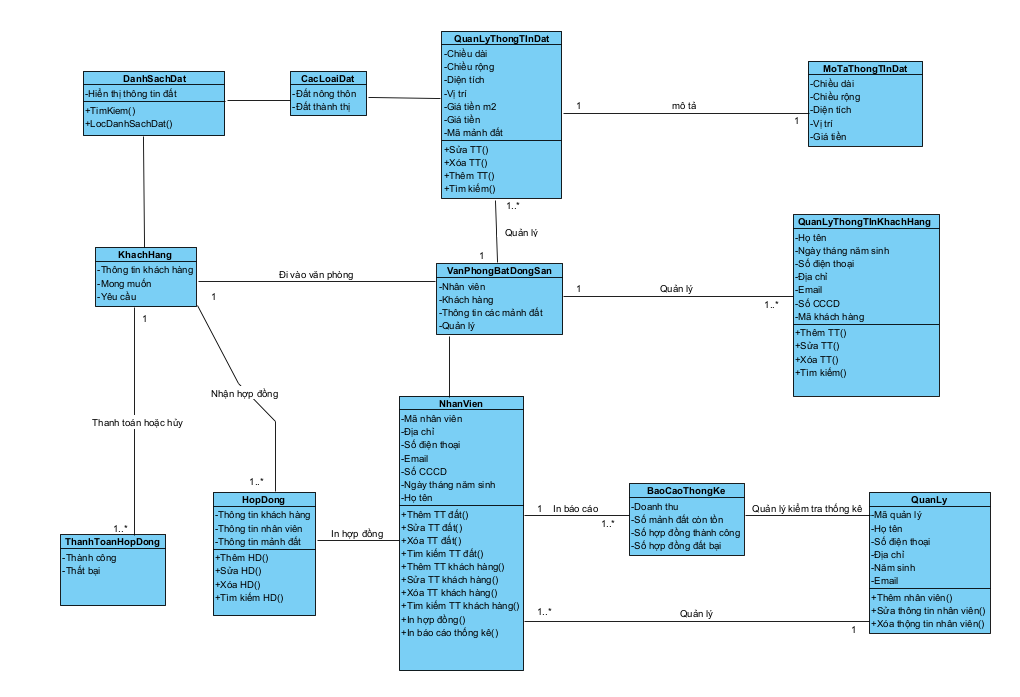
Bảng 3.45 Dòng sự kiện thống kê doanh thu

|  |  |
| --- | --- |
| **Hành động của tác nhân** | **Phản ứng của hệ thống** |
| 1. Mở chức năng thống kê doanh thu | 1. Hiển thị form “GUI\_ThongKe” hiển thị doanh thu của văn phòng. |
| 1. Lựa chọn khoảng thời gian và nhấn nút xác nhận | 1. Kiểm tra tính hợp lệ của khoảng thời gian được chọn |
|  | 1. Hiện thị các thông số và biểu đồ thống kê của văn phòng theo khoảng thời gian được chọn |

* *Dòng sự kiện phụ:*
  + Không có
* *Dòng sự kiện rẽ nhánh:*
  + Khoảng thời gian được chọn không hợp lệ : thông báo “khoảng thời gian được chọn không hợp lệ”
* *Dòng sự kiện ngoại lệ:*
  + Khi ứng dụng bị tắt trước khi kiểm tra tính hợp lệ của khoảng thời gian hoàn thành : huỷ tất cả thao thác trước đó
* *Trạng thái hệ thống khi bắt đầu thực hiện Use case:*
* Đăng nhập thành công với quyền Nhân viên
* *Trạng thái hệ thống sau khi thực hiện Use case:*
* Hiển thị doanh thu, hiển thị biểu đồ doanh thu theo khoảng thời gian được chọn.

## Biểu đồ lớp thực thể [1] [2]

### Biểu đồ lớp



Hình 3.11 Biểu đồ lớp thực thể

### Đặc tả thực thể trong biểu đồ lớp thực thể

#### Lớp NhanVien

Bảng 3.46 Danh sách các thuộc tính lớp NhanVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaNhanVien | String | Không rỗng | Mã chức vụ |
| 2 | TenNhanVien | String | Không rỗng | Tên chức vụ |
| 3 | SoDienThoai | String | Không rỗng | Số điện thoại |
| 4 | DiaChi | String | Không rỗng | Địa chỉ |
| 5 | SoCCCD | String | Không rỗng | Số căn cước công dân |
| 6 | NamSinh | Date | Không rỗng | Năm sinh |

Bảng 3.47 Danh sách các phương thức lớp NhanVien

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên phương thức** | **Ghi chú** |
| 1 | ThemTTKhachHang() | Thêm thông tin khách hàng |
| 2 | SuaTTKhachHang() | Sửa thông tin khách hàng |
| 3 | XoaTTKhachHang | Xóa thông tin khách hàng |
| 4 | TimKiemTTKhachHang() | Tìm kiếm thông tin khách hàng |
| 5 | ThemTTDat() | Thêm thông tin đất |
| 6 | SuaTTDat() | Sửa thông tin đất |
| 7 | XoaTTDat() | Xóa thông tin đất |
| 8 | TimKiemTTDat() | Tìm kiếm thông tin đất |
| 9 | InHopDong() | In hợp đồng |
| 10 | InBaoCao() | In báo cáo thống kê |

#### Lớp TaiKhoan

Bảng 3.48 Danh sách các thuộc tính lớp TaiKhoan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ý nghĩa** |
| 1 | TenDangNhap | String | Không rỗng | Tên đăng nhập |
| 2 | MatKhau | String | Không rỗng | Mật khẩu tài khoản |
| 3 | Email | String | Không rỗng | Địa chỉ email |

Bảng 3.49 Danh sách các phương thức lớp TaiKhoan

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên phương thức** | **Ghi chú** |
| 1 | LayDSTaiKhoan() | Lấy danh sách thông tin tài khoản |
| 2 | Them(tk) | Thêm thông tin tài khoản |
| 3 | Sua(tk) | Sửa thông tin tài khoản |
| 4 | Xoa(tendangnhap) | Xoá tài khoản |
| 5 | LayThongTin(tendangnhap) | Lấy thông tin tài khoản bằng tên đăng nhập |

#### Lớp Dat

Bảng 3.52 Danh sách các thuộc tính lớp Dat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaManhDat | String | Không rỗng | Mã mảnh đất |
| 2 | ChieuDai | Float | Không rỗng | Chiều dài |
| 3 | ChieuRong | Float | Không rỗng | Chiều rộng |
| 4 | DienTich | Float | Không rỗng | Diện tích |
| 5 | ViTri | String | Không rỗng | Vị trí mảnh đất |
| 6 | GiaTien | Money | Không rỗng | Giá tiền mảnh đất |

Bảng 3.53 Danh sách các phương thức lớp Dat

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên phương thức** | **Ghi chú** |
| 1 | LayDSMD() | Lấy danh sách thông tin mảnh đất |
| 2 | Them(MD) | Thêm mảnh đất |
| 3 | Sua(MD) | Sửa thông tin mảnh đất |
| 4 | Xoa(MD) | Xoá mảnh đất |
| 5 | LayThongTin(MD) | Lấy thông tin mảnh đất bằng mã mảnh đất |

#### Lớp KhachHang

Bảng 3.58 Danh sách các thuộc tính lớp KhachHang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ý nghĩa** |
| 1 | Ma | String | Không rỗng | Mã khách hàng |
| 2 | Ten | String | Không rỗng | Tên khách hàng |
| 3 | GioiTinh | String |  | Giới tính khách hàng |
| 4 | SDT | String | Là chuỗi có 10 chữ số | Số điện thoại khách hàng |
| 5 | DiaChi | String | Không rỗng | Địa chỉ khách hàng |
| 6 | Email | String | Không rỗng | Địa chỉ email khách hàng |

#### Lớp HopDong

Bảng 3.60 Danh sách các thuộc tính lớp HopDong

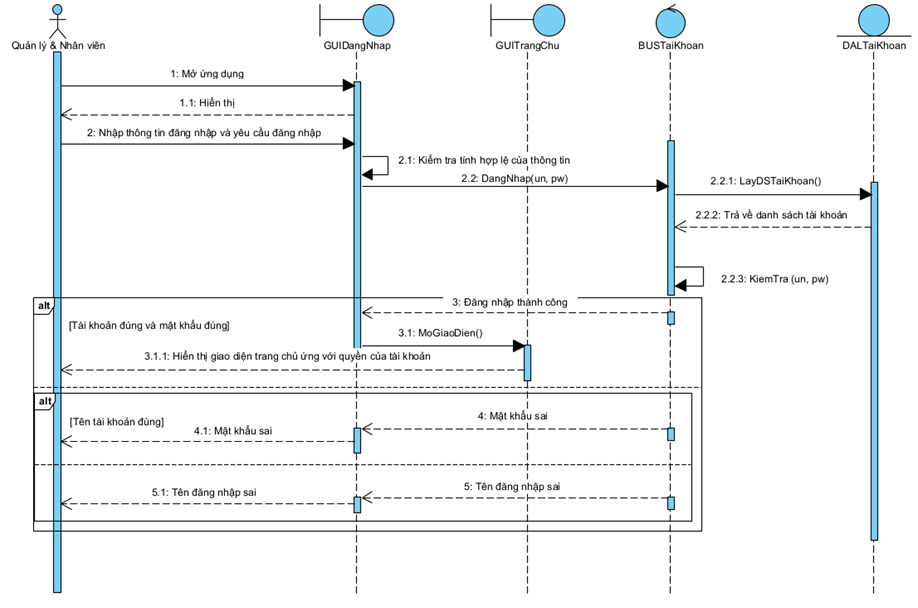
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Ý nghĩa** |
| 1 | MaHopDong | Int | Không rỗng | Mã hợp đồng |
| 2 | MaNhanVien | Int | Không rỗng | Mã nhân viên |
| 3 | MaKhachHang | Int | Không rỗng | Mã khách hàng |
| 4 | MaManhDat | Int | Không rỗng | Mã mảnh đất |

Bảng 3.61 Danh sách các phương thức lớp HopDong

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên phương thức** | **Ghi chú** |
| 1 | LayDSHopDong() | Lấy danh sách thông tin hợp đồng |
| 2 | Them(hd) | Thêm hợp đồng |
| 3 | Sua(hd) | Sửa thông tin hợp đồng |
| 4 | Xoa(ma) | Xoá thông tin hợp đồng |
| 5 | LayThongTin(ma) | Lấy thông tin hợp đồng bằng mã hợp đồng |

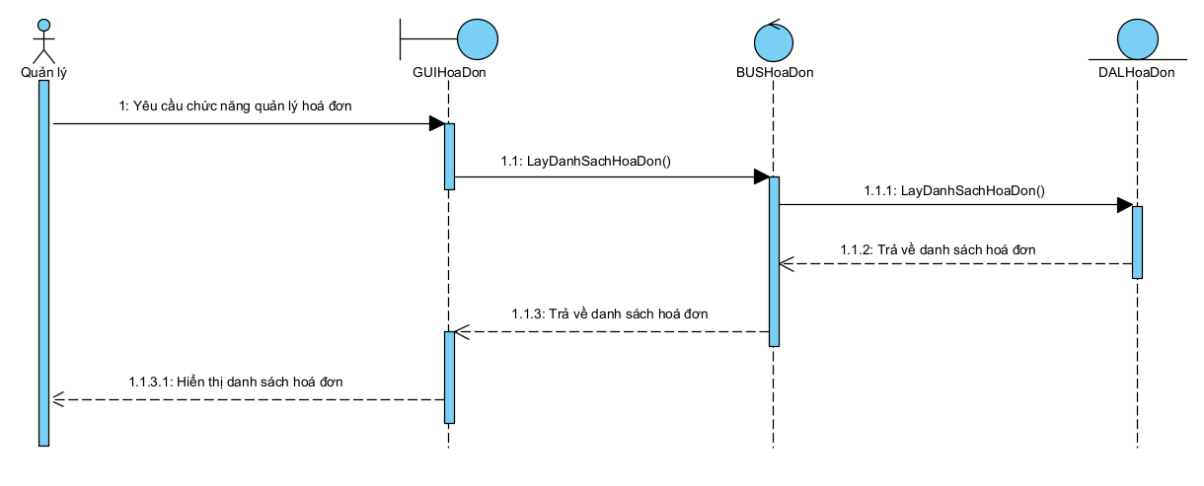
## Biểu đồ tuân tự [1]

### Biểu đồ tuần tự ca sử dụng Đăng nhập

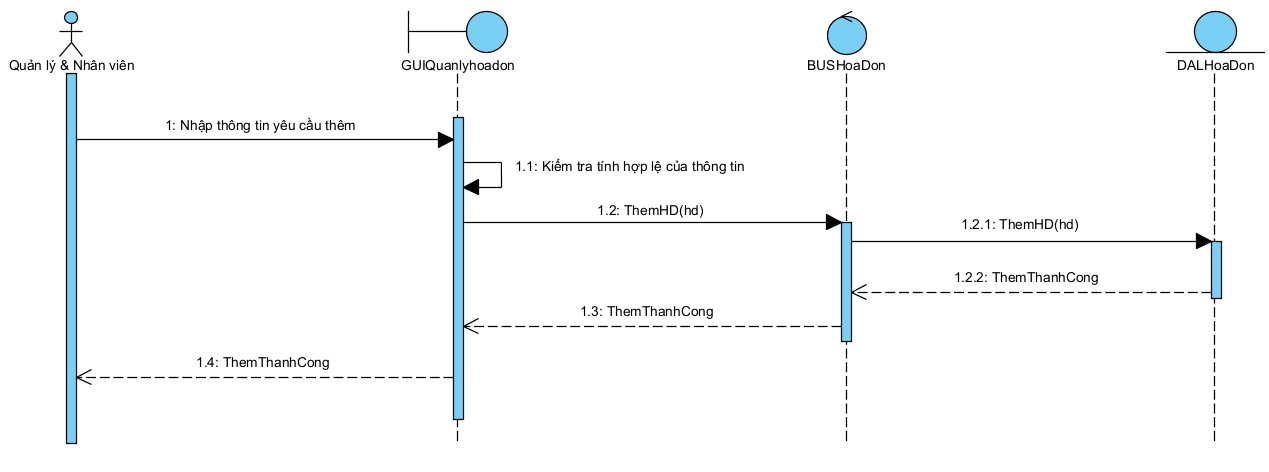


Hình 3.12 Biểu đồ tuần tự tác vụ Đăng nhập

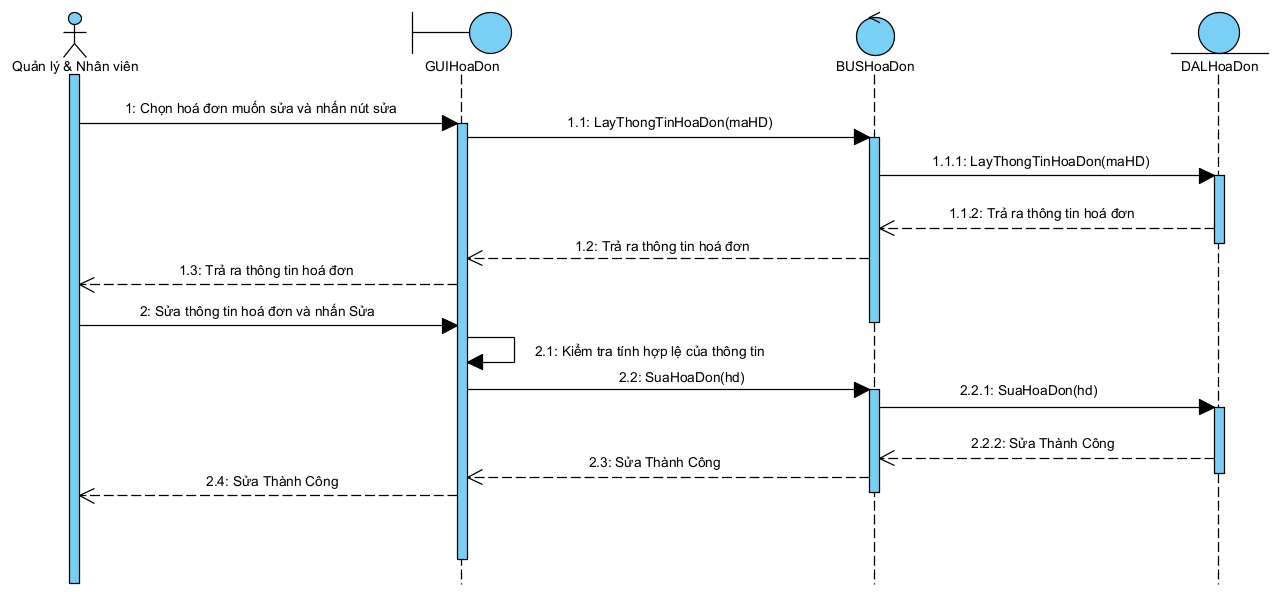
### Biểu đồ tuần tự ca sử dụng Quản lý hợp đồng



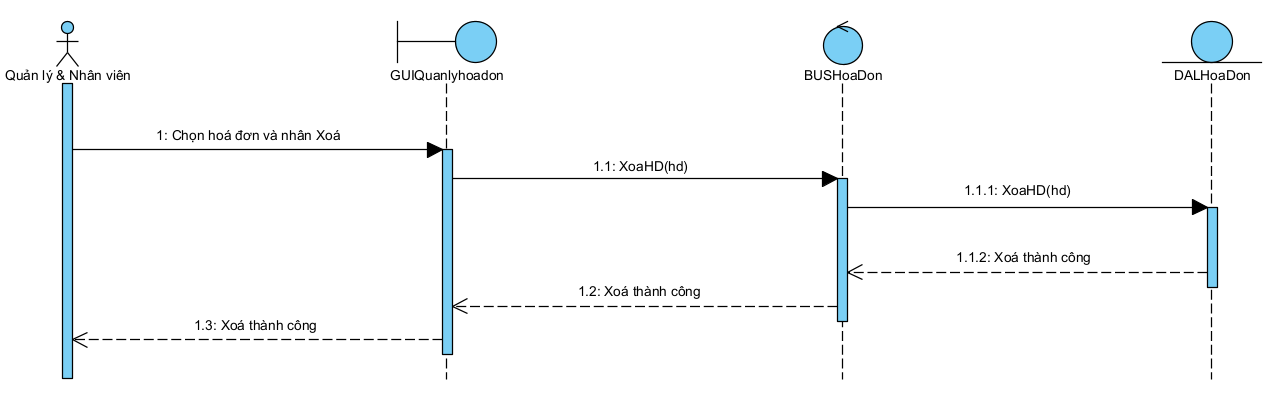
Hình 3.13 Biểu đồ tuần tự tác vụ hiển thị danh sách hợp đồng



Hình 3.14 Biểu đồ tuần tự thêm hợp đồng

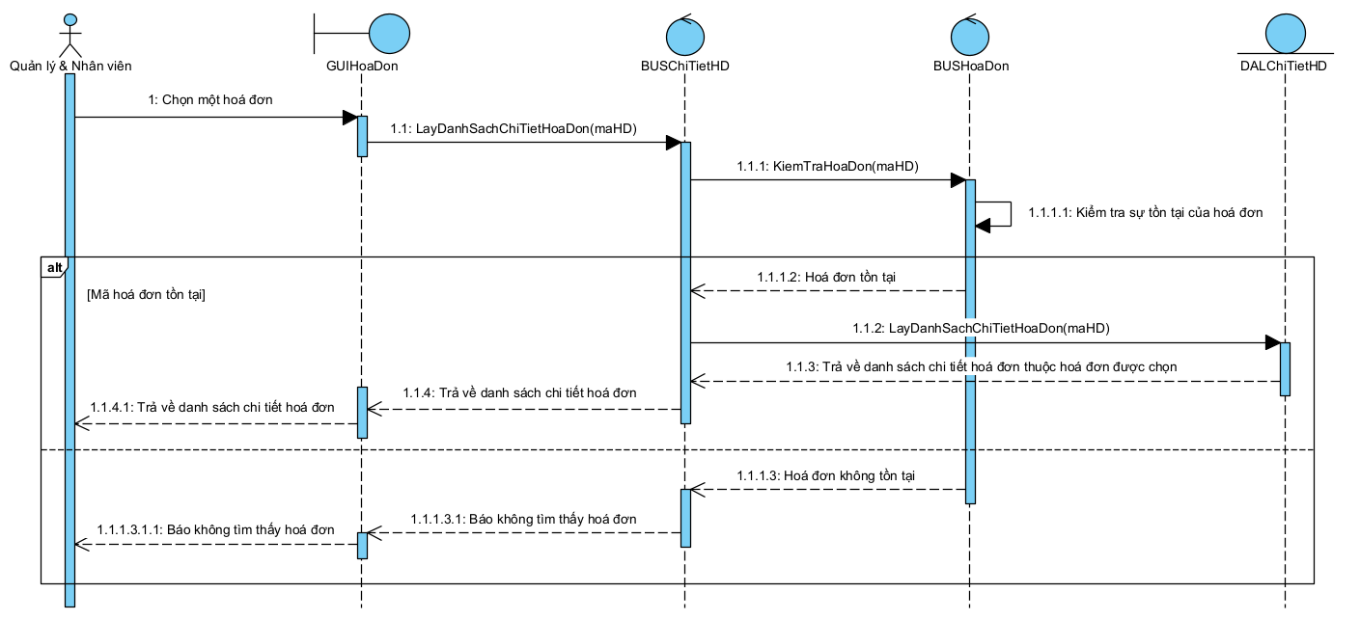


Hình 3.15 Biểu đồ tuần tự sửa hợp đồng

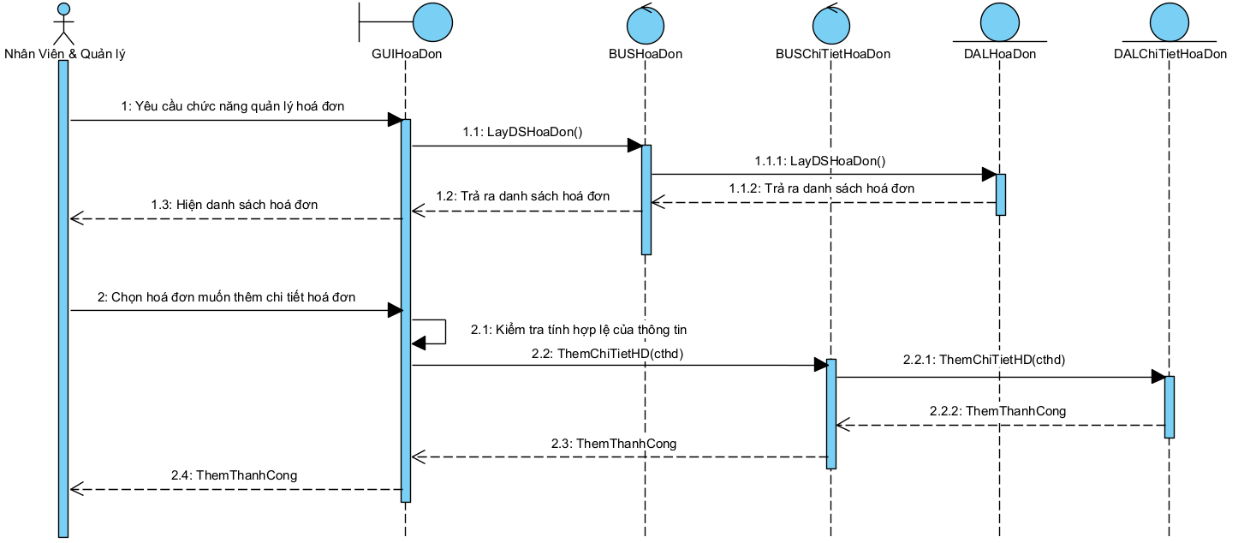


Hình 3.16 Biểu đồ tuần tự xoá hợp đồng

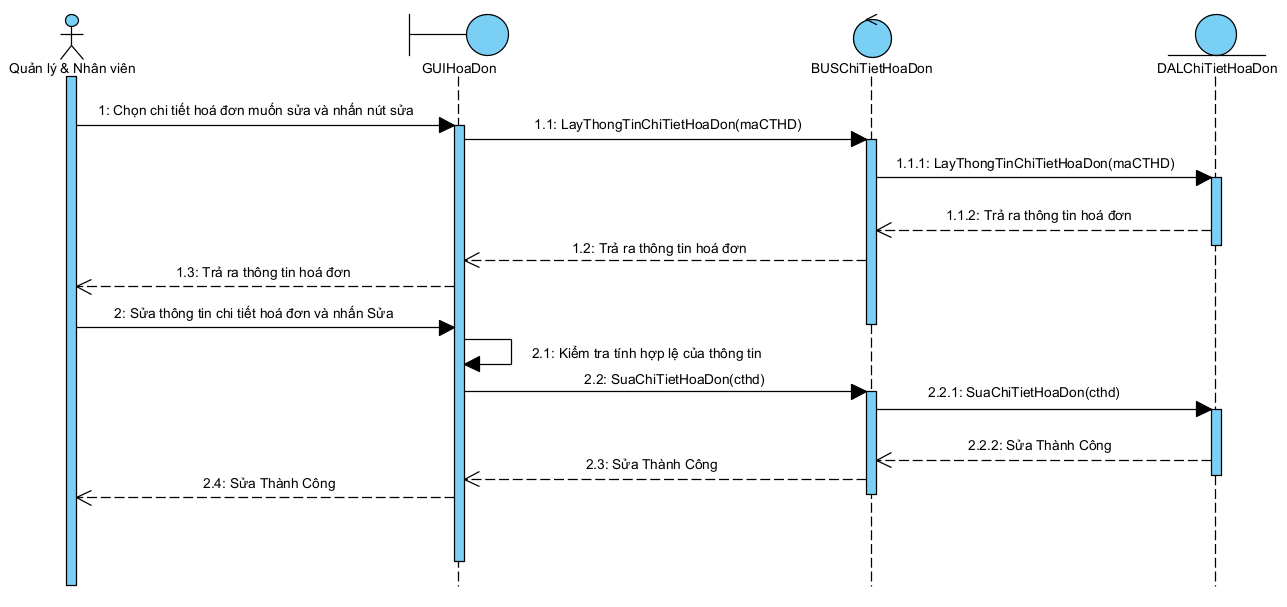
### Biểu đồ tuần tự ca sử dụng Quản lý chi tiết hợp đồng



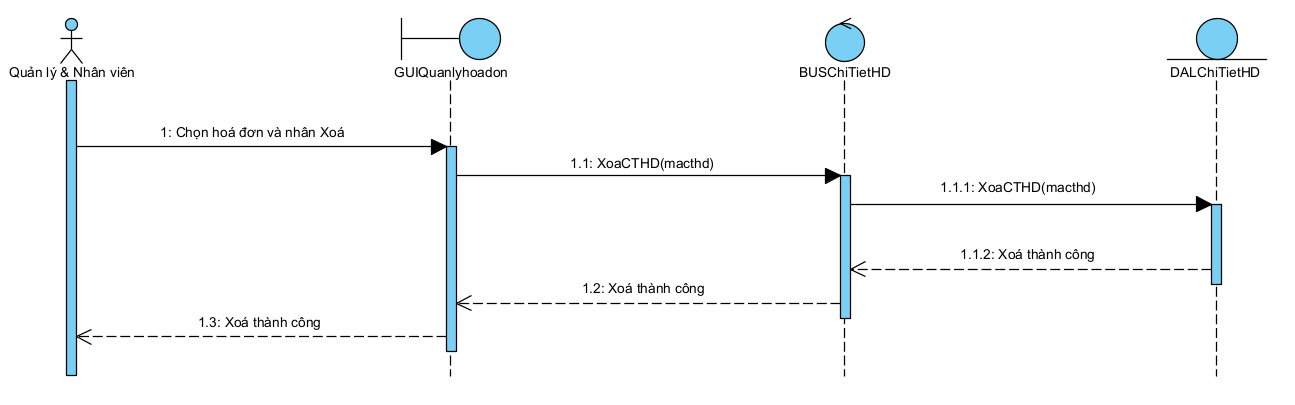
Hình 3.17 Biểu đồ tuần tự tác vụ hiển thị danh sách chi tiết hợp đồng



Hình 3.18 Biểu đồ tuần tự thêm chi tiết hợp đồng

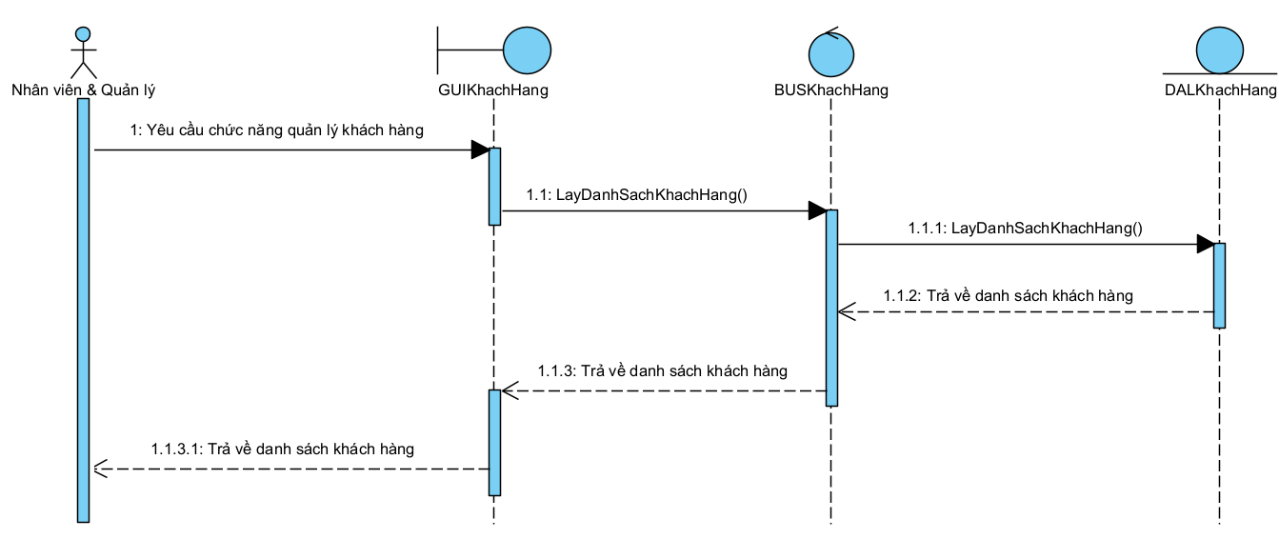


Hình 3.19 Biểu đồ tuần tự sửa chi tiết hợp đồng

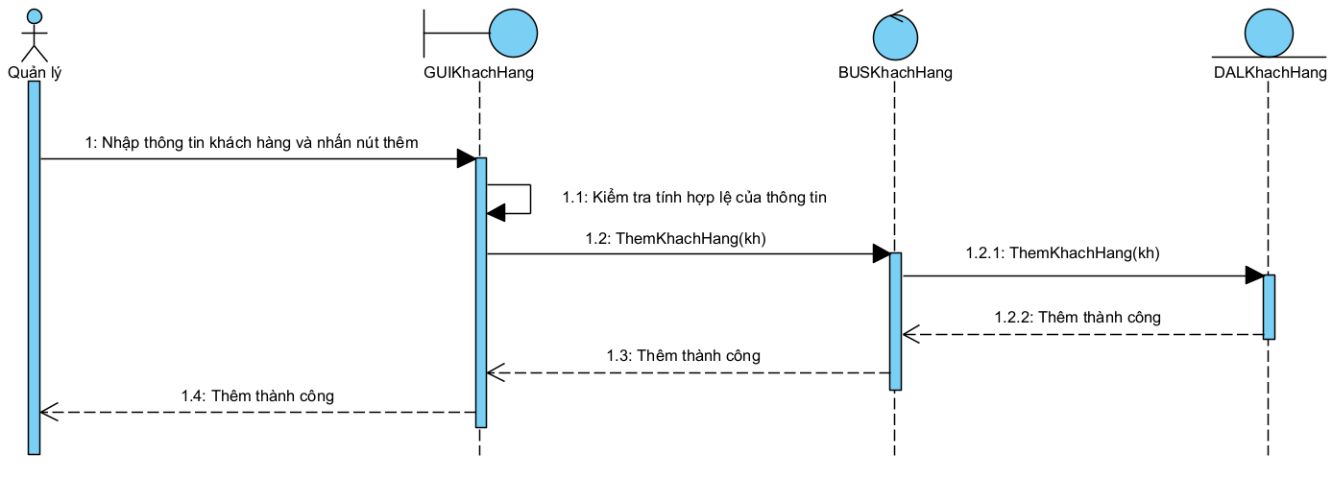


Hình 3.20 Biểu đồ tuần tự xoá chi tiết hợp đồng

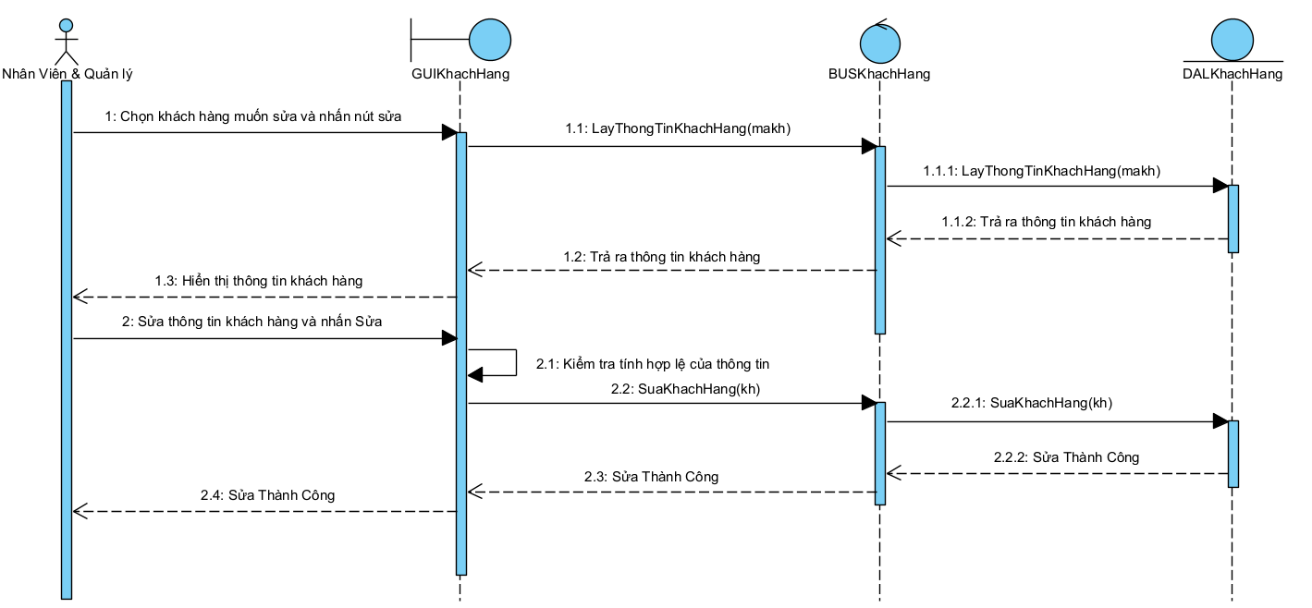
### Biểu đồ tuần tự ca sử dụng Quản lý khách hàng



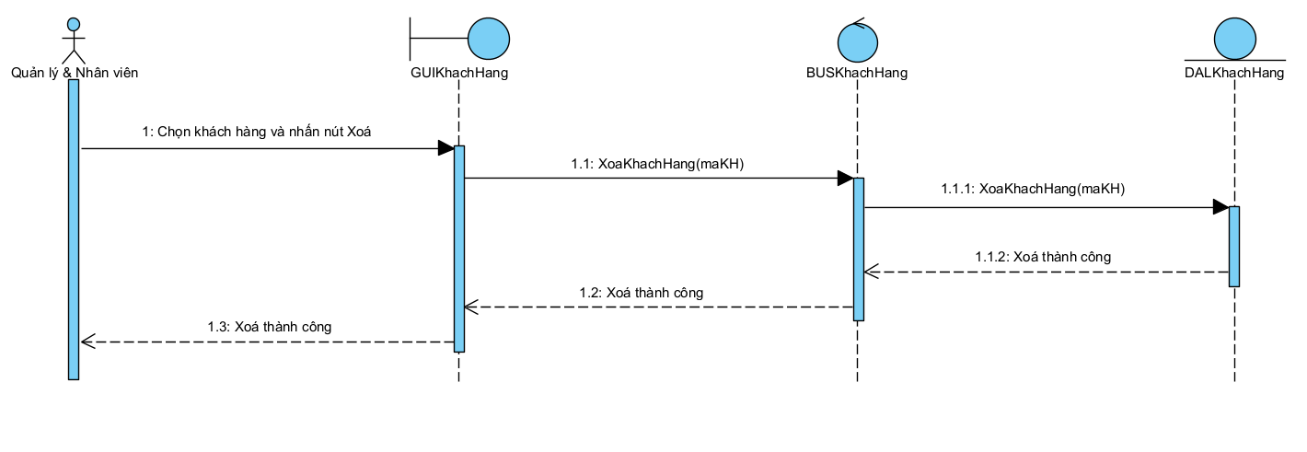
Hình 3.21 Biểu đồ tuần tự tác vụ hiển thị khách hàng



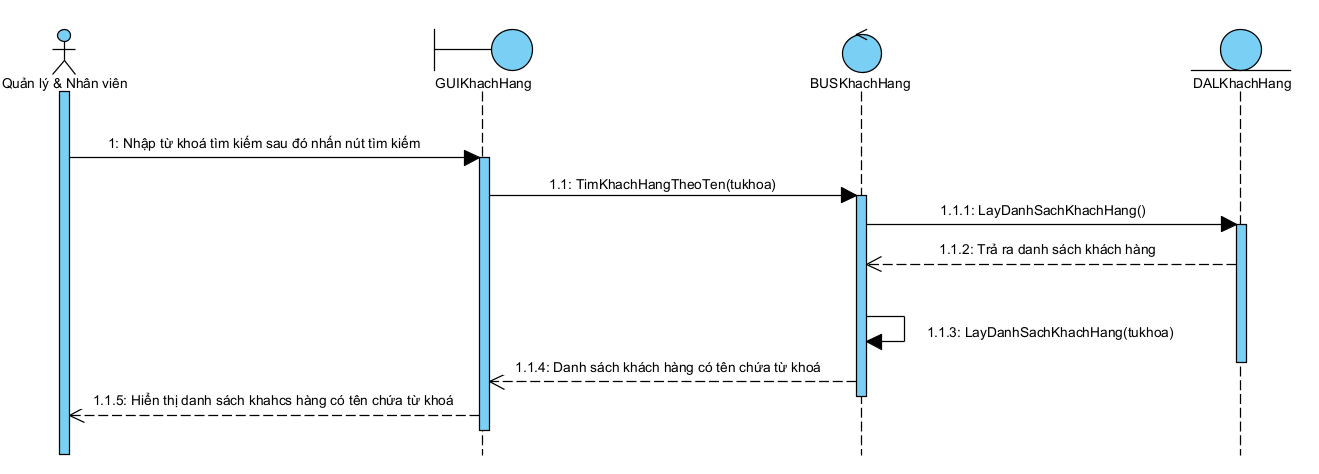
Hình 3.22 Biểu đồ tuần tự thêm khách hàng



Hình 3.23 Biểu đồ tuần tự sửa khách hàng

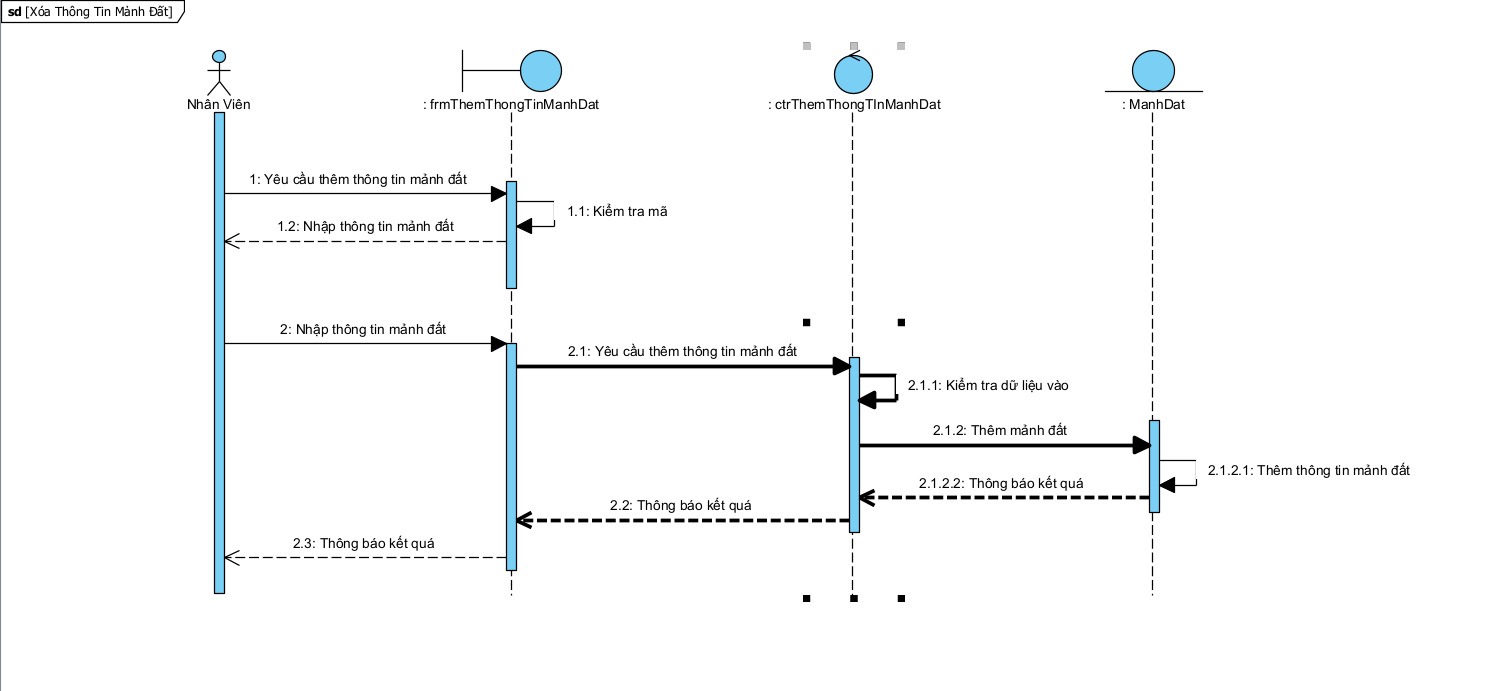


Hình 3.24 Biểu đồ tuần tự xoá khách hàng

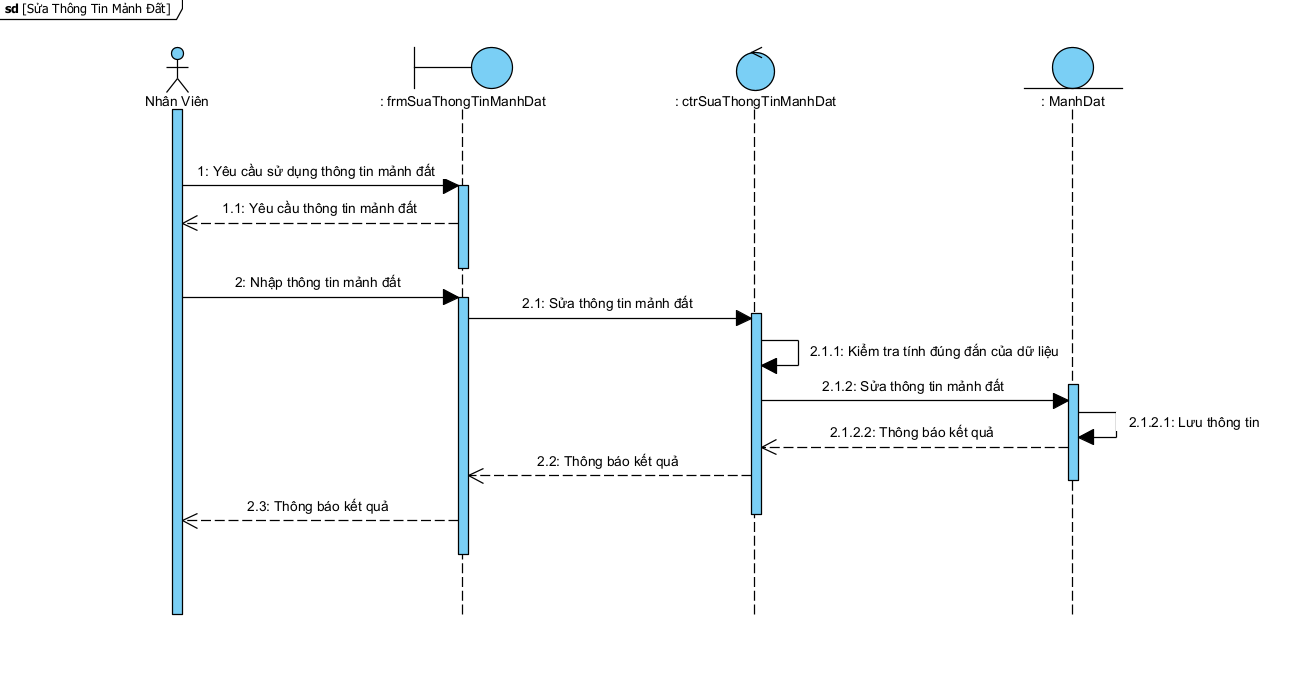


Hình 3.25 Biểu đồ tuần tự tìm khách hàng theo tên

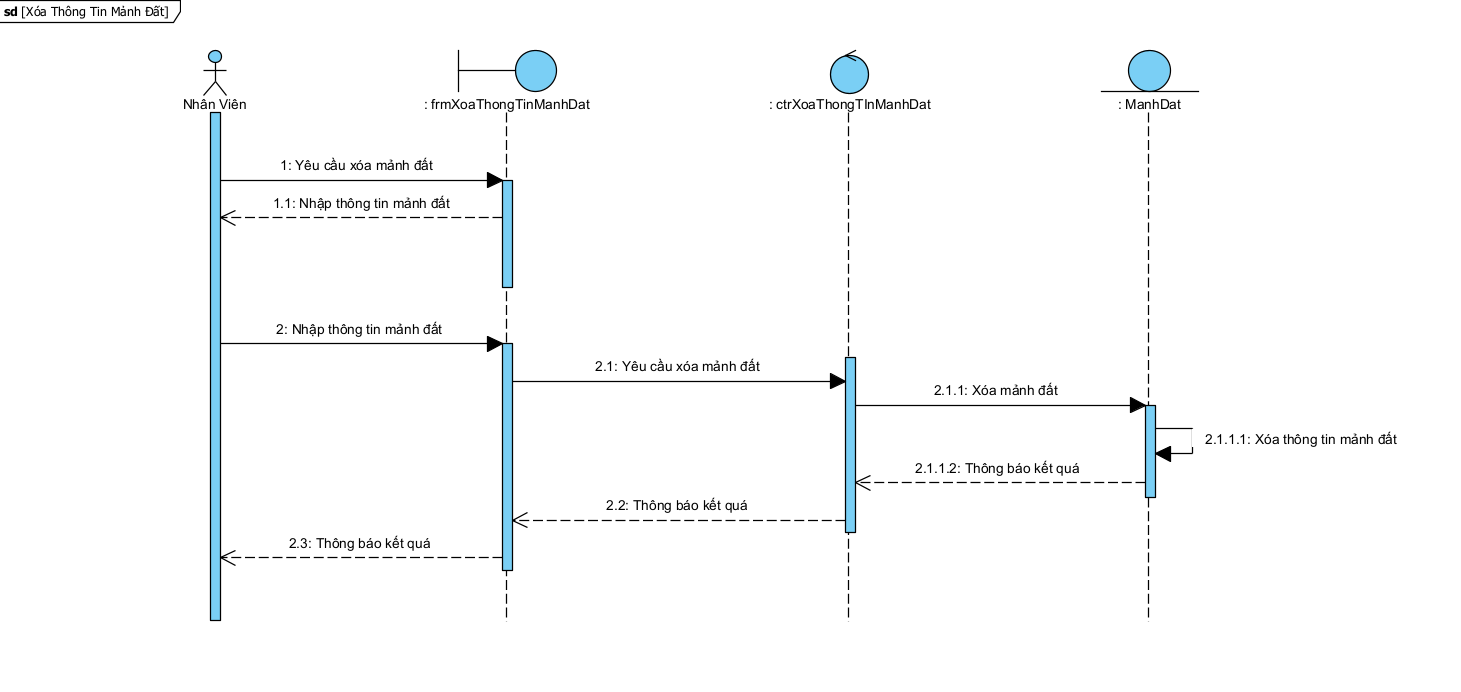
### Biểu đồ tuần tự ca sử dụng Quản lý mảnh đất



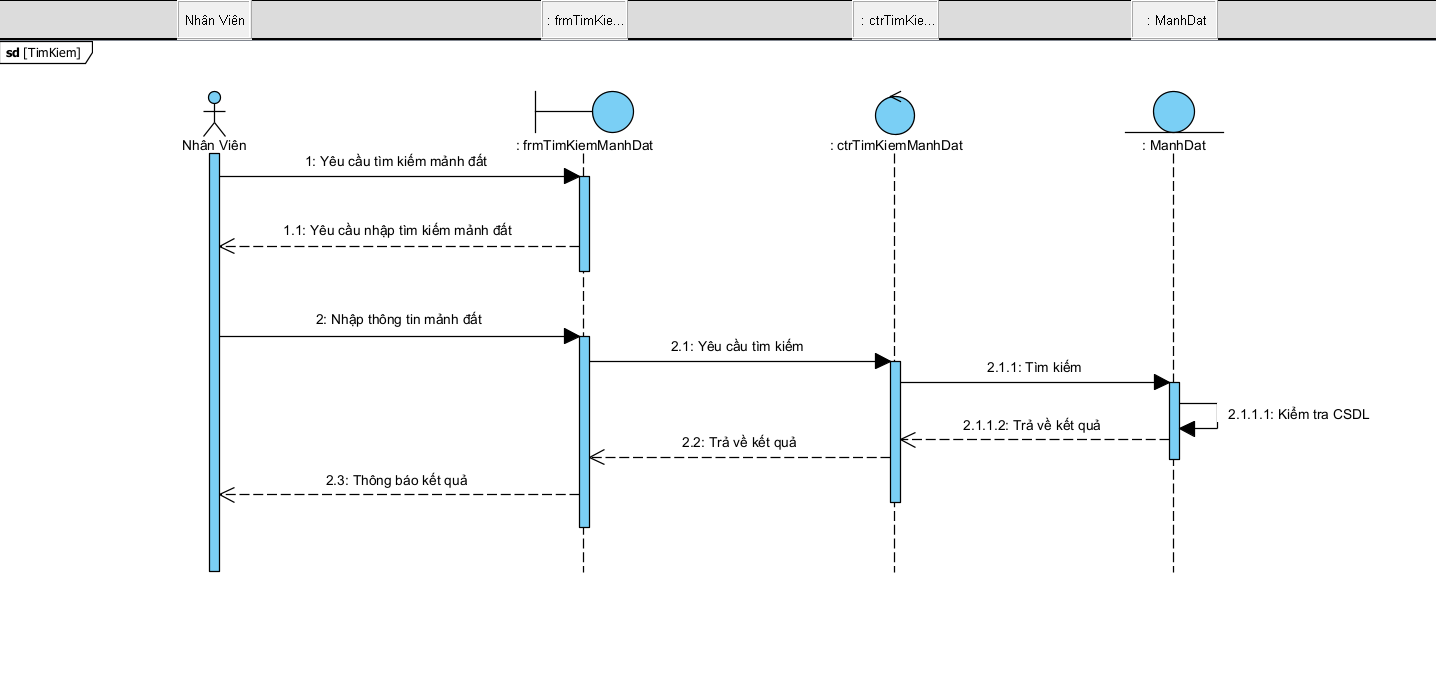
Hình 3.27 Biểu đồ tuần tự thêm thông tin mảnh đất



Hình 3.28 Biểu đồ tuần tự sửa thông tin mảnh đất

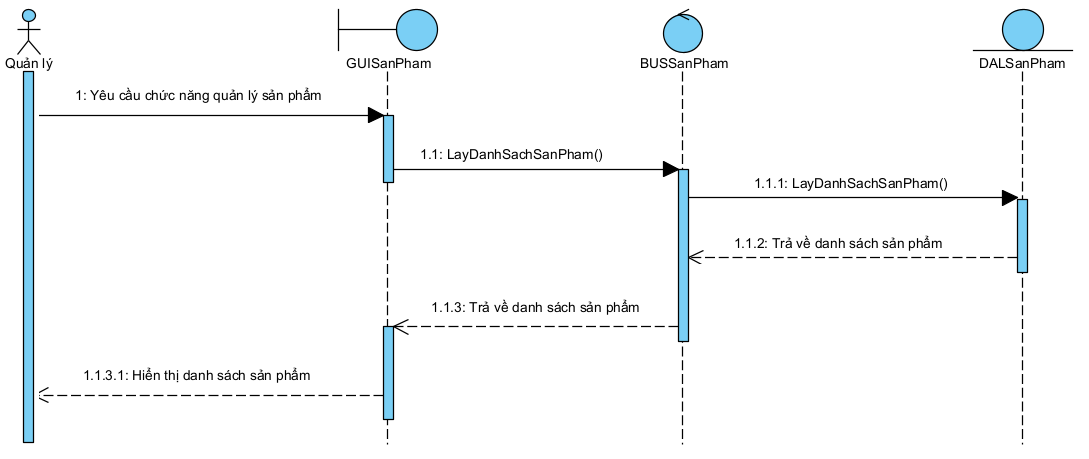


Hình 3.29 Biểu đồ tuần tự xoá thông tin mảnh đất

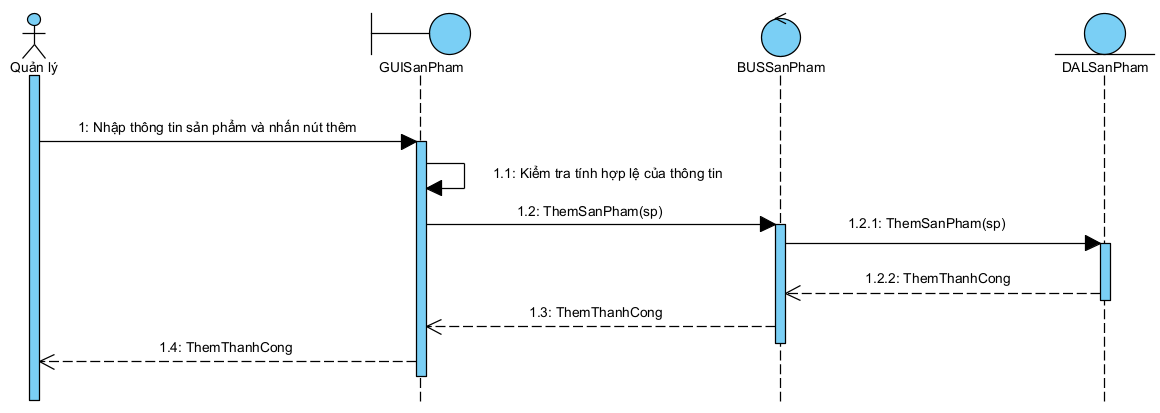


Hình 3.30 Biểu đồ tuần tự tìm mảnh đất theo tên

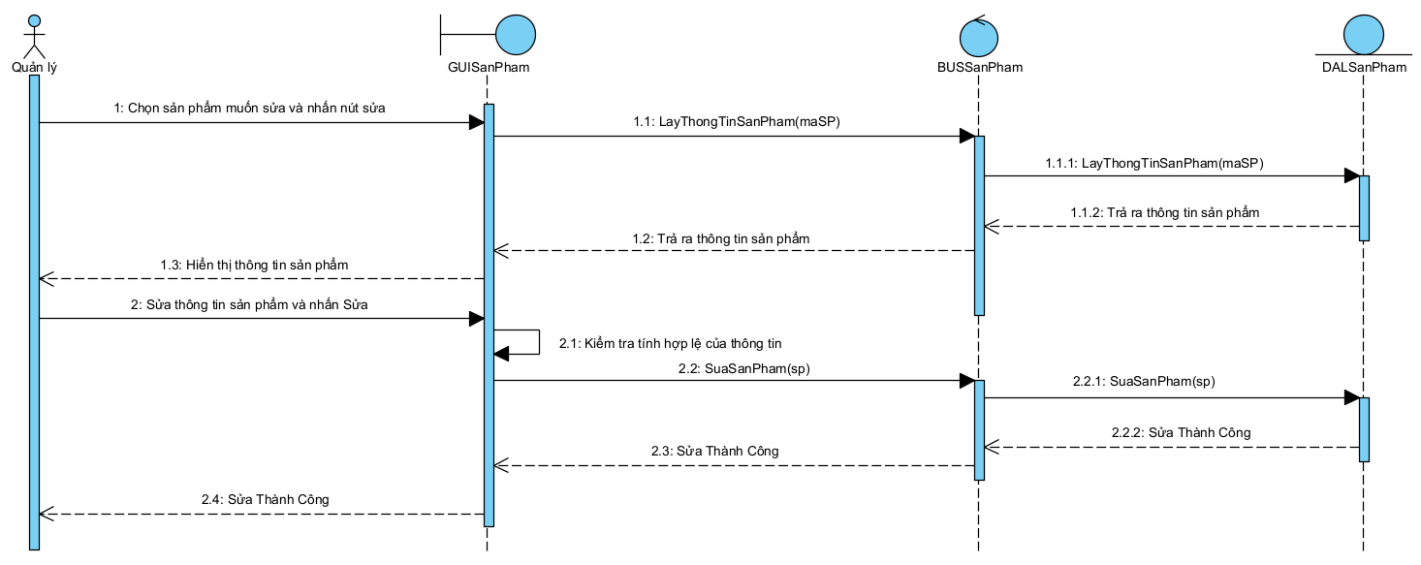
### Biểu đồ tuần tự ca sử dụng Quản lý mảnh đất(sản phẩm của văn phòng)



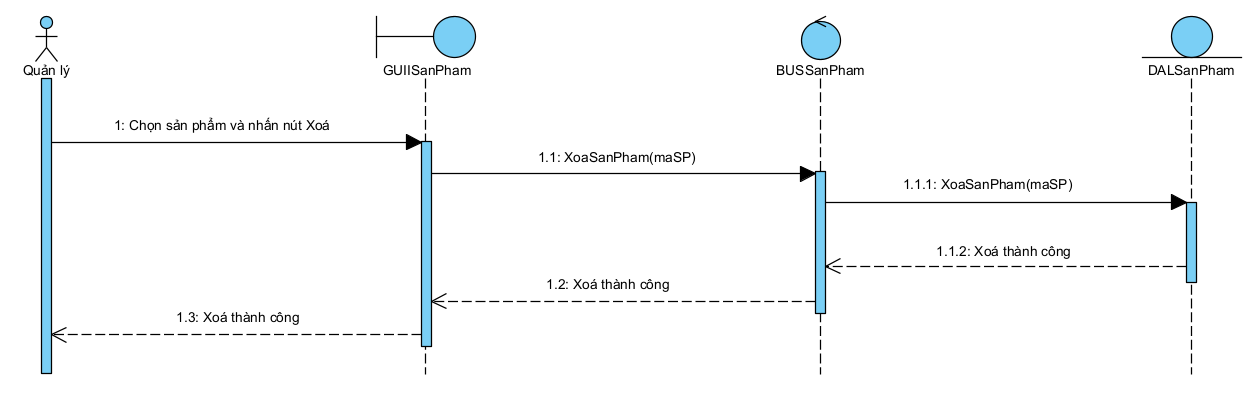
Hình 3.36 Biểu đồ tuần tự tác vụ hiển thị thông tin mảnh đất



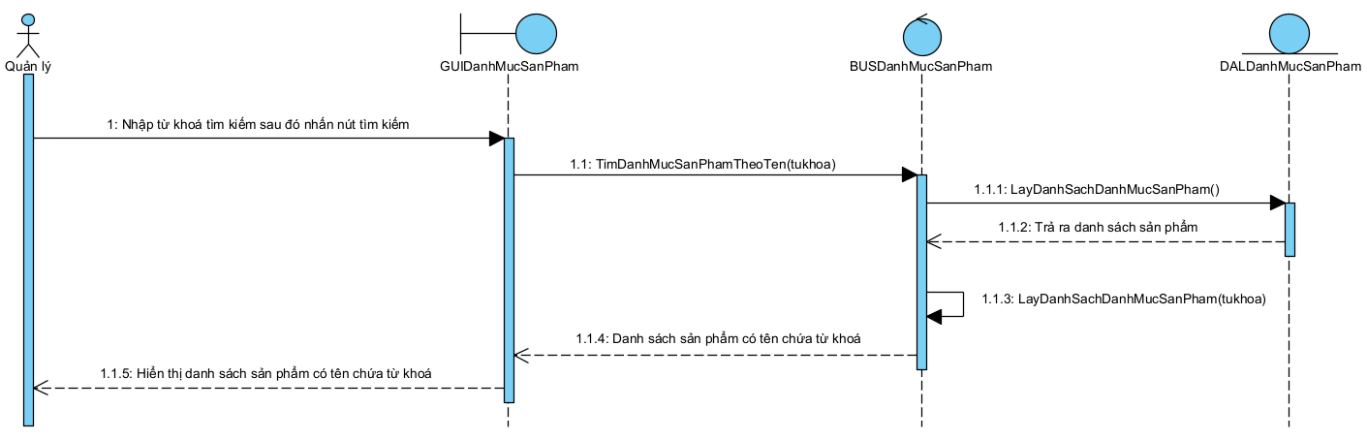
Hình 3.37 Biểu đồ tuần tự thêm thông tin mảnh đất



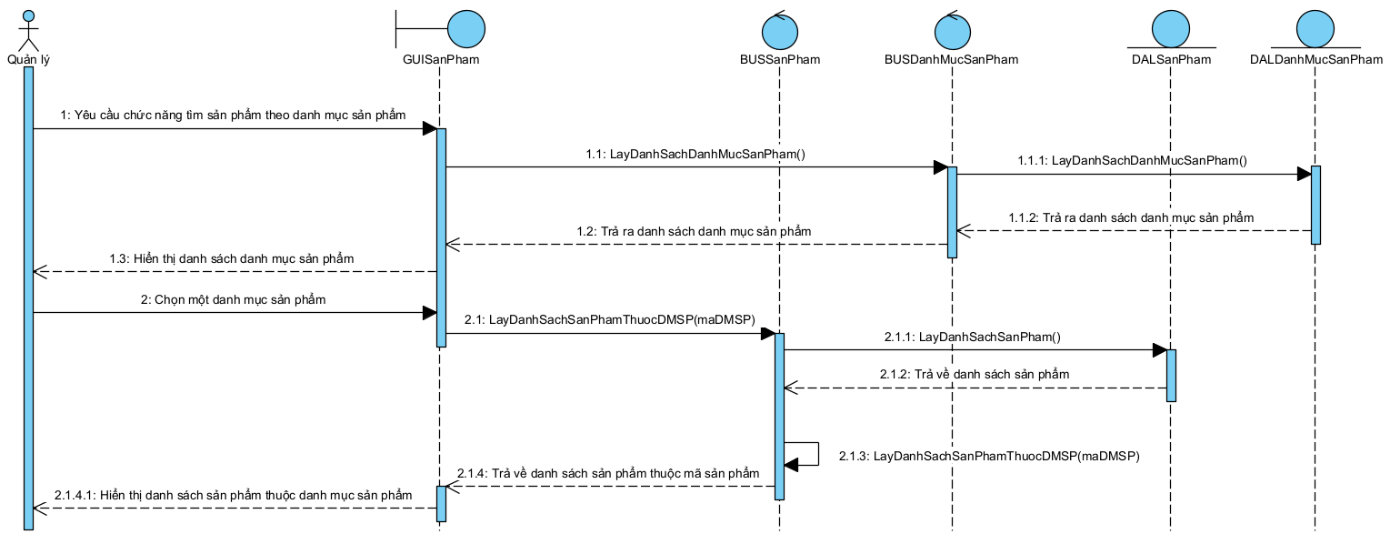
Hình 3.38 Biểu đồ tuần tự sửa thông tin mảnh đất



Hình 3.39 Biểu đồ tuần tự xoá thông tin mảnh đất

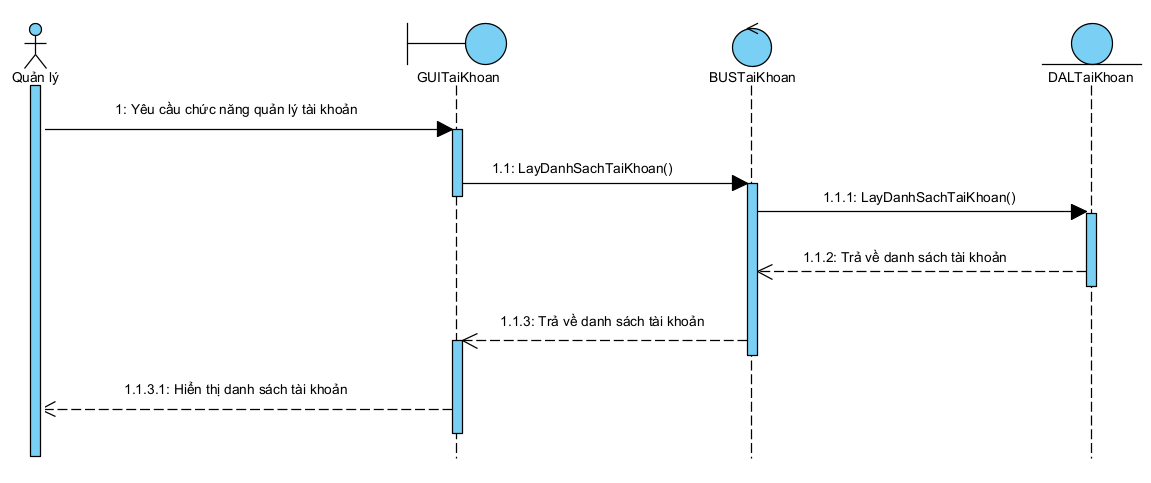


Hình 3.40 Biểu đồ tuần tự tìm mảnh đất theo tên

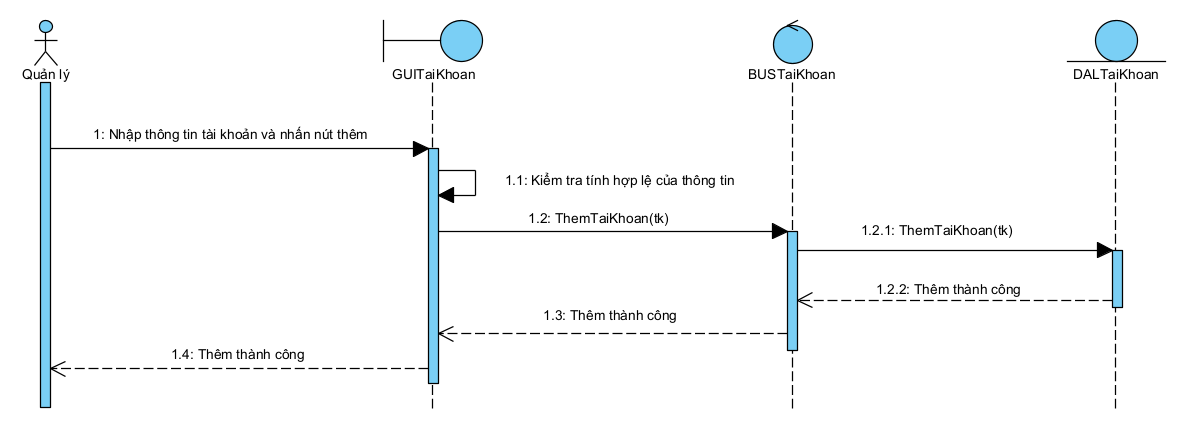


Hình 3.41 Biểu đồ tuần tự tìm mảnh đất theo danh mục mảnh đất

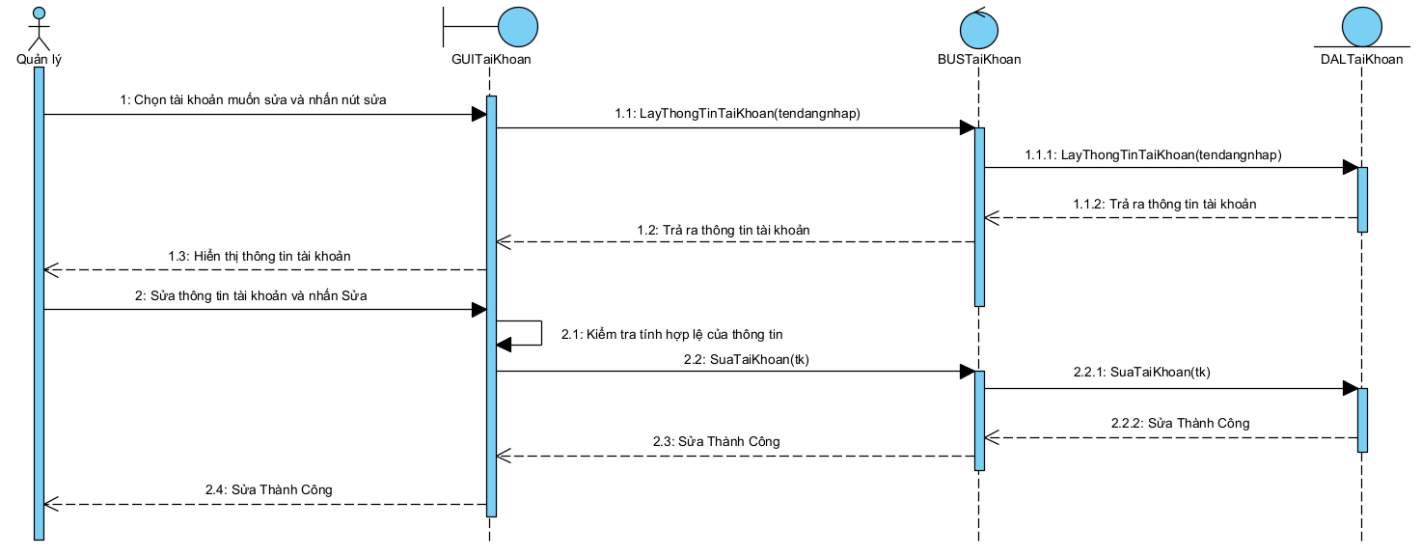
### Biểu đồ tuần tự ca sử dụng Quản lý tài khoản



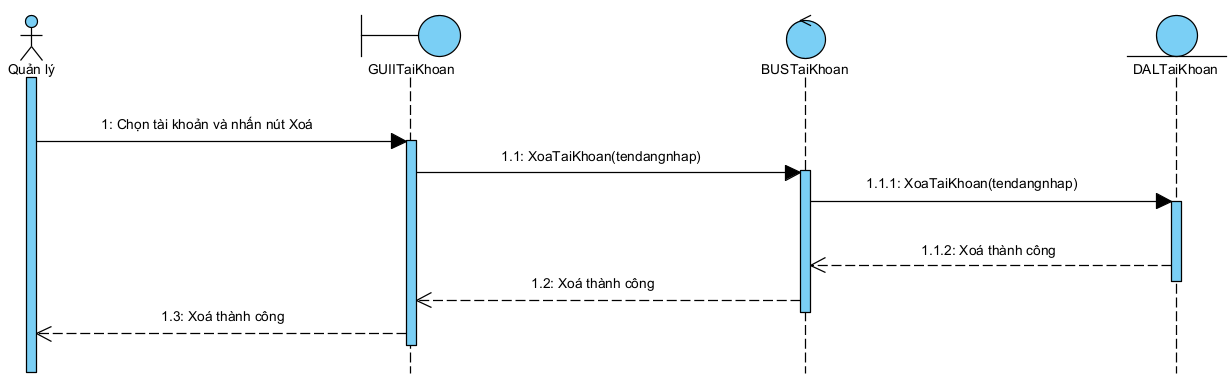
Hình 3.42 Biểu đồ tuần tự tác vụ hiển thị thông tin tài khoản



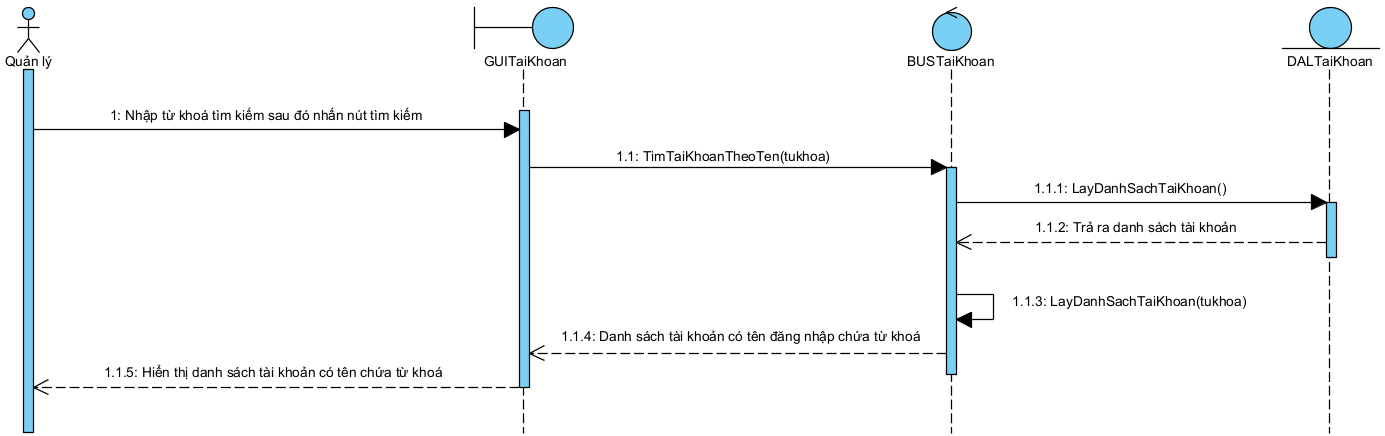
Hình 3.43 Biểu đồ tuần tự thêm thông tin tài khoản



Hình 3.44 Biểu đồ tuần tự sửa thông tin tài khoản



Hình 3.45 Biểu đồ tuần tự xoá thông tin tài khoản



Hình 3.46 Biểu đồ tuần tự tìm tài khoản theo tên đăng nhập

### Biểu đồ tuần tự ca sử dụng Thống kê doanh thu

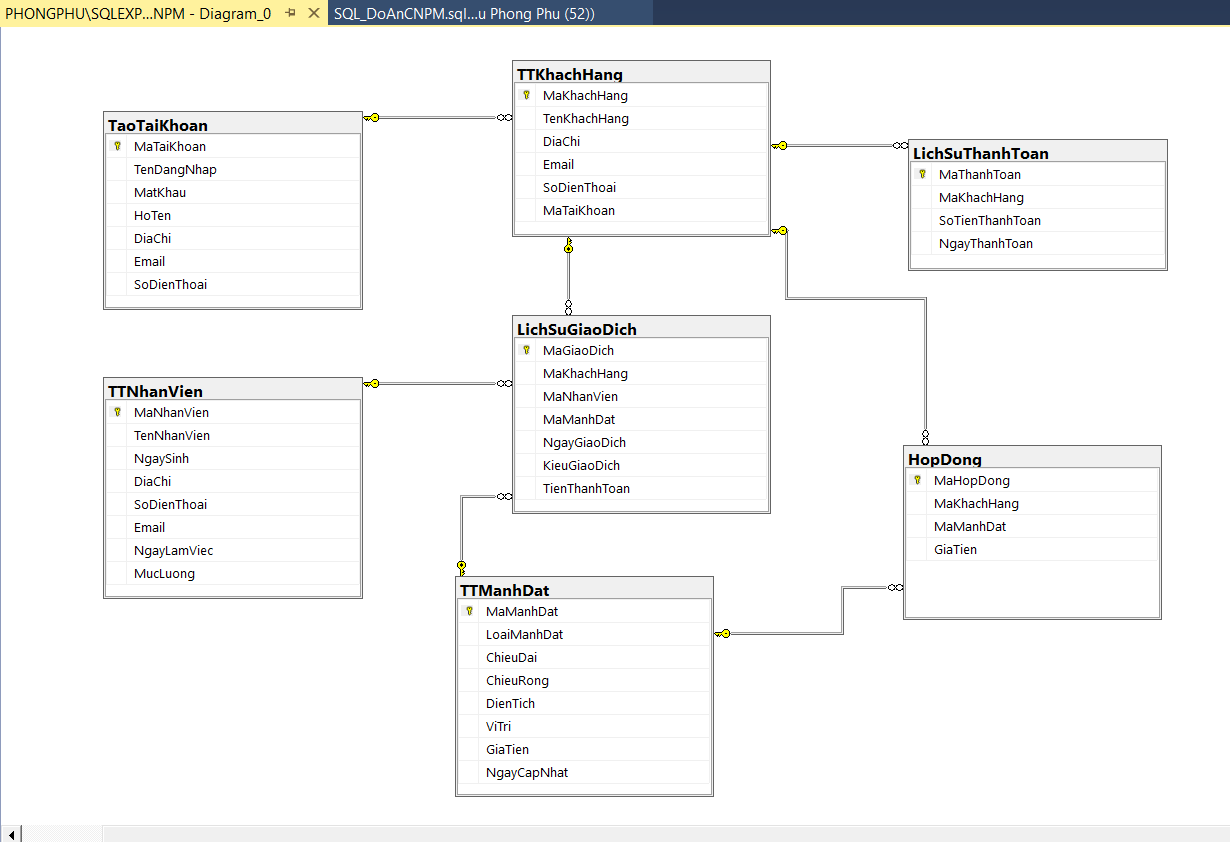
Ảnh có chứa văn bản, biểu đồ, hàng, Song song

Mô tả được tạo tự động

Hình 3.47 Biểu đồ tuần tự thống kê doanh thu

## Thiết kế CSDL [3] [1]

### Lược đồ CSDL



Hình 3.48 Diagram CSDL hệ thống

### Mô tả chi tiết từng bảng trong CSDL

*Bảng 2.1 Bảng Tạo Tài Khoản*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **1** | MaTaiKhoan | Varchar | Khóa chính | Mã để quản lý tài khoản |
| **2** | TenDangNhap | Varchar |  | Là tên đăng nhập của tài khoản |
| **3** | MatKhau | Varchar |  | Là mật khẩu của tài khoản |
| **4** | HoTen | Nvarchar |  | Tên người dùng |
| **5** | DiaChi | Nvarchar |  | Địa chỉ người dùng |
| **6** | Email | Varchar |  | Địa chỉ thư điện tử người dùng |
| **7** | SoDienThoai | Varchar |  | Số điện thoại cá nhân của người dùng |

*Bảng 2.2 Bảng Thông Tin Khách Hàng*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **1** | MaKhachHang | Varchar | Khóa chính | Là mã để quản lý một khách hàng nào đó |
| **2** | TenKhachHang | Nvarchar |  | Là tên của người dùng |
| **3** | Email | Varchar |  | Là địa chỉ điện tử của người dùng |
| **4** | DiaChi | Nvarchar |  | Là địa chỉ nơi chốn của người dùng |
| **5** | SoDienThoai | Varchar |  | Số điện thoại cá nhân của người dùng |
| **6** | MaTaiKhoan | Varchar | Khóa phụ | Để liên kết với bảng Tài Khoản |

*Bảng 2.3 Bảng Thông Tin Nhân Viên*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **1** | MaNhanVien | Varchar | Khóa chính | Mã để quản lý nhân viên |
| **2** | TenNhanVien | Nvarchar |  | Tên của nhân viên |
| **3** | NgaySinh | Date |  | Kiểm tra độ tuổi của nhân viên |
| **4** | DiaChi | Nvarchar |  | Địa chỉ nơi ở của nhân viên |
| **5** | SoDienThoai | Varchar |  | Số điện thoại cá nhân của nhân viên |
| **6** | Email | Varchar |  | Địa chỉ thư điện tử của nhân viên |
| **7** | NgayLamViec | Date |  | Ngày làm việc chấm công |
| **8** | MucLuong | Money |  | Mức lương cơ bản của nhân viên |

*Bảng 2.4 Thông Tin Mảnh Đất*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **1** | MaManhDat | Varchar | Khóa chính | Là mã để quản lý mảnh đất nào đó |
| **2** | LoaiManhDat | Varchar |  | Loại mảnh đất ứng với giá thành khác nhau |
| **3** | ChieuDai | Float |  | Độ dài của mảnh đất |
| **4** | ChieuRong | Float |  | Độ rộng của mảnh đất |
| **5** | DienTich | Float |  | Diện tích của mảnh đất |
| **6** | ViTri | Nvarchar |  | Địa điểm vị trí cụ thể |
| **7** | GiaTien | Money |  | Giá tiền của mảnh đất |
| **8** | NgayCapNhat | Date |  | Ngày cập nhật lại thông tin |

*Bảng 2.5 Bảng Hợp Đồng*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **1** | MaHopDong | Varchar | Khóa chính | Là mã quản lý hợp đồng giữa người mua và người bán |
| **2** | MaKhachHang | Varchar | Khóa phụ | Được liên kết với bảng thông tin khách hàng |
| **3** | MaManhDat | Varchar | Khóa phụ | Được liên kết với bảng thông tin mã mảnh đất |
| **4** | GiaTien | Money |  | Giá tiền chốt giá cuối để giao dịch |

*Bảng 2.6 Bảng Lịch Sử Thanh Toán*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **1** | MaThanhToan | Varchar | Khóa chính | Là mã quản lý hóa đơn thanh toán |
| **2** | MaKhachHang | Varchar | Khóa phụ | Được liên kết với bảng thông tin khách hàng |
| **3** | SoTienThanhToan | Float |  | Số tiền thanh toán giao dịch giữa 2 bên |
| **4** | NgayThanhToan | Date |  | Ngày xác nhận thanh toán giao dịch |

*Bảng 2.7 Bảng Lịch Sử Giao Dịch*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **1** | MaGiaoDich | Varchar | Khóa chính | Mã để quản lý tài khoản |
| **2** | MaKhachHang | Varchar | Khóa phụ | Là mã quản lý một khách hàng nào đó |
| **3** | MaNhanVien | Varchar | Khóa phụ | Là mã quản lý một nhân viên nào đó |
| **4** | MaManhDat | Varchar | Khóa phụ | Là mã quản lý một mảnh đất nào đó |
| **5** | NgayGiaoDich | Date |  | Quản lý ngày giao dịch giữa người mua và người bán |
| **6** | KieuGiaoDich | Nvarchar |  | Có thể thanh toán trực tiếp tại quầy hoặc thanh toán trực tiếp trên ứng dụng |
| **7** | TienThanhToan | Money |  | Là loại tiền tệ để trao đổi giữa người mua và người bán |

# KẾT LUẬN

**Kết quả đạt được**

* Hoàn thành chức năng quản lý thông tin đất thổ cư, bao gồm các thông tin về vị trí, diện tích, trạng thái pháp lý, giá bán, chủ sở hữu, vv.
* Thực hiện được chức năng tìm kiếm và tra cứu thông tin đất thổ cư một cách chính xác và nhanh chóng.
* Quản lý giao dịch bán đất thổ cư hiệu quả, từ việc lập hợp đồng, thanh toán đến việc cập nhật thông tin sau khi giao dịch hoàn tất.
* Tự động tạo ra các báo cáo thống kê về số lượng đất thổ cư đã bán ra, doanh thu thu được, số lượng đất còn lại, trạng thái pháp lý, vị trí đất, diện tích, vv.
* Cung cấp chức năng quản lý người dùng để chỉ cho phép các người dùng được phép truy cập và sửa đổi dữ liệu.

**Hạn chế của đề tài**

* Độ chính xác của dữ liệu: Dữ liệu về các mảnh đất, thông tin giao dịch, thông tin người dùng,.. có thể bị sai sót hoặc thiếu sót. Điều này có thể ảnh hưởng đến tính chính xác của kết quả thống kê, báo cáo và các quyết định kinh doanh.
* Hạn chế về kỹ thuật: Để triển khai chương trình quản lý bán đất thổ cư, cần sử dụng các công nghệ phần mềm và kỹ thuật lập trình. Việc thiếu hiểu biết hoặc kỹ năng về các công nghệ này có thể ảnh hưởng đến khả năng triển khai và bảo trì chương trình.
* Hạn chế về nguồn lực: Triển khai chương trình quản lý bán đất thổ cư ở tỉnh Hưng Yên yêu cầu đòi hỏi một số nguồn lực như máy tính, phần mềm và nhân lực để triển khai và duy trì chương trình. Điều này có thể là một rào cản đối với các tổ chức hoặc doanh nghiệp nhỏ không có ngân sách đầu tư đủ lớn.
* Hạn chế về quy định pháp lý: Quản lý bán đất thổ cư liên quan đến nhiều quy định pháp lý khác nhau như quy định về quyền sở hữu, đất đai, thuế và các quy định liên quan khác. Việc không tuân thủ các quy định này có thể dẫn đến hậu quả pháp lý nghiêm trọng cho các tổ chức hoặc doanh nghiệp.

**Hướng phát triển**

* Tích hợp công nghệ blockchain: Tích hợp công nghệ blockchain vào ứng dụng sẽ giúp tăng tính an toàn và bảo mật của dữ liệu, đồng thời giảm thiểu việc gian lận, giả mạo dữ liệu trong quá trình giao dịch.
* Kết nối với hệ thống GIS: Kết nối với hệ thống thông tin địa lý (GIS) để có thể hiển thị thông tin đất trên bản đồ, giúp người dùng dễ dàng tìm kiếm và tra cứu thông tin về vị trí đất cũng như giúp quản lý dữ liệu đất được chính xác hơn.
* Phát triển ứng dụng di động: Xây dựng ứng dụng di động giúp người dùng có thể truy cập thông tin và thực hiện giao dịch bất động sản bất cứ khi nào và ở bất kỳ đâu, đồng thời cải thiện trải nghiệm người dùng.
* Tích hợp hệ thống thanh toán trực tuyến: Tích hợp các hệ thống thanh toán trực tuyến vào ứng dụng giúp người dùng có thể thanh toán và giao dịch trực tuyến một cách nhanh chóng và tiện lợi.
* Phát triển chức năng hỗ trợ pháp lý: Xây dựng chức năng hỗ trợ pháp lý giúp người dùng có thể kiểm tra và đánh giá tính pháp lý của đất trước khi thực hiện giao dịch.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | K. CNTT, Phân tích thiết kế phần mềm, Hưng Yên: Đại học SPKT Hưng Yên, 2009. |
| [2] | K. CNTT, Cơ sở dữ liệu, Hưng Yên: Đại học SPKT Hưng Yên, 2012. |
| [3] | K. CNTT, Lập trình ứng dụng windows forms, Hưng Yên: Đại học SPKT Hưng Yên, 2019. |
| [4] | W3schools, "W3schools," [Online]. Available: https://www.w3schools.com/. |
| [5] | "Quản lý bán đất thổ cư". https://muaban.net/bat-dong-san/ban-dat-tho-cu-ha-noi |
| [6] | "Thiết kế cơ sở dữ liệu". https://iviettech.vn/blog/10578-thiet-ke-co-so-du-lieu-cac-buoc-thuc-hien.html |
| [7] | "Lập trình ứng dụng Winforms". https://devmaster.edu.vn/lap-trinh-ung-dung-Windows-Forms-C-sharp-devmaster-winform.html |
| [8] | "Hướng dẫn sử dụng SQL Server". https://openplanning.net/10239/huong-dan-hoc-sql-cho-nguoi-moi-bat-dau-voi-sql-server |
| [9] | "Báo cáo thống kê bất động sản". https://baochinhphu.vn/vars-cong-bo-bao-cao-thi-truong-bat-dong-san-quy-ii-2022-102220715123728657.htm |
| [10] | "Thiết kế giao diện người dùng". https://howkteam.vn/course/phan-tich-thiet-ke-phan-mem/thiet-ke-giao-dien-nguoi-dung-trong-pttkpm-1421 |
| [11] | K. CNTT, Kiểm thử phần mềm, Hưng Yên: Đại học SPKT Hưng Yên, 2016. |