BỘ GIÁO DỤC ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC THĂNG LONG**

---o0o---

Logo, company name

Description automatically generated

**CHUYÊN ĐỀ TỐT NGHIỆP**

**HỆ THỐNG CỔNG THÔNG TIN ĐẠI HỌC THĂNG LONG**

**SINH VIÊN THỰC HIỆN: Lê Nhật Cương - A30168 Nguyễn Hữu Đại – A30406**

**CHUYÊN NGÀNH: Khoa học máy tính**

**GIÁO VIÊN HƯỚNG DẪN: Mai Thúy Nga**

**HÀ NỘI - 2021**

**LỜI CẢM ƠN**

Lời đầu tiên em xin phép gửi lời cảm ơn chân thành nhất tới toàn bộ các thầy cô giáo trong khoa Toán – Tin cũng như các thầy cô giảng dạy trong trường Đại học Thăng Long đã truyền đạt những kiến thức quý báu và bổ ích cho em trong những năm học vừa qua.

Chuyên đề tốt nghiệp là cơ hội để chúng em có thể áp dụng, tổng kết lại những kiến thức mà mình đã học. Đồng thời, rút ra được những kinh nghiệm thực tế quý giá trong suốt quá trình thực hiện dự án. Sau một thời gian tập chung công sức cho dự án, đặc biệt với sự hỗ trợ nhiệt tình của cô Mai Thuý Nga, đã giúp chúng em hoàn thành dự án một cách thuận lợi và gặt hái được những kết quả mong muốn. Bên cạnh những kết quả khiêm tốn mà em đã đạt được, chắc chắn không tránh khỏi những thiếu sót khi thực hiện dự án của mình. Sự phê bình, góp ý của quý thầy cô sẽ là những bài học kinh nghiệm quý báu cho sự nghiệp tương lai của chúng em.

Vì kiến thức còn hạn chế, trong quá trình triển khai và thực hiện CĐTN em không tránh khỏi những sai sót không mong muốn, chính vì vậy em rất mong nhận được ý kiến đóng góp từ các thầy cô cũng như các bạn trong khoa để hoàn thiện CĐTN này cũng như bổ sung những kiến thức còn thiếu sót.

Cuối cùng em xin gửi lời chúc tới toàn thể các thầy cô trong khoa Toán – Tin cũng như các thầy cô giảng dạy trong trường Đại học Thăng Long luôn luôn mạnh khỏe và tràn đầy nhiệt huyết để truyền đạt kiến thức cho các thế hệ sinh viên mai sau.

|  |  |
| --- | --- |
|  | Hà Nội, ngày tháng năm 2021  Sinh viên |

**LỜI GIỚI THIỆU**

Chúng ta đang sống trong một thời đại mới, thời đại phát triển rực rỡ của CNTT. CNNT đã ở một bước phát triển cao đó là số hóa tất cả các dữ liệu thông tin, luân chuyển mạnh mẽ và kết nối tất cả chúng ta lại với nhau. Mọi loại thông tin, số liệu âm thanh, hình ảnh có thể đưa về dạng kỹ thuật số để bất kỳ máy tính nào cũng có thể lưu trữ, xử lý và chuyển tiếp cho nhiều người. Những công cụ và sự kết nối của thời đại kỹ thuật số cho phép chúng ta dễ dàng thu thập, chia sẻ thông tin và hành động trên cơ sở những thông tin này theo phương thức hoàn toàn mới, kéo theo hàng loạt sự thay đổi về các quan niệm, các tập tục, các thói quen truyền thống, và thậm chí cả cách nhìn các giá trị trong cuộc sống. Không có lĩnh vực nào, không có nơi nào không có mặt của CNTT.

Nhận thấy được những lợi ích của công nghệ thông tin, các trường đại học đã và đang áp dụng vào việc quản lý thông tin. Việc ứng dụng tin học vào quản lý cổng thông tin không phải là mới. Tuy nhiên, đối với Đại học Thăng Long đòi hỏi cần có một hệ thống đáp ứng được yêu cầu quản lý văn bản của nhà trường, quản lý bảng điểm của sinh viên, quản lý bài viết của cựu sinh viên và doanh nghiệp.

Xuất phát từ yêu cầu thực tế trên, em xin tìm hiểu và thực hiện đề tài “**Hệ thống cổng thông tin Đại học Thăng Long**” với các chức năng chính như: quản lý văn bản, quản lý bảng điểm, quản lý bài viết.

Cấu trúc của tài liệu:

**CHƯƠNG 1:** TỔNG QUAN HỆ THỐNG

**CHƯƠNG 2:** CÔNG NGHỆ SỬ DỤNG VÀ KIÊN TRÚC TỔNG THỂ

**CHƯƠNG 3:** ĐẶC TẢ CHỨC NĂNG CHÍNH CỦA HỆ THỐNG

**CHƯƠNG 4:** PHÂN TÍCH THIẾT KẾ CHỨC NĂNG CHÍNH

**CHƯƠNG 5:** THIẾT KẾ DATABASE VÀ API ĐẶC BIỆT

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN HỆ THỐNG 1](#_Toc81825888)

[1.1. Mô tả bài toán 1](#_Toc81825889)

[1.2. Phân tích bài toán 1](#_Toc81825890)

[1.3. Yêu cầu nghiệp vụ 1](#_Toc81825891)

[1.3.1. Sơ đồ tổng quan chức năng hệ thống 2](#_Toc81825892)

[1.3.2. Các tác nhân tham gia hệ thống 2](#_Toc81825893)

[1.3.3. Các chức năng chính của hệ thống 2](#_Toc81825894)

[CHƯƠNG 2. CÔNG NGHỆ SỬ DỤNG VÀ KIẾN TRÚC TỔNG THỂ 4](#_Toc81825895)

[2.1. Công nghệ sử dụng 4](#_Toc81825896)

[2.1.1. SQL – Server 4](#_Toc81825897)

[2.1.2. ORM - Entity Framework Core 4](#_Toc81825898)

[2.1.3. ASP.NET Core SignalR 5](#_Toc81825899)

[2.1.4. ClearScript 7](#_Toc81825900)

[2.1.5. ABP Framework 7](#_Toc81825901)

[2.2. Kiến trúc tổng thể 10](#_Toc81825902)

[2.2.1. Domain Driven Design 10](#_Toc81825903)

[2.2.2. Mô tả tổng quan 12](#_Toc81825904)

[2.2.3. Kiến trúc module của hệ thống theo DDD 13](#_Toc81825905)

[CHƯƠNG 3. ĐẶC TẢ CHỨC NĂNG CHÍNH CỦA HỆ THỐNG 20](#_Toc81825906)

[*3.1.* Quản lý máy client 20](#_Toc81825907)

[*3.2.* Quản lý script 21](#_Toc81825908)

[*3.3.* Thực thi script 25](#_Toc81825909)

[*3.4.* Cập nhật máy client 27](#_Toc81825910)

[*3.5.* Quản lý ứng dụng 28](#_Toc81825911)

[*3.6.* Cập nhật ứng dụng 30](#_Toc81825912)

[*3.7.* Quản lý nhóm quyền 32](#_Toc81825913)

[*3.8.* Quản lý người dùng 35](#_Toc81825914)

[*3.9.* Đăng nhập 41](#_Toc81825915)

[*3.10.* Đăng xuất 43](#_Toc81825916)

[*3.11.* Kết nối với hệ thống kiểm soát trạng thái 44](#_Toc81825917)

[*3.12.* Ngắt kết nối với hệ thống kiểm soát trạng thái 45](#_Toc81825918)

[*3.13.* Hiển thị nội dung file log 47](#_Toc81825919)

[*3.14.* Gửi email danh sách ứng dụng ngừng hoạt động 48](#_Toc81825920)

[CHƯƠNG 4. PHÂN TÍCH THIẾT KẾ CHỨC NĂNG CHÍNH 50](#_Toc81825921)

[*4.1.* Quản lý máy client 50](#_Toc81825922)

[4.1.1. Sơ đồ lớp phân tích 50](#_Toc81825923)

[4.1.2. Sơ đồ trình tự 50](#_Toc81825924)

[4.2. Quản lý script 51](#_Toc81825925)

[4.2.1. Sơ đồ lớp phân tích 51](#_Toc81825926)

[4.2.2. Sơ đồ trình tự 51](#_Toc81825927)

[4.3. Thực thi script 53](#_Toc81825928)

[4.3.1. Sơ đồ lớp phân tích 53](#_Toc81825929)

[4.3.2. Sơ đồ trình tự 53](#_Toc81825930)

[4.4. Cập nhật máy client 53](#_Toc81825931)

[4.4.1. Sơ đồ lớp phân tích 53](#_Toc81825932)

[4.4.2. Sơ đồ trình tự 54](#_Toc81825933)

[4.5. Quản lý ứng dụng 54](#_Toc81825934)

[4.5.1. Sơ đồ lớp phân tích 54](#_Toc81825935)

[4.5.2. Sơ đồ trình tự 55](#_Toc81825936)

[4.6. Cập nhật ứng dụng 56](#_Toc81825937)

[4.6.1. Sơ đồ lớp phân tích 56](#_Toc81825938)

[4.6.2. Sơ đồ trình tự 56](#_Toc81825939)

[4.7. Kết nối hệ thống biên dịch hỗ trợ với hệ thống kiểm soát trạng thái 56](#_Toc81825940)

[4.7.1. Sơ đồ lớp phân tích 56](#_Toc81825941)

[4.7.2. Sơ đồ trình tự 57](#_Toc81825942)

[4.8. Ngắt kết nối với hệ thống kiểm soát trạng thái 57](#_Toc81825943)

[4.8.1. Sơ đồ lớp phân tích 57](#_Toc81825944)

[4.8.2. Sơ đồ trình tự 57](#_Toc81825945)

[4.9. Gửi email danh sách ứng dụng ngừng hoạt động 58](#_Toc81825946)

[4.9.1. Sơ đồ lớp phân tích 58](#_Toc81825947)

[4.9.2. Sơ đồ trình tự 58](#_Toc81825948)

[CHƯƠNG 5. THIẾT KẾ DATABASE VÀ API ĐẶC BIỆT 59](#_Toc81825949)

[5.1. Thiết kế cơ sở dữ liệu 59](#_Toc81825950)

[5.2. Định nghĩa database 59](#_Toc81825951)

[5.2.1. Bảng thông tin máy client - ClientMachines 59](#_Toc81825952)

[5.2.2. Bảng thông tin script – Scripts 60](#_Toc81825953)

[5.2.3. Bảng quan hệ script với script phụ thuộc – Dependencies 60](#_Toc81825954)

[5.2.4. Bảng thông tin ứng dụng – Tools 60](#_Toc81825955)

[5.2.5. Bảng thông tin người dùng – Users 61](#_Toc81825956)

[5.2.6. Bảng thông tin nhóm quyền – Roles 61](#_Toc81825957)

[5.2.7. Bảng quan hệ người dùng với nhóm quyền – UserRoles 61](#_Toc81825958)

[5.3. Mô hình kiến trúc API 62](#_Toc81825959)

[5.4. Thiết kế API chi tiết 62](#_Toc81825960)

[5.4.1. Quản lý máy client 62](#_Toc81825961)

[5.4.2. Quản lý script 63](#_Toc81825962)

[5.4.3. Quản lý ứng dụng 63](#_Toc81825963)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI 65](#_Toc81825964)

[TÀI LIỆU THAM KHẢO 66](#_Toc81825965)

**DANH MỤC MINH HỌA**

**DANH MỤC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**GIẢI THÍCH CÁC THUẬT NGỮ**

|  |  |
| --- | --- |
| **Thuật ngữ** | **Diễn giải** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# TỔNG QUAN HỆ THỐNG

## Mô tả bài toán

Đại học Thăng Long là một trong những trường Đại học đi đầu trong việc ứng dụng công nghệ thông tin. Hoạt động chính là tuyển sinh và giảng dạy cho sinh viên. Hiện tại Đại học Thăng Long đang sử dụng rất nhiều kênh thông tin phục vụ mục đích truyền tải các thông báo của nhà trường đến sinh viên. Những thông báo này bao gồm các nội dung: Tuyển sinh, đăng ký học, đào tạo, thu học phí, việc làm, thi,… . Do sử dụng nhiều kênh thông tin nên sinh viên dễ bị nhầm lẫn với những thông tin đã cũ hoặc những thông tin không chính thống. Để tiếp cận được những thông tin chính thống nhanh nhất và chính xác nhất, trường Đại học Thăng Long cần một hệ thống cổng thông tin.

Hệ thống cổng thông tin của Đại học Thăng Long đáp ứng được những yêu cầu sau: Cho phép sinh viên có thể xem được tất cả những thông tin (thông báo) của nhà trường, xem kết quả học tập, xem thông tin tuyển dụng.

Chính vì vậy, sự ra đời của “Hệ thống cổng thông tin Đại học Thăng Long” sẽ giải quyết được các vấn đề trên của Đại học Thăng Long. Hệ thống sẽ giúp việc quản lý, kiểm duyệt nội dung để sinh viên được tiếp cận với thông tin chính xác và nhanh nhất.

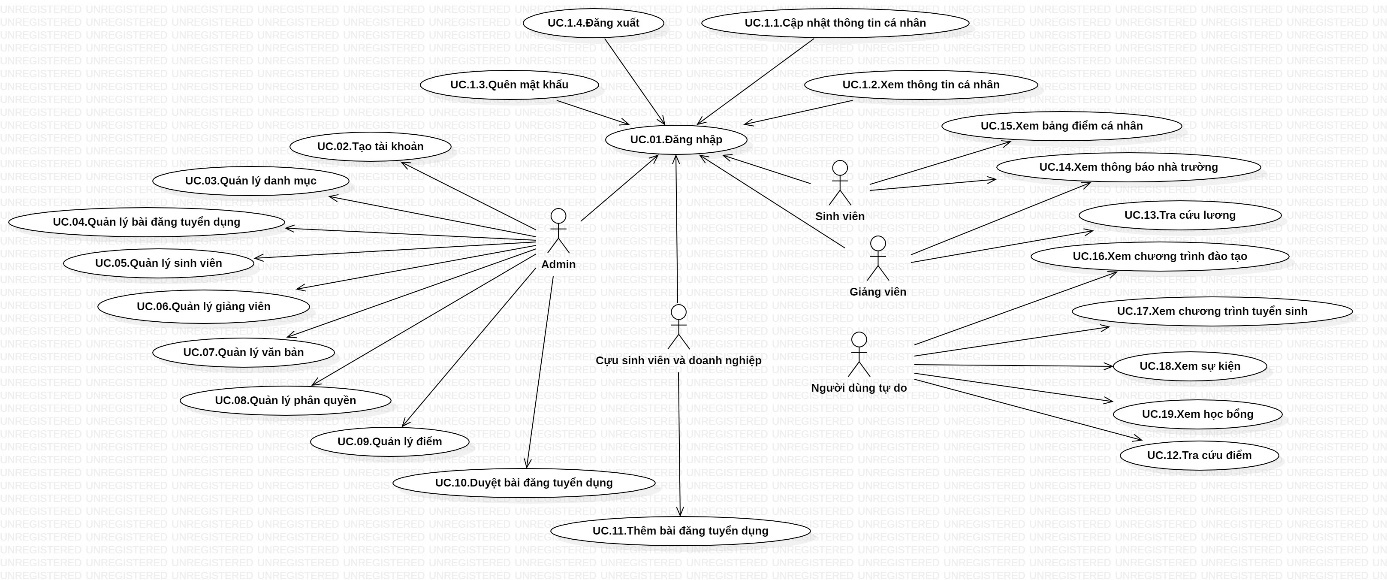
## Phân tích bài toán

* Phạm vi dự án: Cổng thông tin điện tử tại Đại học Thăng Long
* Nhóm chức năng chính: Hiển thị thông báo của nhà trường, hiển thị bảng điểm của sinh viên.
* Nhóm chức năng phục vụ việc chuẩn bị dữ liệu để tạo bài viết về thông báo của nhà trường: Thêm, sửa, xóa bài viết về thông báo của nhà trường.
* Nhóm chức năng phục vụ việc chuẩn bị dữ liệu để hiển thị bảng điểm của sinh viên: đọc dữ liệu từ hệ thống khác(file exel).

## Yêu cầu nghiệp vụ

* BR1: Đăng nhập hệ thống: Hệ thống phải đảm bảo khả năng đăng nhập, đăng xuất, chỉnh sửa thông tin tài khoản của người dùng.
* BR2: Tạo tài khoản: Hệ thống phải đảm bào người quản trị có khả năng tạo tài khoản để cấp cho người dùng hệ thống.
* BR3: Xây dựng cấu trúc trang web. Hệ thống phải đảm bảo khả năng quản lý (hiển thị, thêm, sửa, xóa) các thành phần cấu trúc trang web. Bao gồm: quản lý danh mục dành cho sinh viên tương lại, sinh viên, cựu sinh viên, giảng viên, phụ huynh
* BR4: Quản lý bài đăng tuyển dụng. Hệ thống phải đảm bảo khả năng quản lý (thêm, sửa, xóa) bài đăng tuyển dụng của cựu sinh viên và doanh nghiệp.
* BR4.1: Phê duyệt bài đăng tuyển dụng. Bài đăng của cựu sinh viên và doanh nghiệp được hiển thị khi người quản trị hệ thống phê duyệt.
* BR5: Quản lý văn bản. Hệ thống phải đảm bảo khả năng quản lý (hiển thị, thêm, sửa, xóa) văn bản của nhà trường.
* BR6: Quản lý sinh viên. Hệ thống phải đảm bảo khả năng quản lý(thêm, sửa, xoá) thông tin của sinh viên.
* BR6.1: Quản lý điểm của sinh viên. Hệ thống phải đảm bảo khả năng quản lý (hiển thị, đọc dữ liệu từ file exel, cập nhật) để hiển thị điểm của sinh viên.
* BR6.2: Xem bảng điểm cá nhân. Hệ thống phải đảm bảo sinh viên có quyền được xem bảng điểm cá nhân.
* BR7: Quản lý giảng viên. Hệ thống phải đảm bảo khả năng quản lý(thêm, sửa, xoá) thông tin của giảng viên.
* BR7.1: Tra cứu bảng lương. Hệ thống phải đảm bảo giảng viên có quyền được tra cứu bảng lương của mình.
* BR8: Quản lý phân quyền. Hệ thống phải đảm bảo khả năng người quản trị có thể phân quyền cho các loại tài khoản có trong hệ thống.

### Sơ đồ tổng quan chức năng hệ thống

******

Hình 1.1. Sơ đồ tổng quan chức năng hệ thống

### Các tác nhân tham gia hệ thống

* Admin: Là người chịu trách nhiệm quản lý và được cấp toàn quyền thao tác với các modules có trong hệ thống.
* Sinh viên: Là người đang theo học tại trường Đại học Thăng Long, họ có quyền xem những thông báo của nhà trường và tra cứu bảng điểm cá nhân.
* Giảng viên: Là người đang giảng dạy tại trường Đại học Thăng Long, họ có quyền xem những thông báo của nhà trường và tra cứu bảng lương.
* Người dùng tự do: Là những người đang có ý định tìm hiểu về trường Đại học Thăng Long (bao gồm sinh viên tương lai và phụ huynh). Họ có thể xem được Chương trình đào tạo, tuyển sinh, sự kiên, học bổng, tra cứu điểm.
* Cựu sinh viên và doanh nghiệp: Là người đã tốt nghiệp Đại học Thăng Long, họ có quyền thêm bài viết.

### Các chức năng chính của hệ thống

* **UC#01 – Đăng nhập**: Hệ thống yêu cầu các tác nhân tham gia hệ thống đăng nhập để có thể thực hiện các chức năng hệ thống cung cấp.
* **UC#1.1 – Cập nhật thông tin cá nhân**: Hệ thống cho phép người dùng có thể chỉnh sửa thông tin cá nhân.
* **UC#1.2 – Xem thông tin cá nhân**: Hệ thống cho phép người dùng có thể xem thông tin cá nhân.
* **UC#1.3 – Quên mật khẩu**: Hệ thống cho phép người dùng có thể lấy lại mật khẩu trong trường hợp mất hoặc quên mật khẩu.
* **UC#1.4 – Đăng xuất**: Hệ thống cho phép người dùng có thể đăng xuất khỏi hệ thống.
* **UC#02 – Tạo tài khoản**: Hệ thống cho phép tác nhân Admin có quyền tạo tài khoản để cấp cho người dùng.
* **UC#03 – Quản lý danh mục**: Hệ thống cho phép tác nhân Admin có thể quản lý (thêm, sửa, xoá) các danh mục hiển thị của website.
* **UC#04 – Quản lý bài đăng tuyển dụng**: Hệ thống cho phép tác nhân Admin có thể quản lý(thêm, sửa, xoá) các bài viết của Cựu sinh viên và doanh nghiệp.
* **UC#05 – Quản lý sinh viên** : Hệ thống cho phép tác nhân Admin có thể quản lý(thêm, sửa, xoá) các thông tin của sinh viên.
* **UC#06 – Quản lý giảng viên**: Hệ thống cho phép tác nhân Admin có thể quản lý(thêm, sửa, xoá) các thông tin của giảng viên.
* **UC#07 – Quản lý văn bản**: Hệ thống cho phép tác nhân Admin có thể quản lý(thêm, xoá) các văn bản mà nhà trường đưa ra.
* **UC#08 – Quản lý phân quyền**: Hệ thống cho phép tác nhân Admin có thể quản lý (thêm, sửa, xoá) các quyền của hệ thống.
* **UC#09 – Quản lý điểm**: Hệ thống cho phép tác nhân Admin có thể quản lý(thêm, sửa, xoá) điểm của sinh viên.
* **UC#10 – Duyệt bài đăng tuyển dụng**: Hệ thống cho phép tác nhân Admin có thể duyệt bài viết của Cựu sinh viên và doanh nghiệp.
* **UC#11 – Thêm bài đăng tuyển dụng**: Hệ thống cho phép tác nhân Cựu sinh viên và doanh nghiệp có thể thêm bài viết mới.
* **UC#12 – Tra cứu điểm**: Hệ thống cho phép tác nhân Phụ huynh có thể tra cứu điểm của con cái mình.
* **UC#13 – Tra cứu lương**: Hệ thống cho phép tác nhân Giảng viên có thể tra cứu bảng lương của mình.
* **UC#14 – Xem thông báo của nhà trường**: Hệ thống cho phép tác nhân Sinh viên và Giảng viên có thể xem được những thông báo của nhà trường đưa ra.
* **UC#15 – Xem bảng điểm cá nhân**: Hệ thống cho phép tác nhân Sinh viên có thể xem được bảng điểm cá nhân.
* **UC#16 – Xem chương trình đào tạo**: Hệ thống cho phép tác nhân Sinh viên tương lai có thể xem được chương trình đào tạo của nhà trường.
* **UC#17 – Xem chương trình tuyển sinh**: Hệ thống cho phép tác nhân Sinh viên tương lai có thể xem được chương trình tuyển sinh của nhà trường.
* **UC#18 – Xem sự kiện**: Hệ thống cho phép tác nhân Sinh viên tương lai có thể xem được những sự kiện của nhà trường.
* **UC#19 – Xem học bổng**: Hệ thống cho phép tác nhân Sinh viên tương lai có thể xem được chương trình học bổng của nhà trường.

# CÔNG NGHỆ SỬ DỤNG VÀ KIẾN TRÚC TỔNG THỂ

## Công nghệ sử dụng

### SQL – Server

Microsoft SQL Server(MSSQL) là hệ thống quản lý cơ sở dữ liệu quan hệ được phát triển bởi Microsoft. MSSQL là một hệ quản trị cơ sở dữ liệu rất phổ biến trên thế giới, nó là nơi chứa dữ liệu cho nhiều phần mềm doanh nghiệp lớn, nhiều website cũng như hệ thống sản xuất. MSSQL rất dễ sử dụng. MSSQL đi kèm với các công cụ tuyệt vời mà sẽ giúp lập trình viên tiết kiệm rất nhiều thời gian trong các lĩnh việc sửa chữa và năng cấp. Với MSSQL, việc thiết lập gần như mọi thứ, từ cài đặt trên máy ảo đến việc viết và chỉnh sửa truy vấn là vô cùng dễ dàng với MSSQL. Đặc biệt là so với các hệ quản trị CSDL khác. Nếu có vấn đề trong bất kỳ giai đoạn phát triển nào, sẽ có rất nhiều hỗ trợ trực tuyến và tài liệu bổ sung ngoài hỗ trợ sản phẩm trực tiếp, trong khi các tùy chọn hỗ trợ cho các bằng các hệ quản trị CSDLkhông gần như mạnh mẽ.

### ORM - Entity Framework Core

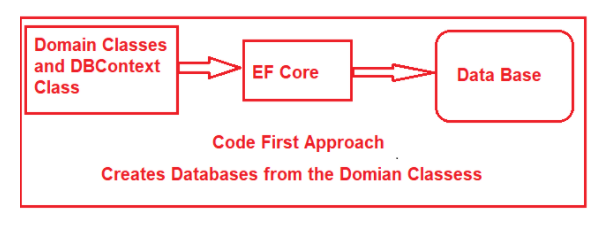
ORM (**O**bject **R**elational **M**apping), là một kỹ thuật/cơ chế lập trình thực hiện ánh xạ CSDL sang các đối tượng trong các ngôn ngữ lập trình hướng đối tượng như Java, C# …(các table tương ứng các class, mối ràng buộc giữa các table tương ứng quan hệ giữa các class has-one, has-many).

* Ưu điểm của ORM:
* Giúp lập trình viên tập trung vào lập trình hướng đối tượng.
* Tính độc lập – làm việc được với nhiều loại database, nhiều kiểu dữ liệu khác nhau. Dễ dàng thay đổi loại database hơn, các câu truy vấn không phụ thuộc vào loại database nào.
* Đơn giản, dễ sử dụng, cung cấp nhiều API truy vấn. Viết code dễ hiểu hơn, phù hợp với các case CRUD (Create, Read, Update, Delete).
* Nhược điểm:
* Khả năng truy vấn bị hạn chế, nhiều trường hợp ta vẫn phải dùng native SQL để truy vấn database.
* Khó tối ưu được SQL (do câu lệnh SQL được ORM tự động sinh ra).

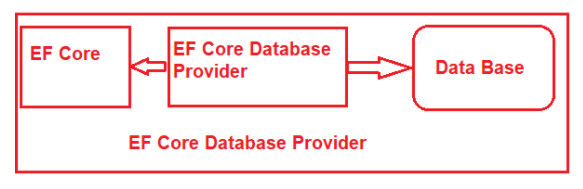
Entity Framework Core (EF Core) là một ORM Framework để truy cập dữ liệu trong .NET. Nó được phát hành cùng .Net Core và là phiên bản mở rộng, nhẹ, mã nguồn mở và đa nền tảng của công nghệ truy cập dữ liệu Entity Framework.

Entity Framework Core hỗ trợ 2 cách tiếp cận sau:

* EF Core Code First Approach:
* Trong hướng tiếp cận là Code First của EF Core, trước tiên chúng ta cần tạo các domain class như Student, Branch, Address,…và một class đặc biệt dẫn xuất từ class DbContext của Entity Framework.



* Theo hướng tiếp cận Code-First, EF Core tạo cơ sở dữ liệu và bảng bằng cách sử dụng migration dựa trên các conventions, configuration mặc định.
* Chúng ta có thể thay đổi các conventions mặc định để tạo các bảng quan hệ cho cơ sở dữ liệu. Cách tiếp cận này hữu ích trong thiết kế theo hường nghiệp vụ (Domain-Driven Design – DDD)
* EF Core Database First Approach:
* Nếu đã có một cơ sở dữ liệu hoàn chỉnh, thì chúng ta nên sử dụng EF Core Database First. EF Core sẽ tạo các class DbContext và domain class dựa trên lược đồ cơ sở dữ liệu đã có sẵn bằng EF Core Command.



### ASP.NET Core SignalR

ASP.NET Core SignalR là một thư viện mã nguồn mở giúp đơn giản hóa việc thêm chức năng web real-time trong phát triển ứng dụng. Chức năng web real-time có khả năng server đấy những nội dung tới client đã được kết nối một cách tức thì. Nó khác với giao thức HTTP thông thường: Server đợi những request từ client và trả về response tương ứng.

SignalR cung cấp một API đơn giản cho việc tạo server-to-client remote procedure call (RPC) để gọi những hàm javascript trong trình duyệt (và những nền tảng khác) từ code .Net của server-side. SignalR cũng bao gồm API cho việc quản lý kết nối (connect và disconnect events) và những kết nối nhóm.



Hình 2.2. Mô tả truyền nội dụng giữa client và server bằng SignalR

SignalR xử lý quản lý kết nối một cách tự động, và truyền đi thông điệp tới tất cả các client đã được kết nối một cách đồng loạt, giống như một chat room. Bạn cũng có thể gửi những thông điệp tới những client được xác định. Kết nối giữa client và server là liên tục, không giống như kết nối HTTP cổ điển.

SignalR sử dụng phương thức truyền tải WebSocket mới, và trở lại với phương thức cũ hơn nơi cần thiết. Trong khi có thể viết ứng dụng sử dụng WebSocket một các trực tiếp, khi sử dụng SignalR rất nhiều chức năng mở rộng cần triển khai đã được làm sẵn cho ứng dụng. Điều này có nghĩa là ta có thể viết ứng dụng với những ưu điểm của WebSocket mà không phải lo lắng về việc phân chia code cho những client cũ hơn.

### ClearScript

ClearScript là một thư viện cho phép thêm các tập lệnh vào ứng dụng .NET. Hiện tại nó hỗ trợ mã JavaScript và VBScript.

ClearScript hỗ trợ V8 – JavaScript Engine. Không giống như các công cụ Windows Script. V8 sẽ biên dịch và thực thi mã nguồn JavaScript, xử lý cấp phát bộ nhớ cho đối tượng, và loại bỏ đối tượng không còn cần đến. V8 cung cấp tất cả các kiểu dữ liệu, toán tử, đối tượng và hàm được chỉ định trong tiêu chuẩn ECMA.

ClearScript cho phép mã script có thể truy cập vào danh sách tính năng của .NET – constructors, methods, properties, events, indexers, extension methods, statics generics, enums,…Nó thực hiện điều này mà không yêu cầu phần mở rộng ngôn ngữ script dưới bất kì hình thức nào. ClearScript cũng cho phép máy chủ có thể gọi các hàm script và truy cập trực tiếp vào đối tượng script. Ngoài ra, ClearScript hỗ trợ gỡ lỗi trong qua trình thực thi script.

### ABP Framework

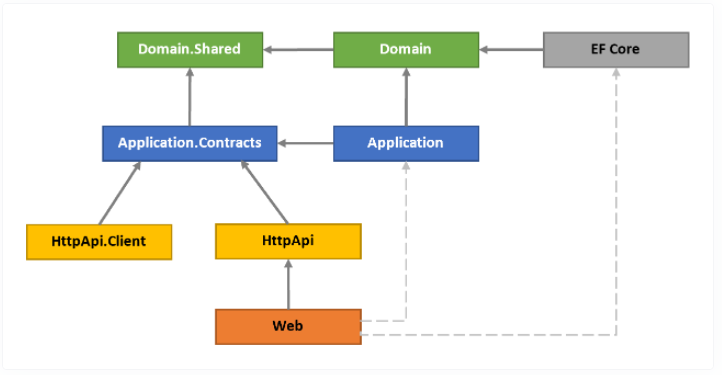
ABP Framework có tên đầy đủ là ASP.NET Boilerplate Framework. ABP là một ứng dụng framework mã nguồn mở tập trung vào phát triển ứng dụng web dựa trên nền tảng ASP.NET Core và cũng hỗ trợ phát triển các loại ứng dụng khác. ABP cung cấp một điểm bắt đầu cơ bản cho các ứng dụng web hiện đại sử dụng các thực tiễn tốt nhất và các công cụ phổ biến nhất.

ABP Framework được phát hành vào năm 2013, lúc đó chưa có .Net Core & ASP.NET Core và cũng chưa có Angular2+. Tất cả chúng đều được phát triển từ đầu. ASP.NET Core đã ra đời và giới thiệu nhiều giải pháp tích hợp sẵn (thư viện mở rộng) cho **dependency injection, logging, localization, configuration**,…Các nhà phát triển ABP đã sử dụng thư viện của bên thứ 3 điển hình có thể nói đến Datatable JS.

ABP cung cấp một kiến trúc phần mềm hoàn chỉnh, module và layer dựa trên **Domain Driven Design**:

* Domain Layer được chia thành 2 project:
* Domain là project thiết yếu trong Domain Layer. Chứa các entities, value object, domain service, repository interfaces,…
* Domain.Shared là project đơn giản chứa các const, enum và các types có thể dùng chung với tất cả các class trong các module.
* Application Layer được chia thành 2 project:
* Application.Contracts chứa các application service interface và DTOs sử dụng bởi interface đó.
* Application là project thiết yếu trong Application Layer, thực hiện triển khai các interface đã được xác định trong project Contracts.
* Presentation Layer:
* Web là một ứng dụng ASP.NET Core MVC/Razor Pages đơn giản. Tại đây ứng dụng sẽ được thực thi.
* Remote Service Layer:
* HttpApi thường chứa MVC Controllers và models có liên quan nếu có. Vì vậy có thể viết các HTTP API trong project này.
* HttpApi.Client là một project hữu ích khi có một ứng dụng C# cần sử dụng API. Sau khi ứng dụng khách tham chiếu đến project này, nó có thể trực tiếp sử dụng Application Service.
* Infrastructure Layer được chia thành 2 project:
* EntityFrameworkCore là project cần thiết để tích hợp EF Core. Tại đây chứa DbContext, database mappings, triển khai các repositories và các nội dung khác liên quan đến EF Core.
* EntityFrameworkCore.Migration là project đặc biệt để quản lý thay đổi cấu hình trong cơ sở dữ liệu.

**Sự phụ thuộc của các project trong solution:**



Hình 2.1. Sơ đồ mô tả kiến trúc module của ABP

* Domain.Shared là project mà tất cả các project khác trực tiếp hoặc gián tiếp phụ thuộc vào. Vì vậy, tất cả các data types trong này có thể sử dụng trong tất cả các project trong solution.
* Domain chỉ phụ thuộc duy nhất vào Domain.Shared bởi vì nó là một phần của Domain Layer.
* Application.Contracts phụ thuộc vào Domain.Shared. Tại đây, có thể sử dụng các types trong DTOs.
* Application phụ thuộc vào Application.Contracts. Tại project sẽ triển khai các application service interfaces và sử dụng các DTOs bên trong Contracts. Nó cũng phụ thuộc vào Domain, vì trong triển khai application service sử dụng các domain objects được định nghĩa bên trong project Domain.
* EntityFrameworkCore phụ thuộc vào Domain. Nó thực hiện ánh xạ các entities với các bảng trong database và triên khai repository interfaces đã xác định trong Domain.
* HttpApi phụ thuộc vào ApplicationContracts. Khi định nghĩa các controller bên trong HttpApi cần sử dụng application service interfaces
* HttpApi.Client phụ thuộc vào ApplicationContracts để có thể sử dụng application service như giải thích trước.
* Web phụ thuộc vào HttpApi vì nó sử dụng HTTP API được định nghĩa bên trong HttpApi và gián tiếp phụ thuộc vào Application.Contracts để có thể sử dụng application service trong Pages/Component.

## Kiến trúc tổng thể

### Domain Driven Design

Domain Driven Design (DDD) là một cách tiếp cận cho phát triển các phần mềm phức tạp, sự phức tạp ở đây là do nghiệp vụ của lĩnh vực kinh doanh (domain business). Cách tiếp cận của DDD là kết nối chặt chẽ giữa việc triển khai ứng dụng với sự phát triển của mô hình kinh doanh. Tiền đề tạo nên DDD:

* Đặt trọng tâm dự án vào nghiệp vụ chính (core domain) và các logic của nghiệp vụ (domain logic);
* Mô hình hóa là trọng tâm, là nền tảng cho các thiết kế phức tạp;
* Sự cộng tác đầy sáng tạo giữa nhóm nhà phát triển và các domain expert (chuyên gia nghiệp vụ) tạo nên tiếng nói chung để xác định và giải quyết hiệu quả các vấn đề.

DDD tập trung vào khái niệm domain (nghiệp vụ) và bóc tách bài toán dựa trên các nghiệp vụ đó. DDD giúp giải quyết vấn đề xây dựng các hệ thống phức tạp. Mô hình này nhà phát triển và chuyên gia nghiệp vụ phải hiểu chính xác các yêu cầu từ trước. Sau đó, họ xác định các hành vi, hiểu các quy tắc, áp dụng business login. Tiếp theo, các nhà phát triển sẽ triển khai chúng trong các lớp (Application Layer, Domain Layer, Infrastructure Layer,…).

Ưu điểm:

* Tạo ra ngôn ngữ chung giúp cho việc giao tiếp giữa nhóm phát triển với các chuyên gia nghiệp vụ trở nên dễ dàng và cũng như giữa các nhà phát triển với nhau.
* Hệ thống được xây dựng theo hướng nghiệp vụ, nó sẽ linh hoạt hơn để thay đổi vì các yêu cầu chức năng phụ hợp theocách tự nhiên.
* Các thành viên trong nhóm có thể độc lập phát triển cùng lúc. Các phần của hệ thống tương tác với nhau thông qua các định nghĩa và nguyên tắc được trình bày trong interface, abstract class, base class,…Các triển khai sẽ được hoàn thành trong các lớp còn lại.
* Nâng cấp mở rộng các chức năng một cách linh hoạt không ảnh hưởng đáng kểtới tổng thể hệ thống.
* DDD chia rõ ràng các chức năng giữa các layer mang lại khả năng viết code chuẩn rõ ràng và đáng tin cậy hơn. Ngoài ra, có thể dễ dàng tìm kiếm code, hạn chế trùng lặp và giảm chi phí bảo trì.

Nhược điểm:

* Yêu cầu chuyên môn sâu rộng về nghiệp vụ. Nhóm phát triển cần có ít nhất một chuyên gia nghiệp vụ. Họ sẽ xác định tất cả các quy trình, thủ tục và thuật ngữ của nghiệp vụ.
* Chi phí phát triển tương đối cao không phù hợp cho các dự án ngắn hạn hoặc dự án không có nghiệp phức tạp cao.

Kiến trúc layer theo hướng tiếp cận domain-driven design bao gồm 4 layer cơ bản:

A picture containing timeline

Description automatically generated

Hình 2.5. Kiến trúc layer theo hướng tiếp cận với domain-driven design

* **Presentation Layer**: Bao gồm giao diện người dùng, sử dụng tầng Application Layer để đáp ứng tương tác của người dùng. Ngoài ra chịu trách nhiệm truyền tải các nội dung từ client đến server và ngược lại.
* **Application Layer**: Nằm giữa Presentation và Domain Layer. Chịu trách nhiệm đối với các business object để thực hiện các tác vụ ứng dụng cụ thể. Triển khai các use-case logic của ứng dụng.
* **Domain Layer**: Bao gồm các business object và các quy tắc nghiệp vụ chính. Domain Layer còn là trung tâm của ứng dụng.
* **Infrastructure Layer**: Bao gồm kết nối với cơ sở dữ liệu thông qua ORM Framework – EF Core hoặc ADO.NET và triển khai các repository interface.

### Mô tả tổng quan

Sơ đồ dưới đây mô tả quá trình tương tác giữa hệ thống với các thành phần liên quan bao gồm: User, Browser, Web Server, Database, Complier Application, Application

Diagram

Description automatically generated

Hình 2.3. Sơ đồ mô tả sự tương tác của hệ thống với các thành phần

1. Người dùng tương tác với hệ thống qua web page chạy trên trình duyệt.
2. Trình duyệt sẽ gửi request đến Web Server, sau đó nhận response rồi trả về trình duyệt.
3. Web Server truy vấn cơ sở dữ liệu qua ORM Framework.
4. Ứng dụng biên dịch thực hiện các tương tác như nhận script, trả về kết quả thực thi script và gửi thông tin của máy client tới Web Server thông qua kết nối WebSocket sử dụng thư viện SignalR.
5. Các ứng dụng gửi các thông tin cập nhật đến Web Server thông qua API.
6. Ứng dụng biên dịch có thể thực hiện xử lý bật tắt ứng dụng thông qua process của windows.

Diagram

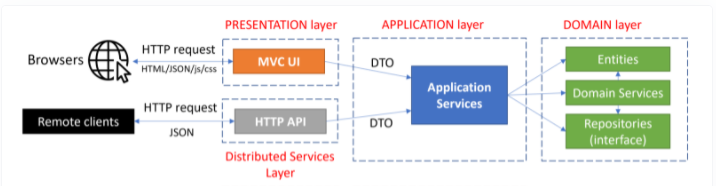
Description automatically generated

Hình 2.4. Sơ đồ trạng thái sự tương tác của hệ thống

### Kiến trúc module của hệ thống theo DDD

Hệ thống được thiết kế theo hướng tiếp cận Domain Driven Design với những yều cầu rõ ràng từ bài toán và các chức năng có thể triển khai độc lập với nhau. Với ưu điểm của DDD hệ thống có thể mở rộng các chức năng và nâng cấp bảo trì trong tương lai một cách dễ dàng.

Hình dưới đây mô tả luồng thực thi sử kiện cho một ứng dụng web đã được phát triển dựa trên mẫu DDD.



Hình 2.6. Mô tả quy trình thực thi sự kiện của ứng dụng web theo mẫu DDD

* Request thường được bắt đầu với một tương tác của người dùng trên giao diện sẽ tạo ra một HTTP Request đến server.
* Controller sẽ xử lý request, sau đó inject application service interface và gọi các phương thức của service tướng ứng với request và trả về DTO.
* Application Service sử dụng các Domain Objects (Entities, Repository interfaces,…) để triển khai use case.

Các module chính được sử dụng trong các mỗi layer:

* **Presentation Layer**: Web, HttpApi;
* **Application Layer**: Application.Contracts, Application;
* **Domain Layer**: Domain.Shared, Domain;
* **Infrastructure Layer**: EntityFrameworkCore, BackgroundWoker.

Diagram

Description automatically generated

Hình 2.7. Sơ đồ kiến trúc module của hệ thống

* **Cấu trúc solution:**

**A screen shot of a computer

Description automatically generated with low confidence**

Hình 2.8. Cấu trúc solution

* **Domain**.**Shared:** Module Domain.Shared chứa các hàm xử lý tính toán, định nghĩa object enum và xử lý đa ngôn ngữ được sử dụng chung cho các module được liên kết tới module này.

Text

Description automatically generated

Hình 2.10. Module Domain.Shared trong solution

* **Domain**:

Text

Description automatically generated

Hình 2.9. Module Domain trong solution

* Module Domain phụ thuộc vào module Domain.Shared.
* Entity là một đối tượng có các thuộc tính riêng (trạng thái, dữ liệu) và các phương thức triển khai business logic được thực thi trên các thuộc tính này. Một entity sẽ được ánh xạ tới một bảng trong cơ sở dữ liệu.
* Repository interface được sử dụng bởi Application Layer để thực hiện các thao tác với cơ sở dữ liệu. Nó ẩn đi nội dung code xử lý nghiệp vụ phức tạp, mỗi repository được sử dụng cho mỗi entity.
* **Application**.**Contracts:**

Text

Description automatically generated

Hình 2.11. Module Application.Contracts trong solution

* Module Application.Contracts phụ thuộc vào module Domain.Shared và chứa các application service interface và entity DTO.
* Entity DTO là một đối tượng đơn giản không chứa bất kì nghiệp vụ logic nào, nó chỉ sử dụng để chuyển dữ liệu giữa Application Layer và Presentation Layer. Mục đích để giảm bớt lượng thông tin không cần thiết phải chuyển đi và cũng tăng độ bảo mật.
* Khai báo các permission cần có trong hệ thống.
* **Application:**

Text

Description automatically generated

Hình 2.12. Module Application trong solution

* Module Application phụ thuộc vào module Application.Contracts và Domain.
* Thực hiện triển khai các application service interface được xác định trong module Contracts. Một application service thường thực hiện lấy data từ repository và trả về thông qua DTO. Nó được sử dụng bởi module Web và HttpApi.
* **EntityFrameworkCore:**

Text

Description automatically generated

Hình 2.13. Module EntityFrameworkCore trong solution

* Module EntityFrameworkCore phụ thuộc vào module Domain và truy vấn đến cơ sở dữ liệu.
* Thực hiện cấu hình DbContext, các quan hệ giữa các entity và kết nối với cơ sở dữ liệu.
* Thực hiện triển khai các repository theo interface của repository đã được xác định tại module Domain.
* **BackgroundWoker:**

**Text

Description automatically generated**

Hình 2.14. Module BackgroundWokers trong solution

* Module BackgroundWoker phụ thuộc vào module Domain
* Thực hiện gọi một xử lý định kì theo thời gian cấu hình sẵn. Cụ thể trong đối với hệ thống này là xử lý gửi email thông báo.
* **HttpApi:**

**Text

Description automatically generated**

Hình 2.15. Module HttpApi trong solution

* Module HttpApi phụ thuộc vào module Application.Contracts
* Thực hiện định nghĩa các controller tương ứng đối với mỗi interface của application service.
* **Web:**

**Text

Description automatically generated**

Hình 2.16. Module Web trong solution

* Module Web phụ thuộc vào module HttpApi, có thể gọi các HTTP API tại chính module Web. Ngoài ra, nó gián tiếp phụ thuộc vào module Application.Contracts để sử dụng application service trong các Pages.
* Trong module chứa các xử lý PageModel, Templates và các file thư viện javascript, css.

# ĐẶC TẢ CHỨC NĂNG CHÍNH CỦA HỆ THỐNG

## Quản lý máy client

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#01** | | **Quản lý máy client** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái cho phép người dùng có thể thực hiện các thao tác quản lý máy client bao gồm:   * Tìm kiếm trên danh sách máy client: Người dùng có thể nhập từ khóa để tìm kiếm; * Xóa một bản ghi máy client: Người dùng có thể xóa một bản ghi máy client; * Chọn số lượng bản ghi hiển thị và phân trang. | |
| **Tác nhân** | | Người dùng | |
| **Tiền điều kiện** | | Sau khi quản trị hệ thống đăng nhập và chọn chức năng hiển thị danh sách máy client. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị đầy đủ danh sách máy client lên màn hình. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi đăng nhập vào hệ thống kiểm soát trạng thái, người dùng chọn chức năng “Quản lý máy client” trên thanh menu. Tại giao diện quản lý máy client như sau:   * Danh sách máy client: * Danh sách máy client được hiển thị 15 bản ghi trên một trang và có phân trang từ 1 đến hết; * Có ô nhập điều kiện tìm kiếm trên danh sách. * Cho phép sắp xếp trên một số cột trên danh sách. * Nút chức năng: * Xóa   **Luồng 1 – Thực hiện tìm kiếm theo điều kiện nhập vào:**   * Tại màn hình quản lý máy client, người dùng nhập từ khóa vào ô tìm kiếm. Khi ngừng nhập hệ thống hiển thị kết quả tìm kiếm trên danh sách.   **Luồng 2 – Thực hiện xóa một bản ghi máy client:**   * Tại màn hình quản lý máy client, người dùng sẽ click vào nút “Xóa” trên một bản ghi máy client. Hệ thống sẽ hiển thị một thông báo xác nhận xóa; * Nhấn “Đồng ý” để xóa bản ghi vừa chọn; * Nhần “Đóng” để trở lại. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| N/A | | | |
| **Giao diện minh họa** | | | |
| Graphical user interface, text, application  Description automatically generated  Hình 3.1. Giao diện màn hình danh sách máy client | | | |

## Quản lý script

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#02** | | **Quản lý script** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái cho phép người dùng có thể thực hiện các thao tác quản lý script bao gồm:   * Tìm kiếm trên danh sách script: Người dùng có thể nhập từ khóa để tìm kiếm; * Thêm mới script: Người dùng có thể thêm mới một bản ghi script; * Cập nhật script: Người dùng có thể câp nhật một bản ghi script; * Xóa script: Người dùng có thể xóa một bản ghi script; * Chọn số lượng bản ghi hiển thị và phân trang. | |
| **Tác nhân** | | Người dùng. | |
| **Tiền điều kiện** | | Sau khi quản trị hệ thống đăng nhập và chọn chức năng quản lý script | |
| **Hậu điều kiện** | **Thành công** | * Hiển thị đầy đủ bản ghi trên danh sách; * Cập nhật thành công cho quản lý script và lưu dữ liệu vào hệ thống. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi đăng nhập vào hệ thống kiểm soát trạng thái, người dùng chọn chức năng “Quản lý script” trên thanh menu. Tại giao diện quản lý script như sau:   * Danh sách script: * Danh sách script được hiển thị 15 bản ghi trên một trang và có phân trang từ 1 đến hết; * Có ô nhập điều kiện tìm kiếm trên danh sách; * Cho phép sắp xếp trên một số cột trên danh sách. * Nút chức năng: * Thêm mới; * Cập nhật; * Xóa.   **Luồng 1 – Thực hiện thêm mới script:**   * Tại màn hình quản lý script, người dùng click vào nút “Thêm mới”, hệ thống sẽ hiển thị màn hình thêm mới script. * Trên màn hình thêm mới script người dùng cần nhập đúng thông tin theo yêu cầu: * Tên script: Định dạng text; * Nội dung script: Định dạng text; * Chọn script phụ thuộc (nếu có): Danh sách script đã có được hiển thị dưới dạng checkbox. * Nhấn “Lưu” để hoàn tất thêm mới; * Nhấn “Hủy bỏ”   **Luồng 2 – Thực hiện cập nhật script:**   * Tại màn hình quản lý script, người dùng click nút “Cập nhật”, hệ thống sẽ hiển thị màn hình cập nhật script. * Trên màn hình cập nhật script người dùng cần nhập đúng thông tin theo yêu cầu (quy tắc nhập giống màn hình thêm mới). * Nhấn “Lưu” để hoàn tất cập nhật. * Nhấn “Hủy bỏ”.   **Luồng 3 – Thực hiện xóa một bản ghi script:**   * Tại màn hình quản lý script, người dùng sẽ click vào nút “Xóa” trên một bản ghi script. Hệ thống sẽ hiển thị một thông báo xác nhận xóa; * Nhấn “Đồng ý” để xóa bản ghi vừa chọn; * Nhần “Đóng” để trở lại.   **Luồng 4 – Thực hiện tìm kiếm theo điều kiện nhập vào:**   * Tại màn hình quản lý script, người dùng nhập từ khóa vào ô tìm kiếm. Khi ngừng nhập hệ thống hiển thị kết quả tìm kiếm trên danh sách. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| Hiển thị thông báo trong trường hợp thêm mới và cập nhật không hợp lệ (không tuân thủ đúng quy tắc đã đặt ra tại luồng sự kiện chính).  Hệ thống hiển thị thông báo không hợp lệ trong trường hợp vi phạm điều kiện xóa script. | | | |
| **Giao diện minh họa** | | | |
| Hình 3.2. Giao diện màn hình danh sách script    Hình 3.3. Giao diện màn hình thêm mới script    Hình 3.4. Giao diện màn hình cập nhật script    Hình 3.5. Giao diện màn hình xác nhận xóa | | | |

## Thực thi script

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#03** | | **Thực thi script** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái cho phép người dùng thực hiện thao tác gửi script đến ứng dụng biên dịch script.  Ứng dụng biên dịch script sẽ thực thi script và sau đó trả về kết quả. | |
| **Tác nhân** | | * Người dùng. * Ứng dụng biên dịch script. | |
| **Tiền điều kiện** | | Sau khi quản trị hệ thống đăng nhập và chọn chức năng thực thi script. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị kết quả thực thi script trên màn hình. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi đăng nhập vào hệ thống kiểm soát trạng thái, người dùng chọn chức năng “Thực thi script” trên thanh menu. Tại giao diện quản lý script như sau:   * Danh sách script: * Hiển thị dưới dạng checkbox; * Danh sách máy client: * Hiển thị dưới dạng checkbox. * Tab hiển thị kết quả tương ứng với máy client được chọn để thực thi script. * Nội dung script: Định dạng text. * Các nút chức năng: * Nút icon play; * Nút icon biểu tượng mắt; * Nút icon clean.   **Luồng 1 – Gửi script đến Ứng dụng biên dịch script:**   * Trên màn hình thực thi script người dùng cần nhập đúng thông tin yêu cầu: * Chọn máy client để thực thi script (tối đa 5 máy client); * Nội dung script: Định dạng text; * Chọn các script đã được thêm trước đó nếu sử dụng đến; * Nhấn nút icon play để hoàn tất. * Click trực tiếp tại nút icon play của một script đã có được thêm trước đó tại danh sách script. * Sau khi Ứng dụng biên dịch script thực thi xong kết quả sẽ được hiển thị trên tab kết quả của từng máy client đã chọn.   **Luồng 2 – Thực hiện thực thi script:**   * Ứng dụng biên dịch script sẽ load các script đã có sẵn mà được sử dụng trong script cần thực thi. * Trong quá trình thực thi nếu gặp lỗi thì phải gửi lỗi về hệ thống kiếm soát trạng thái. * Sau khi thực thi thành công kết quả sẽ được gửi về hệ thống kiểm soát trạng thái. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| Ứng dụng biên dịch script bị mất kết nối đến hệ thống kiểm soát trạng thái. | | | |
| **Giao diện minh họa** | | | |
| Hình 3.6. Giao diện màn hình thực thi script | | | |

## Cập nhật máy client

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#04** | | **Cập nhật máy client** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái cho phép ứng dụng biên dịch script thực hiện thao tác cập nhật thông tin của máy client đến hệ thống bao gồm: IP LAN, IP Public, phần trăm CPU, phần trăm Memory. | |
| **Tác nhân** | | * Hệ thống kiểm soát trạng thái. * Ứng dụng biên dịch script. | |
| **Tiền điều kiện** | | Sau Ứng dụng biên dịch script được kết nối đến hệ thống kiểm soát trạng thái. | |
| **Hậu điều kiện** | **Thành công** | Dữ liệu máy client được cập nhật thành công vào hệ thống kiểm soát trạng thái. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi ứng dụng biên dịch script được kết nối đến hệ thống kiểm soát trạng thái sẽ có 2 luồng sự kiện sau:  **Luồng 1 – Gửi thông tin máy client đến hệ thống kiểm soát trạng thái:**   * Ứng dụng biên dịch script thực hiện lấy các thông tin báo gồm: * IP LAN; * IP Public được lấy từ request; * Phần trăm CPU; * Phần trăm Memory. * Thực hiện xử lý định kì gửi các thông tin trên đến hệ thống kiểm soát trạng thái.   Hệ thống kiểm soát trạng thái thực hiện kiểm tra thông tin của máy client có tồn tại trong hệ thống không. Nếu đã tồn tại thực hiện cập nhật máy client, ngược lại thực hiện thêm mới máy client vào hệ thống.  **Luồng 2 – Thực hiện thêm mới máy client:**   * Kiểm tra thông tin của máy client không tồn tại trong hệ thống kiểm soát trạng thái, hệ thống thực hiện thêm mới máy client với các trường thông tin sau: * ID của tài khoản đăng nhập trên máy client; * IP Lan; * IP Public; * Trạng thái (Online/Offline); * Thời gian thêm mới;   **Luồng 3 – Thực hiện cập nhật máy client:**   * Kiểm tra thông tin của máy client đã tồn tại trong hệ thống kiểm soát trạng thái, hệ thống thực hiện cập nhật máy client với các trường thông tin sau: * IP Lan; * IP Public; * Trạng thái (Online/Offline); * Thời gian cập nhật. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| Ứng dụng biên dịch script bị mất kết nối đến hệ thống kiểm soát trạng thái. | | | |

## Quản lý ứng dụng

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#05** | | **Quản lý ứng dụng** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái cho phép người dùng có thể thực hiện các thao tác quản lý ứng dụng bao gồm:   * Tìm kiếm trên danh ứng dụng: Người dùng có thể nhập từ khóa để tìm kiếm; * Bật ứng dụng: Người dùng bật ứng dụng đã cài trên máy client; * Tắt ứng dụng: Người dùng tắt ứng dụng đã cài trên máy client; * Xóa ứng dụng: Người dùng có thể xóa một bản ghi ứng dụng; * Chọn số lượng bản ghi hiển thị và phân trang. | |
| **Tác nhân** | | Người dùng | |
| **Tiền điều kiện** | | Sau khi quản trị hệ thống đăng nhập và chọn chức năng quản lý ứng dụng. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị đầy đủ danh sách ứng dụng trên màn hình. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi đăng nhập vào hệ thống kiểm soát trạng thái, người dùng chọn chức năng “Quản lý ứng dụng” trên thanh menu. Tại giao diện quản lý ứng dụng hiển thị như sau:   * Danh sách ứng dụng: * Danh sách máy client được hiển thị 15 bản ghi trên một trang và có phân trang từ 1 đến hết; * Có ô nhập điều kiện tìm kiếm trên danh sách. * Cho phép sắp xếp trên một số cột trên danh sách. * Nút chức năng: * Bật (đối với ứng dụng đang hoạt động); * Tắt (đối với ứng dụng ngừng hoạt động); * Xóa.   **Luồng 1 – Thực hiện bật ứng dụng:**   * Tại màn hình quản lý ứng dụng, người dùng click vào nút “Bật” tại bản ghi ứng dụng ngừng hoạt động. * Hệ thống kiểm soát trạng thái sẽ gửi thông tin của ứng đến ứng dụng biên dịch script. * Ứng dụng biên dịch script sẽ bật ứng dụng đó.   **Luồng 2 – Thực hiện tắt ứng dụng:**   * Tại màn hình quản lý ứng dụng, người dùng click vào nút “Tắt” tại bản ghi ứng dụng đang hoạt động. * Hệ thống kiểm soát trạng thái sẽ gửi thông tin của ứng đến ứng dụng biên dịch script. * Ứng dụng biên dịch script sẽ tắt ứng dụng đó.   **Luồng 3 – Thực hiện xóa một bản ghi script:**   * Tại màn hình quản lý ứng dụng, người dùng sẽ click vào nút “Xóa” trên một bản ghi ứng dụng. Hệ thống sẽ hiển thị một thông báo xác nhận xóa; * Nhấn “Đồng ý” để xóa bản ghi vừa chọn; * Nhấn “Hủy bỏ” để trở lại. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| Hệ thống hiển thị thông báo không hợp lệ trong trường hợp vi phạm điều kiện xóa script.  Ứng dụng biên dịch script bị mất kết nối đến hệ thống kiểm soát trạng thái. | | | |
| **Giao diện minh họa** | | | |
| ***Graphical user interface, text, application  Description automatically generated***  Hình 3.7.Giao diện màn hình danh sách ứng dụng | | | |

## Cập nhật ứng dụng

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#06** | | **Cập nhật ứng dụng** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái cho phép ứng dụng được cài đặt và chạy trên máy client mà hệ thống quản lý. Ứng dụng sẽ định kì gửi yêu cầu cập nhật thông tin đến hệ thống. | |
| **Tác nhân** | | * Ứng dụng. * Hệ thống kiểm soát trạng thái. | |
| **Tiền điều kiện** | | Ứng dụng được cài đặt và chạy trên máy client mà hệ thống kiểm soát trạng thái quản lý. | |
| **Hậu điều kiện** | **Thành công** | Cập nhật thành công thông tin của ứng dụng vào hệ thống kiểm soát trạng thái. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Khi hệ thống kiểm soát trạng thái nhận được thông tin từ ứng dụng, hệ thống sẽ kiểm tra thông tin có tồn tại trong hệ thống không. Nếu đã tồn tại thì thực hiện cập nhật ứng dụng, ngược lại thực hiện thêm mới ứng dụng vào hệ thống.  **Luồng 1 – Thực hiện thêm mới:**   * Kiểm tra thông tin của ứng dụng không tồn tại trong hệ thống kiểm soát trạng thái, hệ thống sẽ thực hiện thêm mới ứng dụng với các trường thông tin sau: * Tên ứng dụng; * ID của ứng dụng; * Version; * Thời gian thêm mới; * ID tiến trình ứng dụng đang chạy; * Đường dẫn file .exe của ứng dụng. * ID của máy client (được xác định bằng IP Lan và IP Public mà ứng dụng gửi đến hệ thống);   **Luồng 2 – Cập nhật ứng dụng:**   * Kiểm tra thông tin của ứng dụng đã có trong hệ thống kiểm soát trạng thái, hệ thống thực hiện cập nhật ứng dụng với các trường thông tin sau: * Tên ứng dụng; * Version; * Thời gian thêm mới; * ID tiến trình ứng dụng đang chạy; * Đường dẫn file .exe của ứng dụng. * ID của máy client (được xác định bằng IP Lan và IP Public mà ứng dụng gửi đến hệ thống); | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| N/A | | | |

## Quản lý nhóm quyền

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#07** | | **Quản lý nhóm quyền** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái cho phép người dùng có thể thực hiện các thao tác quản lý nhóm quyền bao gồm:   * Thêm mới nhóm quyền: Cho phép người dùng thêm mới một bản ghi nhóm quyền. * Cập nhật nhóm quyền: Cho phép người dùng cập nhật một bản ghi nhóm quyền. * Chỉ định các quyền có trong nhóm quyền đã thêm mới; * Tìm kiếm nhóm quyền theo từ khóa: Cho phép người dùng nhập từ khóa để tìm kiếm. * Chọn số lượng bản ghi hiển thị và phân trang. | |
| **Tác nhân** | | Người dùng. | |
| **Tiền điều kiện** | | Sau khi quản trị hệ thống đăng nhập và chọn chức năng quản lý nhóm quyền. | |
| **Hậu điều kiện** | **Thành công** | Cập nhật thành công cho quản lý nhóm quyền và lưu dữ liệu vào hệ thống. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi đăng nhập vào hệ thống kiểm soát trạng thái, người dùng chọn chức năng “Quản lý nhóm quyền” trên thanh menu. Tại giao diện quản lý nhóm quyền hiển thị như sau:   * Danh sách nhóm quyền: * Danh sách máy client được hiển thị 15 bản ghi trên một trang và có phân trang từ 1 đến hết; * Nút chức năng: * Thêm mới; * Quyền * Cập nhật.   **Luồng 1 – Thực hiện thêm mới nhóm quyền:**   * Tại màn hình quản lý nhóm quyền, người dùng click vào nút “Thêm mới”, hệ thống sẽ hiện thị màn hình thêm mới. * Trên màn hình thêm mới nhóm quyền người dùng cần nhập đúng thông tin theo yêu cầu: * Tên nhóm quyền; * Chọn để “Mặc định”, “Công khai”; * Nhấn “Lưu” để hoàn tất thêm mới; * Nhấn “Hủy bỏ”;   **Luồng 2 – Thực hiện cập nhật nhóm quyền:**   * Tại màn hình quản lý nhóm quyền, người dùng click nút “Cập nhật”, hệ thống sẽ hiển thị màn hình cập nhật nhóm quyền. * Trên màn hình cập nhật nhóm quyền người dùng cần nhập đúng thông tin theo yêu cầu (quy tắc nhập giống màn hình thêm mới). * Nhấn “Lưu” để hoàn tất cập nhật. * Nhấn “Hủy bỏ”.   **Luồng 3 – Thực hiện chỉ định các quyền thuộc nhóm quyền đã thêm mới:**   * Tại màn hình quản lý nhóm quyền, người dùng sẽ click vào nút “Quyền” hệ thống sẽ hiện thị màn hình danh sách các quyền để người dùng chọn; * Nhấn “Lưu” hoàn tất; * Nhấn “Hủy bỏ” để trở lại. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| Hiển thị thông báo trong trường hợp thêm mới và cập nhật nhóm quyền không hợp lệ (không tuân thủ đúng quy tắc đã đặt ra tại luồng sự kiện chính). | | | |
| **Giao diện minh họa** | | | |
| Hình 3.8. Giao diện màn hình danh sách nhóm quyền    Hình 3.9. Giao diện màn hình cập nhật nhóm quyền    Hình 3.10. Giao diện màn hình chọn quyền cho nhóm quyền | | | |

## Quản lý người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#08** | | **Quản lý người dùng** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái cho phép người dùng có thể thực hiện các thao tác quản lý người dùng bao gồm:   * Thêm mới người dùng: Cho phép người dùng thêm mới một bản ghi người dùng. * Cập nhật người dùng: Cho phép người dùng cập nhật một bản ghi người dùng. * Xóa người dùng: Cho phép người dùng xóa một bản ghi người dùng. * Tìm kiếm người dùng theo từ khóa: Cho phép người dùng nhập từ khóa để tìm kiếm. * Chọn số lượng bản ghi hiển thị và phân trang. | |
| **Tác nhân** | | Người dùng. | |
| **Tiền điều kiện** | | Sau khi quản trị hệ thống đăng nhập và chọn chức năng quản lý người dùng. | |
| **Hậu điều kiện** | **Thành công** | Cập nhật thành công cho quản lý người dùng và lưu dữ liệu vào hệ thống. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi đăng nhập vào hệ thống kiểm soát trạng thái, người dùng chọn chức năng “Quản lý người dùng” trên thanh menu. Tại giao diện quản lý người dùng hiển thị như sau:   * Danh sách người dùng: * Danh sách máy client được hiển thị 15 bản ghi trên một trang và có phân trang từ 1 đến hết; * Có ô nhập điều kiện tìm kiếm trên danh sách. * Nút chức năng: * Thêm mới; * Cập nhật. * Xóa * Quyền   **Luồng 1 – Thực hiện thêm mới người dùng:**   * Tại màn hình quản lý người dùng, người dùng click vào nút “Thêm mới”, hệ thống sẽ hiện thị màn hình thêm mới. * Trên màn hình thêm mới người dùng người dùng cần nhập đúng thông tin theo yêu cầu: * Tên đăng nhập; * Tên người dùng; * Họ; * Mật khẩu; * Địa chỉ email; * Số điện thoại; * Nhấn “Lưu” để hoàn tất thêm mới; * Nhấn “Hủy bỏ”;   **Luồng 2 – Thực hiện cập nhật người dùng:**   * Tại màn hình quản lý người dùng, người dùng click nút “Cập nhật”, hệ thống sẽ hiển thị màn hình cập nhật người dùng. * Trên màn hình cập nhật người dùng người dùng cần nhập đúng thông tin theo yêu cầu (quy tắc nhập giống màn hình thêm mới). * Nhấn “Lưu” để hoàn tất cập nhật. * Nhấn “Hủy bỏ”.   **Luồng 3 – Thực hiện xóa người dùng:**   * Tại màn hình quản lý người dùng, người dùng sẽ click vào nút “Xóa” trên một bản ghi người dùng. Hệ thống sẽ hiển thị một thông báo xác nhận xóa; * Nhấn “Đồng ý” để xóa bản ghi vừa chọn; * Nhần “Đóng” để trở lại.   **Luồng 4 – Thực hiện tìm kiếm theo điều kiện nhập vào:**   * Tại màn hình quản lý người dùng, người dùng nhập từ khóa vào ô tìm kiếm. Khi ngừng nhập hệ thống hiển thị kết quả tìm kiếm trên danh sách.   **Luồng 5 – Thực hiện chỉ định các quyền ngoài nhóm quyền đã gắn cho người dùng:**   * Tại màn hình quản lý người dùng, người dùng sẽ click vào nút “Quyền” hệ thống sẽ hiện thị màn hình danh sách các quyền để người dùng chọn; * Nhấn “Lưu” hoàn tất; * Nhấn “Hủy bỏ” để trở lại. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| Hiển thị thông báo trong trường hợp thêm mới và cập nhật người dùng không hợp lệ (không tuân thủ đúng quy tắc đã đặt ra tại luồng sự kiện chính).  Hệ thống hiển thị thông báo không hợp lệ trong trường hợp vi phạm điều kiện xóa người dùng. | | | |
| **Giao diện minh họa** | | | |
| Hình 3.11. Giao diện màn hình danh sách người dùng    Hình 3.12. Giao diện màn hình thêm mới người dùng    Hình 3.13. Giao diện màn hình cập nhật người dùng    Hình 3.14. Giao diện màn hình chọn quyền cho người dùng | | | |

## Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#09** | | **Đăng nhập** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Trước khi thực hiện các thao tác bên trong hệ thống kiểm soát trạng thái và ứng dụng biên dịch script kết nối với hệ thống kiểm soát trạng thái người dùng cần đăng nhập vào cả hai hệ thống. | |
| **Tác nhân** | | Người dùng. | |
| **Tiền điều kiện** | | Cần có tài khoản đăng nhập. | |
| **Hậu điều kiện** | **Thành công** | Hệ thống điều hướng đến màn hình trang chủ. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| * Tại màn hình đăng nhập của cả hai hệ thống, người dùng nhập đầy đủ thông tin đăng nhập bao gồm: Tài khoản, mật khẩu. * Hệ thống kiểm tra thông tin đăng nhập, nếu thành công sẽ chuyển hướng đến trang chủ. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| Hiển thị thông báo đăng nhập không thành công trong trường hợp lỗi thông tin đăng nhập. | | | |
| **Giao diện minh họa** | | | |
| Hình 3.15. Giao diện màn hình đăng nhập hệ thống kiểm soát trạng thái    Hình 3.16. Giao diện màn hình đăng nhập Ứng dụng biên dịch script | | | |

## Đăng xuất

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#10** | | **Đăng xuất** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Chức năng cho phép người dùng đăng xuất khỏi hệ thống. | |
| **Tác nhân** | | Người dùng. | |
| **Tiền điều kiện** | | Tài khoản người dùng đang đăng nhập. | |
| **Hậu điều kiện** | **Thành công** | Hiển thị giao diện đăng nhập. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| * Người dùng click vào nút “Đăng xuất”, hệ thống sẽ thực hiện đăng xuất tài khoản và hiển thị màn hình đăng nhập. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| N/A | | | |

## Kết nối với hệ thống kiểm soát trạng thái

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#11** | | **Kết nối với hệ thống kiểm soát trạng thái** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiếm soát trạng thái cho phép ứng dụng biên dịch script giữ kết nối liên tục với hệ thống. | |
| **Tác nhân** | | Người dùng. | |
| **Tiền điều kiện** | | Quản trị hệ thống đăng nhập thành công trên ứng dụng biên dịch script bằng tài khoản mà hệ thống kiểm soát trạng thái cấp. | |
| **Hậu điều kiện** | **Thành công** | Tạo kết nối thành công giữa hai hệ thống. | |
| **Lỗi** | Lỗi sẽ được ghi log và show lên màn hình Ứng dụng biên dịch script. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Hệ thống kiểm soát trạng thái sẽ cấp cho mỗi ứng dụng biên dịch script được cài trên các máy client một tài khoản khác nhau. Tài khoản chỉ được phép thực hiện cập nhật thông tin của máy client và thực thi script.  Sau khi người dùng sử dụng tài khoản được cấp đăng nhập thành công thì ứng dụng biên dịch script hiển thị như sau:   * Nội dung ghi log. * Nút chức năng: * Kết nối; * Ngắt kết nối; * Mở file log.   **Luồng 1 – Thực hiện kết nối:**   * Khi người dùng đăng nhập thành công hệ thống tự kết nối đến hệ thống kiểm soát trạng thái và hiển thị trên màn hình ứng dụng nội dung “Đã kết nối thành công”. * Nếu bị ngắt kết nối mà không cần đăng nhập lại người dùng có thể click vào nút “Kết nối” trên thanh menu. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| Đăng nhập không thành công do hệ thống kiểm soát trạng thái chưa hoạt động. | | | |
| **Giao diện minh họa** | | | |
| Hình 3.17. Giao diện màn hình thông báo đã kết nối | | | |

## Ngắt kết nối với hệ thống kiểm soát trạng thái

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#12** | | **Ngắt kết nối với hệ thống kiểm soát trạng thái** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Ứng dụng biên dịch script cho phép người dùng thực hiện thao tác ngắt kết nối với hệ thống kiểm soát trạng thái. | |
| **Tác nhân** | | Người dùng. | |
| **Tiền điều kiện** | | Quản trị hệ thống đăng nhập thành công trên ứng dụng biên dịch script bằng tài khoản mà hệ thống kiểm soát trạng thái cấp. | |
| **Hậu điều kiện** | **Thành công** | Tạo kết nối thành công giữa hai hệ thống. | |
| **Lỗi** | Lỗi sẽ được ghi log và show lên màn hình ứng dụng biên dịch script. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi người dùng sử dụng tài khoản được cấp đăng nhập thành công thì ứng dụng biên dịch script. Tại màn hình chính của ứng dụng người dùng thực hiện thao tác sau:  **Luồng 1 – Thực hiện ngắt kết nối:**   * Quản trị hệ thống click vào nút “Ngắt kết nối” trên thanh menu để thực hiện ngắt kết nối giữa hai hệ thống. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| * Kết nối đã được ngắt. | | | |
| **Giao diện minh họa** | | | |
| Hình 3.18. Giao diện màn hình thông báo ngắt kết nối | | | |

## Hiển thị nội dung file log

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#13** | | **Hiển thị nội dung file log** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Ứng dụng biên dịch script cho phép người dùng thực hiện thao tác mởi file ngay tại màn hình chính của ứng dụng. | |
| **Tác nhân** | | Người dùng. | |
| **Tiền điều kiện** | | Quản trị hệ thống đăng nhập thành công trên ứng dụng biên dịch script bằng tài khoản mà hệ thống kiểm soát trạng thái cấp. | |
| **Hậu điều kiện** | **Thành công** | Mở thành công file log. | |
| **Lỗi** | Lỗi sẽ được ghi log và show lên màn hình ứng dụng biên dịch script. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Sau khi người dùng sử dụng tài khoản được cấp đăng nhập thành công thì ứng dụng biên dịch script. Tại màn hình chính của ứng dụng người dùng thực hiện thao tác sau:  **Luồng 1 – Thực hiện mở file log:**   * Quản trị hệ thống click vào nút “Mở file log” trên thanh menu để thực hiện mở file log của hệ thống. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| N/A | | | |
| **Giao diện minh họa** | | | |
| Hình 3.19. Giao diện màn hình mở file log | | | |

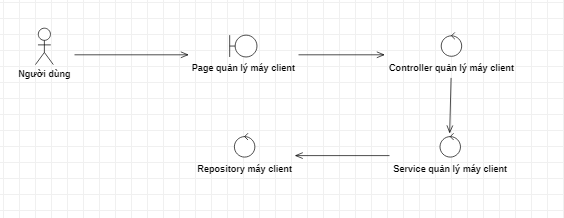
## Gửi email danh sách ứng dụng ngừng hoạt động

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#13** | | **Hiển thị nội dung file log** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Hệ thống kiểm soát trạng thái sẽ định kì gửi danh sách ứng dụng tự ngừng hoạt động mà không có tác động tắt ứng dụng của người dùng đến địa chỉ email của người dùng. | |
| **Tác nhân** | | Hệ thống kiểm soát trạng thái | |
| **Tiền điều kiện** | | Đã đến thời điểm gửi email và có ứng dụng tự ngừng hoạt động. | |
| **Hậu điều kiện** | **Thành công** | Gửi email thành công đến người dùng | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Thời gian định kì gửi email sẽ được cài đặt trong file cấu hình. Cứ sau 1 phút hệ thống tự động kiểm tra đến thời điểm gửi email hay chưa. Sau đó thực hiện luồng sự kiện sau:  **Luồng 1 – Thực hiện lấy danh sách ứng dụng tự ngừng hoạt động:**   * Kiểm tra trạng thái hoạt động của tất cả ứng dụng và đưa ra danh sách các ứng dụng tự dừng hoạt động với 2 điều kiện sau: * Trạng thái của ứng dụng: Ngừng hoạt động; * Trạng thái gửi email của ứng dụng: Chưa được gửi email; * Danh sách gửi gồm các thông tin sau: * IP LAN của máy client; * IP Public của máy client; * Số ứng dụng ngừng hoạt động; * Tổng số ứng dụng hiện tại trên máy client;   **Luồng 2 – Thực hiện gửi email đến người dùng:**   * Luồng 2 xảy ra khi luồng 1 thực hiện thành công. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| N/A | | | |

# PHÂN TÍCH THIẾT KẾ CHỨC NĂNG CHÍNH

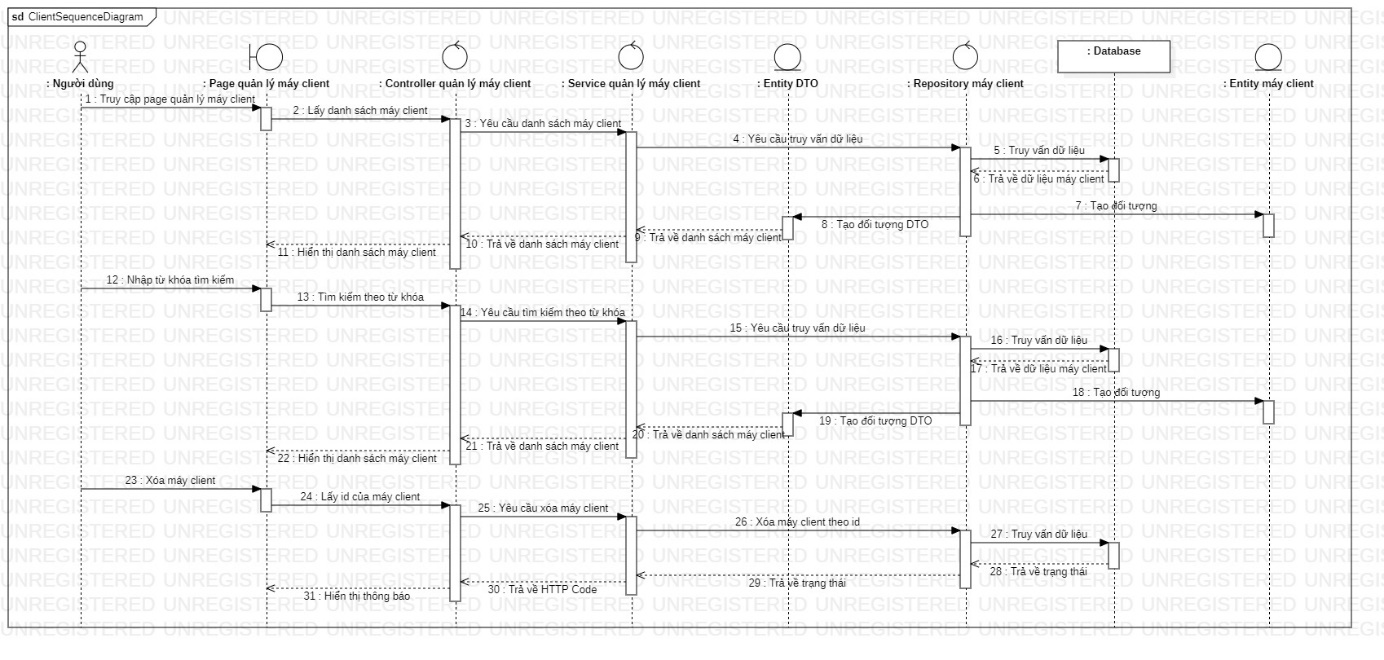
## Quản lý máy client

### Sơ đồ lớp phân tích



Hình 4.1. Sơ đồ lớp phân tích chức năng quản lý máy client

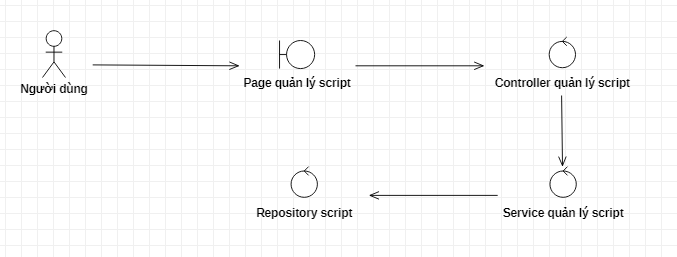
### Sơ đồ trình tự



Hình 4.2. Sơ đồ trình tự chức năng quản lý máy client

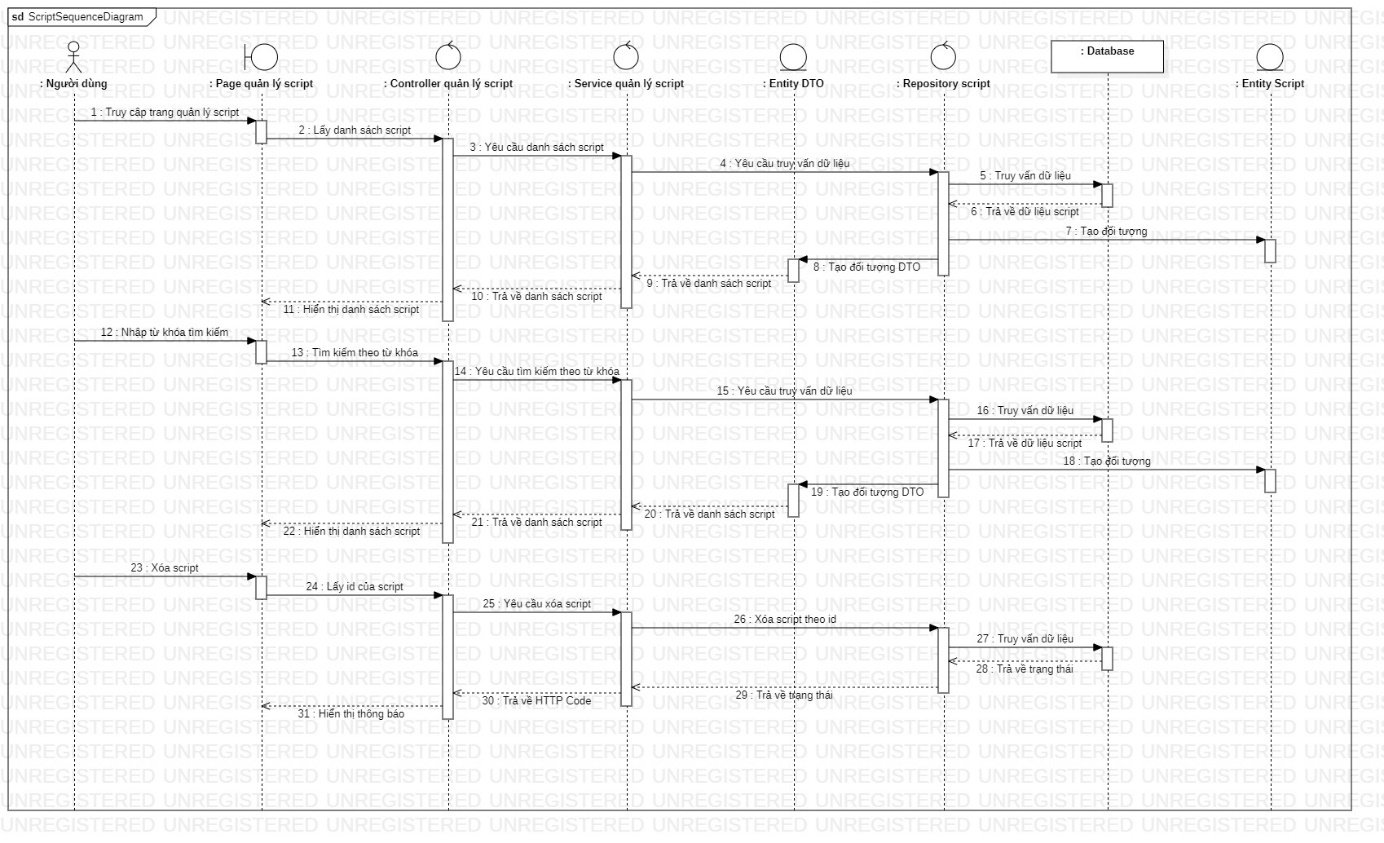
## Quản lý script

### Sơ đồ lớp phân tích

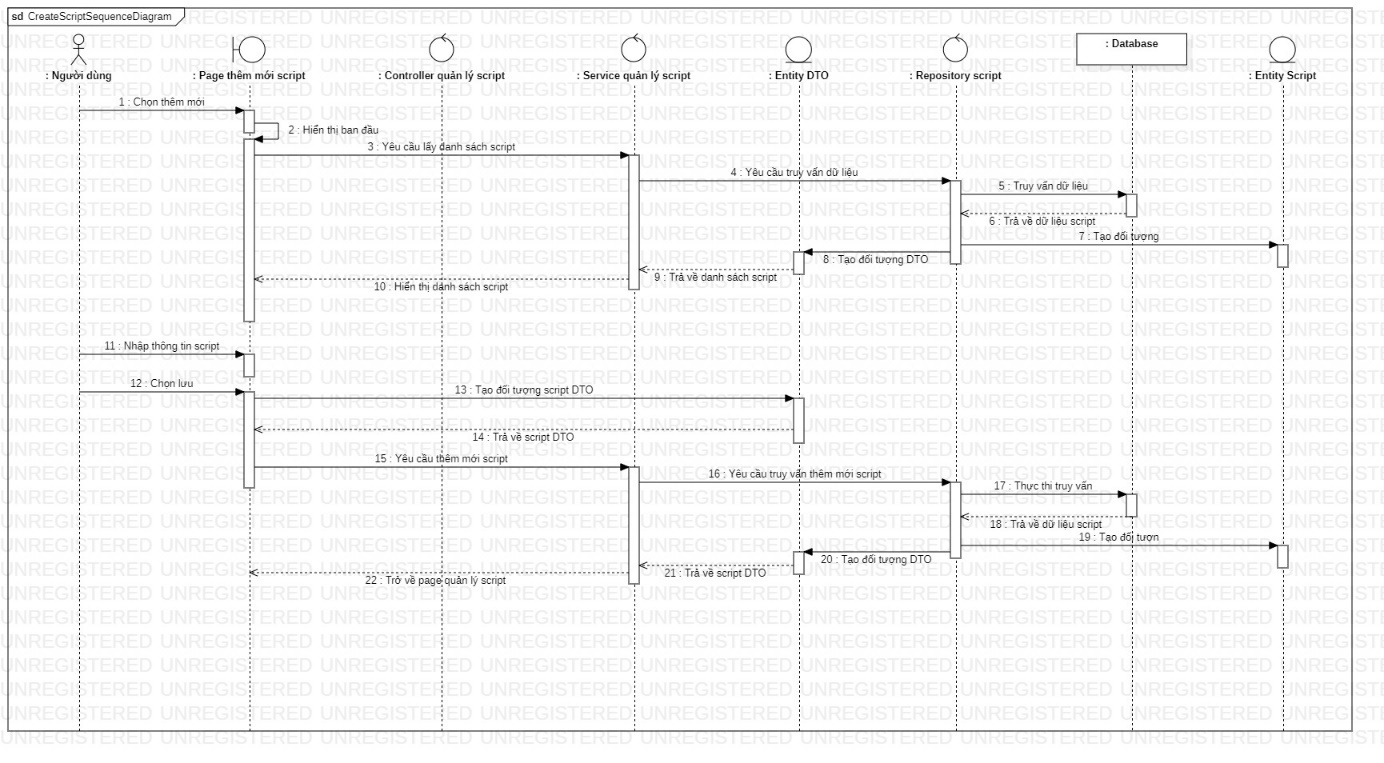


Hình 4.3. Sơ đồ lớp phân tích chức năng quản lý script

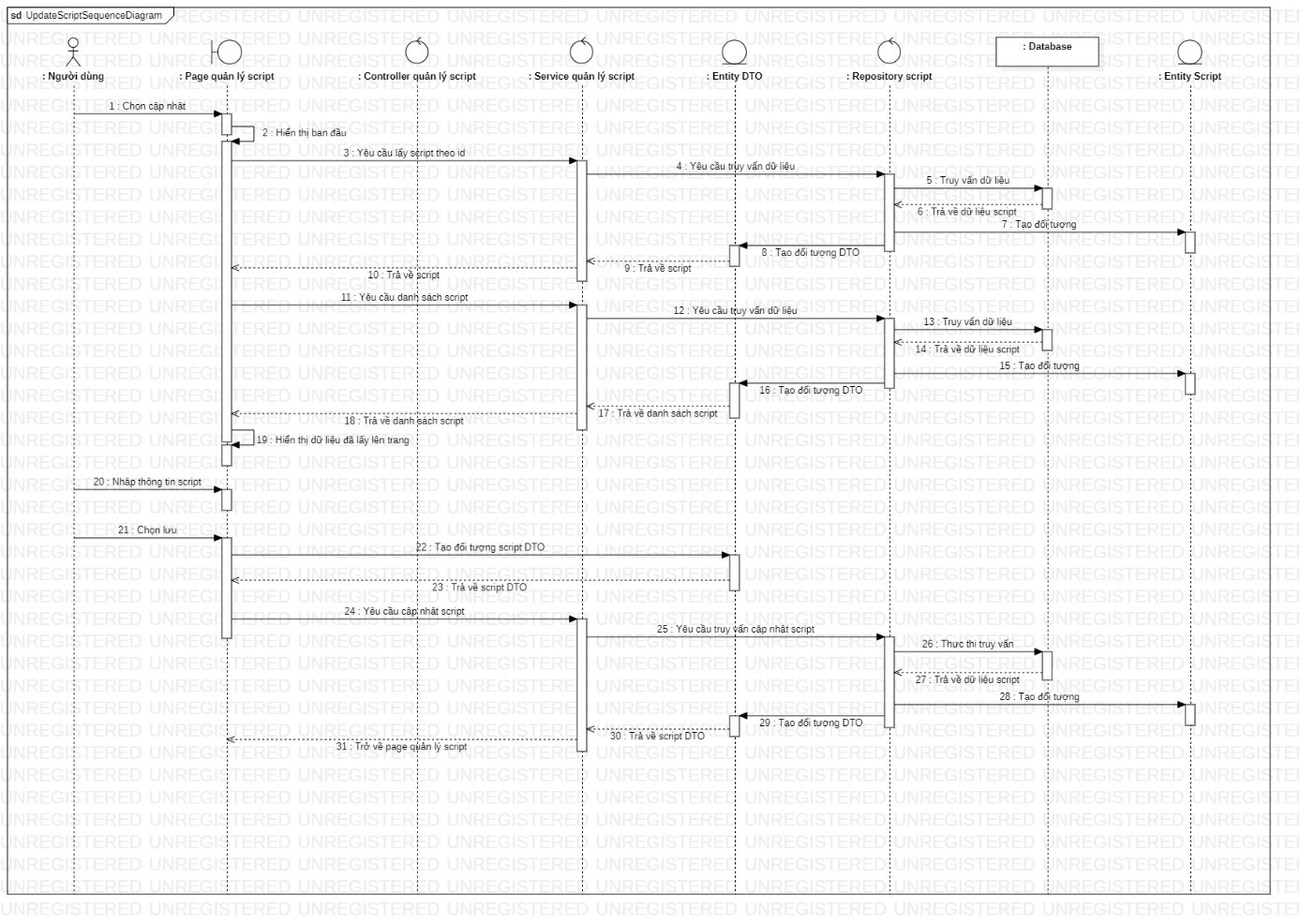
### Sơ đồ trình tự



Hình 4.4. Sơ đồ trình tự chức năng quản lý script



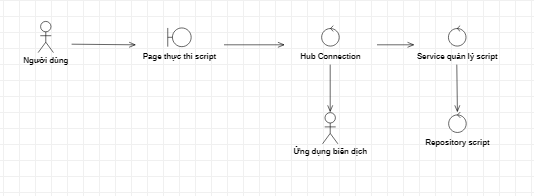
Hình 4.5. Sơ đồ trình tự chức năng thêm mới script



Hình 4.6. Sơ đồ trình tự chức năng cập nhật script

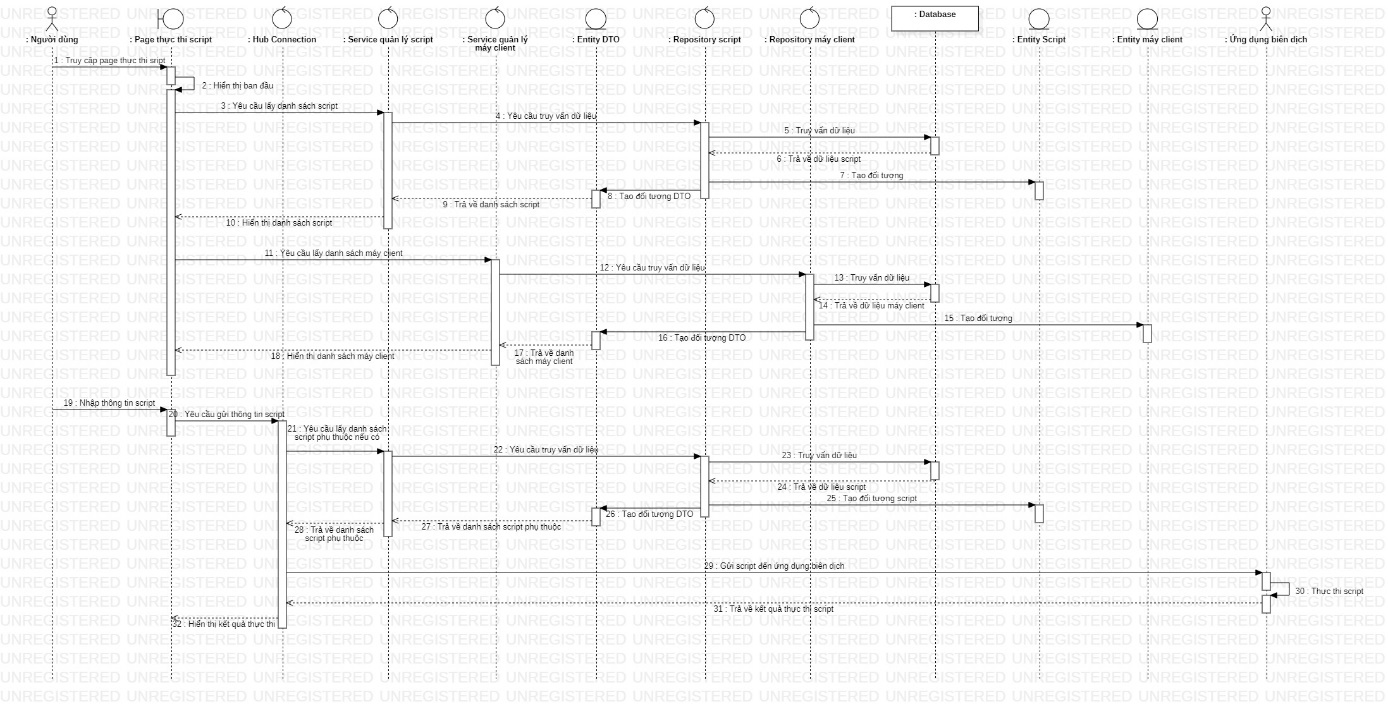
## Thực thi script

### Sơ đồ lớp phân tích



Hình 4.7. Sơ đồ lớp phân tích chức năng thực thi script

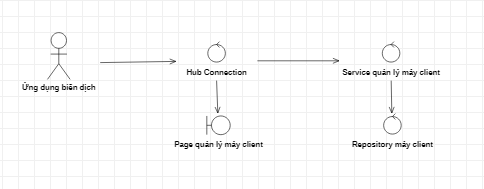
### Sơ đồ trình tự



Hình 4.8. Sơ đồ trình tự chức năng thực thi script

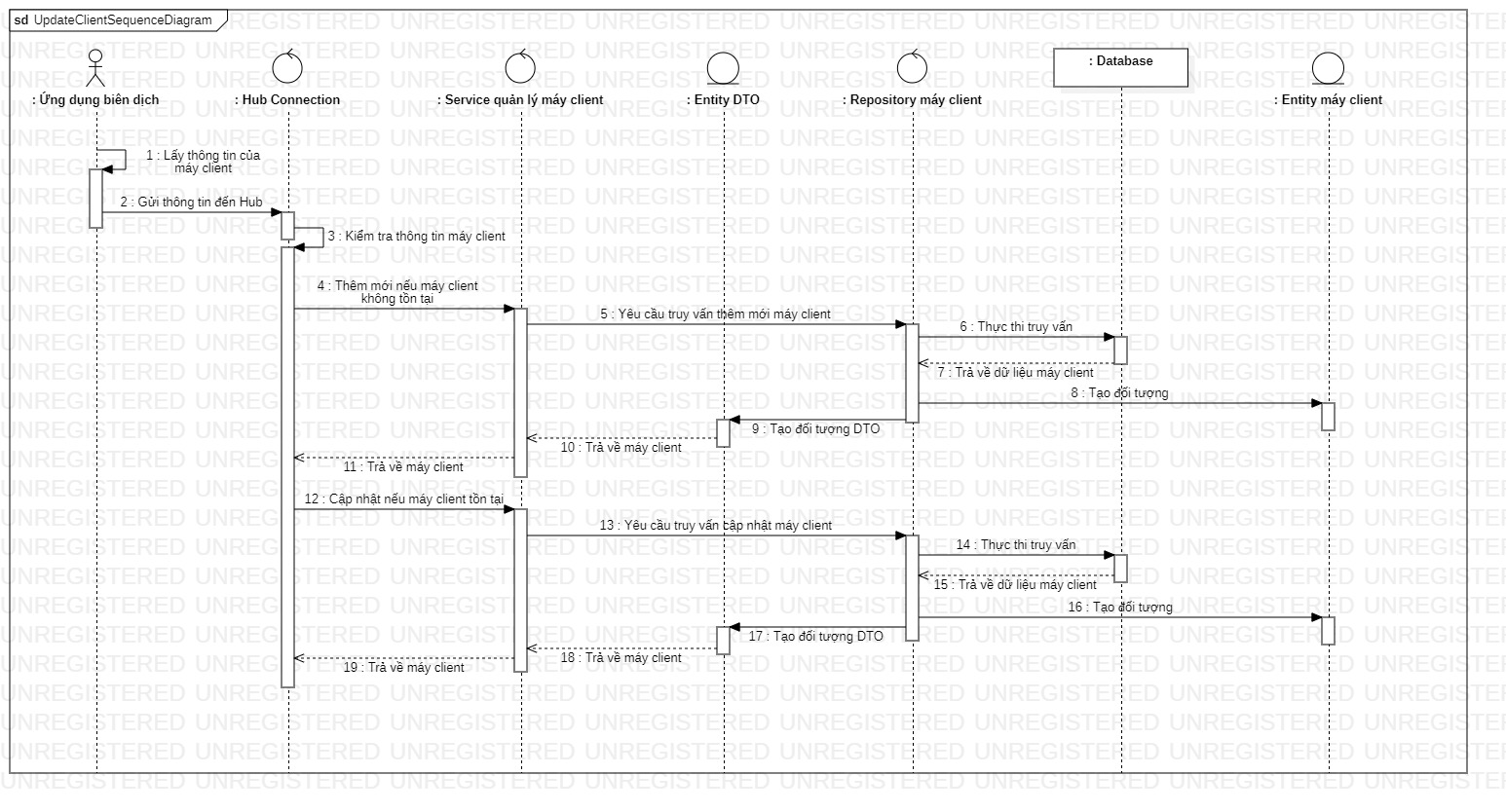
## Cập nhật máy client

### Sơ đồ lớp phân tích



Hình 4.9. Sơ đồ lớp phân tích cập nhật máy client

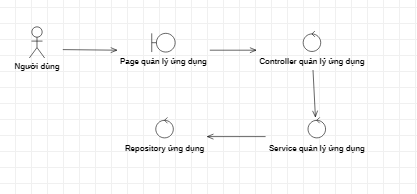
### Sơ đồ trình tự



Hình 4.10. Sơ đồ trình tự chức năng cập nhật máy client

## Quản lý ứng dụng

### Sơ đồ lớp phân tích



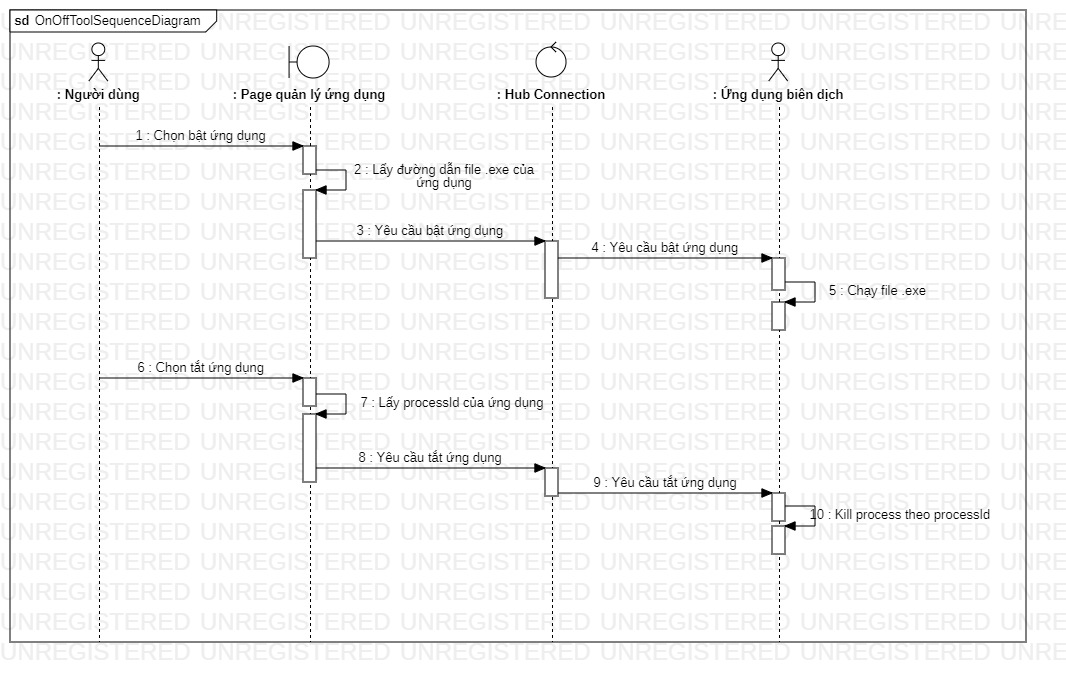
Hình 4.11. Sơ đồ lớp phân tích chức năng quản lý ứng dụng

### Sơ đồ trình tự

Table

Description automatically generated with medium confidence

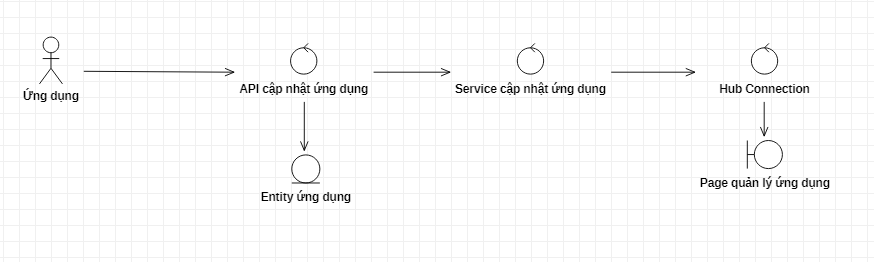
Hình 4.12. Sơ đồ trình tự chức năng quản lý ứng dụng



Hình 4.13. Sơ đồ trình tự tính năng bật/tắt ứng dụng

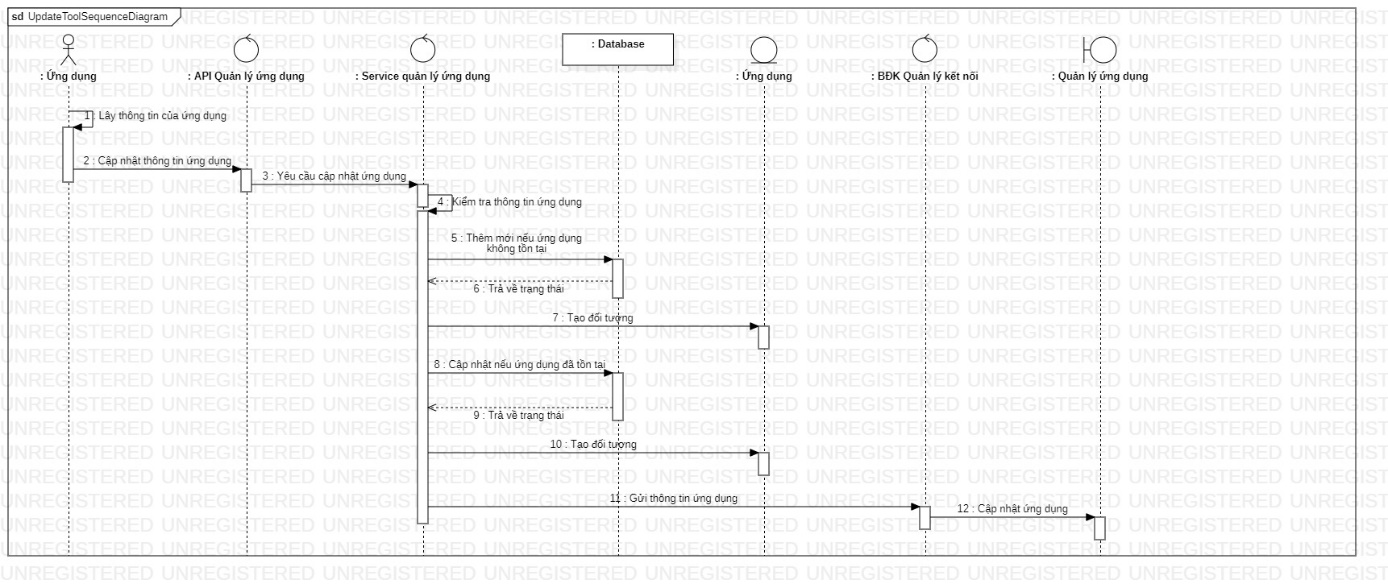
## Cập nhật ứng dụng

### Sơ đồ lớp phân tích



Hình 4.14. Sơ đồ lớp phân tích chức năng cập nhật ứng dụng

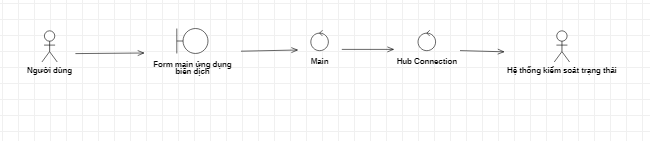
### Sơ đồ trình tự



Hình 4.15. Sơ đồ trình tự chức năng cập nhật ứng dụng

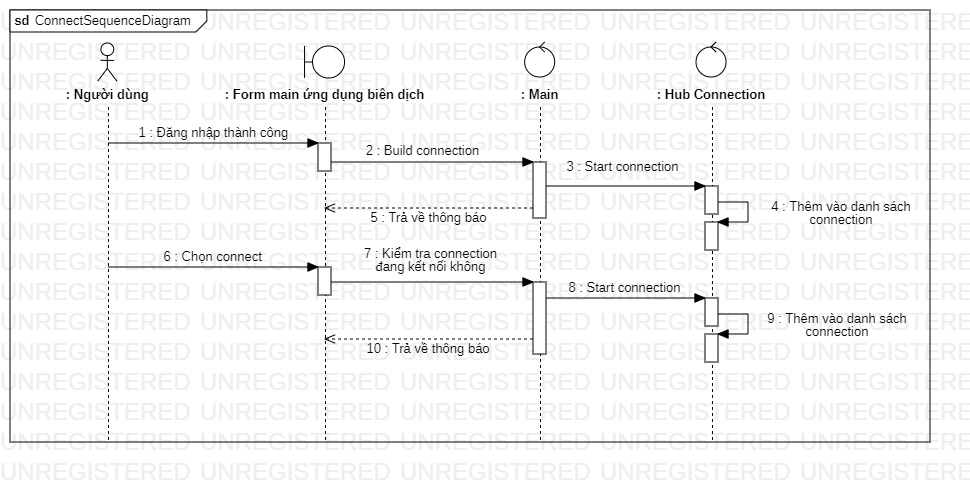
## Kết nối hệ thống biên dịch hỗ trợ với hệ thống kiểm soát trạng thái

### Sơ đồ lớp phân tích



Hình 4.16. Sơ đồ lớp phân tích chức năng kết nối với hệ thống kiểm soát trạng thái

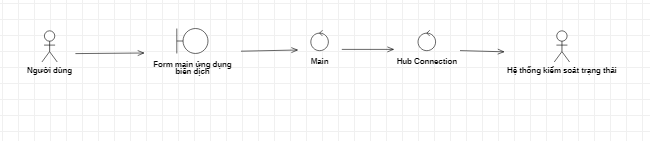
### Sơ đồ trình tự



Hình 4.17. Sơ đồ trình tự chức năng kết nối với hệ thống kiểm soát trạng thái

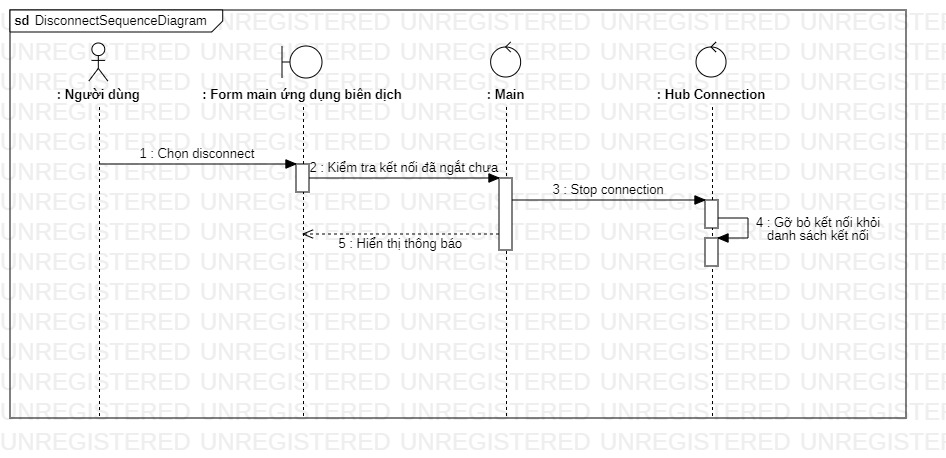
## Ngắt kết nối với hệ thống kiểm soát trạng thái

### Sơ đồ lớp phân tích



Hình 4.18. Sơ đồ lớp phân tích chức năng ngắt kết nối với hệ thống kiểm soát trạng thái

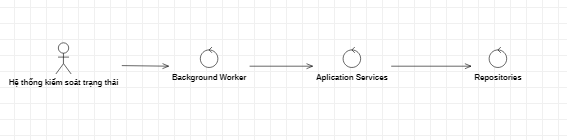
### Sơ đồ trình tự



Hình 4.19. Sơ đồ trình tự chức năng ngắt kết nối với hệ thống kiểm soát trạng thái

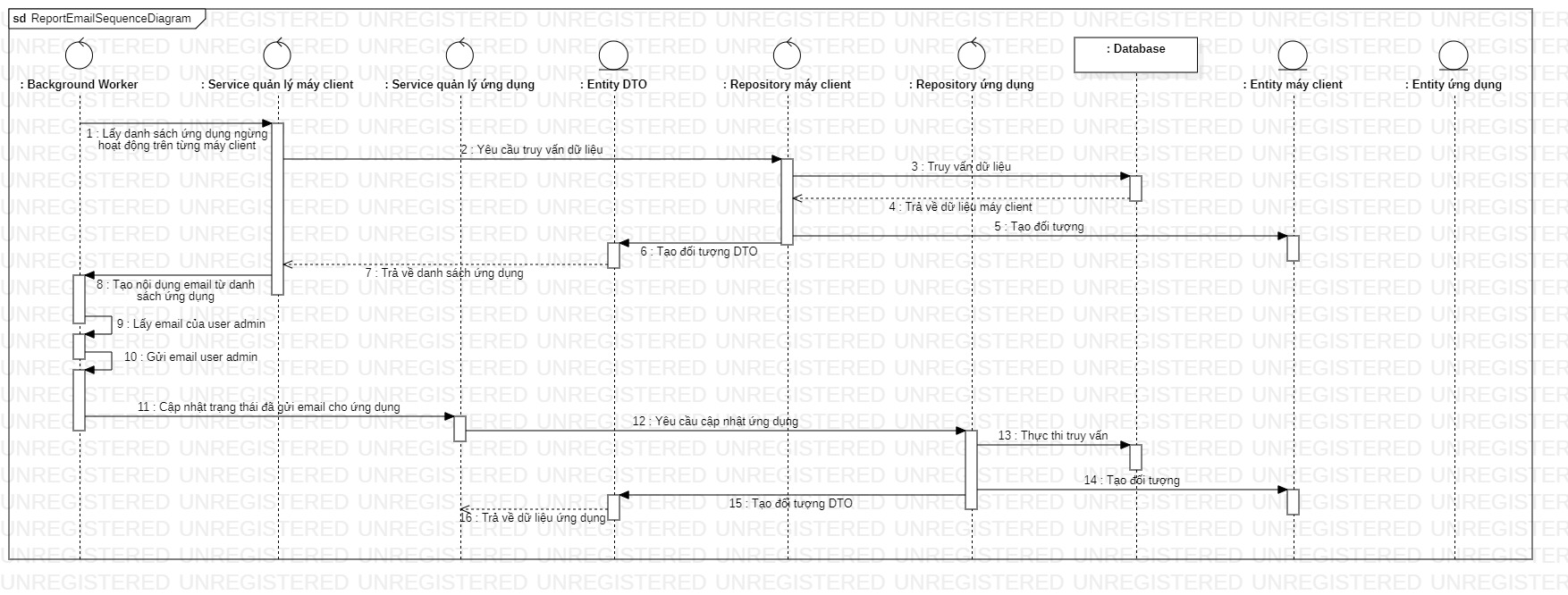
## Gửi email danh sách ứng dụng ngừng hoạt động

### Sơ đồ lớp phân tích



Hình 4.20. Sơ đồ lớp phân tích chức năng gửi email danh sách ứng dụng ngừng hoạt động

### Sơ đồ trình tự



Hình 4.21. Sơ đồ trình tự chức năng gửi email danh sách ứng dụng ngừng hoạt động

# THIẾT KẾ DATABASE VÀ API ĐẶC BIỆT

## Thiết kế cơ sở dữ liệu

Diagram

Description automatically generated

Hình 5.1. Lược đồ cơ sở dữ liệu

|  |  |  |
| --- | --- | --- |
| STT | Bảng | Mô tả |
| 1 | ClientMachines | Thông tin máy client |
| 2 | Scripts | Thông tin script |
| 3 | Dependencies | Quan hệ giữa script với script phụ thuộc |
| 4 | Tools | Thông tin ứng dụng |
| 5 | Users | Thông tin người dùng |
| 6 | Roles | Thông tin nhóm quyền |
| 7 | UserRoles | Quan hệ giữa người dùng với nhóm quyền |

Bảng 4.1. Bảng thiết kế cơ sở dữ liệu

## Định nghĩa database

### Bảng thông tin máy client - ClientMachines

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| 1 | ID | int | Khóa chính |
| 2 | IPLan | varchar | Địa chỉ IP của thiết bị |
| 3 | IPPublic | varchar | Địa chỉ IP thực |
| 4 | LastUpdate | datetime | Thời gian cập nhật lần cuối |
| 5 | UserId | uniqueidentifier | Khóa ngoại người dùng |

Bảng 4.2. Bản thông tin máy client

### Bảng thông tin script – Scripts

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| 1 | ID | int | Khóa chính |
| 2 | Name | nvarchar | Tên script |
| 3 | Content | narchar | Nội dung script |
| 4 | Comment | nvarchar | Mô tả của hảm trong script |

Bảng 4.3. Bảng thông tin script

### Bảng quan hệ script với script phụ thuộc – Dependencies

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| 1 | ID | int | Khóa chính |
| 2 | ScriptId | int | ID của script được phụ thuộc |
| 3 | ScriptDependencyId | int | ID của script phụ thuộc |

Bảng 4.4. Bảng quan hệ script với script phụ thuộc

### Bảng thông tin ứng dụng – Tools

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| 1 | ID | int | Khóa chính |
| 2 | Name | nvarchar | Tên ứng dụng |
| 3 | AppId | uniqueidentifier | ID thực trên ứng dụng |
| 4 | Version | varchar | Phiên bản hiện tại |
| 5 | LastUpdate | datetime | Thời gian cập nhật lần cuối |
| 6 | ExeFilePath | varchar | Đường dẫn tới file .exe |
| 7 | ProcessId | int | ID tiến trình ứng dụng đang chạy |
| 8 | ClientId | int | Khóa ngoại máy client |
| 9 | IsSent | int | Trạng thái gửi email |

Bảng 4.5. Bảng thông tin ứng dụng

### Bảng thông tin người dùng – Users

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| 1 | ID | uniqueidentifer | Khóa chính |
| 2 | Username | varchar | Tên đăng nhập |
| 3 | PasswordHash | varchar | Mật khẩu |
| 4 | Name | nvarchar | Tên người dùng |
| 5 | Surname | nvarchar | Tên họ |
| 6 | Email | varchar | Địa chỉ email |
| 7 | Phone | varchar | Số điện thoại |

Bảng 4.6. Bảng thông tin người dùng

### Bảng thông tin nhóm quyền – Roles

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| 1 | ID | uniqueidentifer | Khóa chính |
| 2 | Name | varchar | Tên nhóm quyền |
| 3 | IsDefault | int | Tự động gán nhóm quyền cho một người dùng mới |
| 4 | IsStatic | int | Không thể xóa hoặc thay đổi tên nhóm quyển |
| 5 | IsPublic | int | Người dùng này có thể xem nhóm quyền của người dùng khác để chể độ public. |

Bảng 4.7. Bảng thông tin nhóm quyền

### Bảng quan hệ người dùng với nhóm quyền – UserRoles

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường dữ liệu | Kiểu dữ liệu | Mô tả |
| 1 | UserId | uniqueidentifer | Khóa chính liên kết của người dùng |
| 2 | RoleId | uniqueidentifer | Khóa chính liên kết của nhóm quyền |

Bảng 4.8. Bảng quan hệ người dùng với nhóm quyền

## Mô hình kiến trúc API

Diagram

Description automatically generated

Hình 5.2. Mô hình kiến trúc API

* Client gọi đến các API EndPoint được public thông qua giao thức HTTP gồm có các phương thức: POST, GET, PUT, DELETE.
* Trong Application Services lấy dữ liệu trong cơ sở dữ liệu thông qua các Repositories Interface. Thực hiện kiểm tra, xử lý logic nghiệp vụ sau đó chuyền qua DTO để trả dữ liệu theo cấu trúc JSON cho Client

## Thiết kế API chi tiết

### Quản lý máy client

|  |  |
| --- | --- |
| EndPoint | Chi tiết |
| /api/clientMachines | * Mục đích: Lấy danh sách máy client theo điều kiện tìm kiếm. * Phương thức: GET * Parameters: * id – number * username – string * ipLan – string * ipPublic – string * clientStatus – number (offline: 0, online: 1) |
| /api/clientMachines/{id} | * Mục đích: Xóa một máy client theo id * Phương thức: DELETE * Parameters: id – int |

### Quản lý script

|  |  |
| --- | --- |
| EndPoint | Chi tiết |
| /api/scripts | * Mục đích: Lấy danh sách script theo điều kiện tìm kiếm. * Phương thức: GET * Parameters: * id – number * name – string |
| /api/scripts/{id} | * Mục đích: Lấy một script theo id * Phương thức: GET * Parameters: id – int |
| /api/scripts | * Mục đích: Thêm mới một script * Phương thức: POST * Request body: * name – string * content – string * comment – string * dependencyIds – array(int) |
| /api/scripts/{id} | * Mục đích: Cập nhật một script * Phương thức: PUT * Parameters: id – number * Request body: * name – string * content – string * comment – string * dependencyChosens – array(object) |
| /api/script/{id} | * Mục đích: Xóa một script theo id * Phương thức: DELETE * Parameters: id – number |

### Quản lý ứng dụng

|  |  |
| --- | --- |
| EndPoint | Chi tiết |
| /api/tools | * Mục đích: Lấy danh sách ứng dụng theo điều kiện tìm kiếm. * Phương thức: GET * Parameters: * id – number * name – string * appId – string * version – string * toolStatus – number (inactive: 0, active: 1) * clientId – number |
| /api/tools /{id} | * Mục đích: Xóa một ứng dụng theo id * Phương thức: DELETE * Parameters: id – number |
| /api/tools/report | * Mục địch: Thêm mới hoặc cập nhật thông tin cho một ứng dụng. * Phương thức: POST * Request body: * name – string * appId – guid * version - string * ipLan – string * ipPublic – string * clientId – number * exeFilePath – string * processId – number * lastUpdate – datatime * sentMail - boolean |

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

Trong quá trình xây dựng hệ thống quản lý trạng thái ứng dụng, em đã thực hành và làm các chương trình thử nghiệm như: Tạo một ứng dụng web quản lý đơn giản với ABP Framework, thực hiện kết nối giữa client với server bằng SignalR, tạo chương trình thực thi script với ClearScript tương ứng một số công nghệ đã giới thiệu trong Chương 2. Mục đích là giải quyết vấn đề kỹ thuật chưa rõ ràng và chọn giải pháp phù hợp để đáp ứng yêu cầu hệ thống đề ra.

Sau khi hoàn thiện đề tài của mình, em đã tích lũyđược nhiều kiến thức và kỹ năng trong quá trình phát triển phần mềm như:

* Phân tích hệ thống theo yêu cầu thực tế;
* Tìm hiểu và phân tích giải pháp phù hợp cho bài toán;
* Thiết kế các mô hình đối tượng, các sơ đồ quan hệ;
* Kỹ năng quản lý tài liệu;
* Kỹ năng quản lý công việc.

Thêm vào đó cũng đem lại cho em sự hiểu biết về kiến trúc hệ thống dưới các góc nhìn khác nhau. Nắm bắt được kiến trúc trong hệ sinh thái ASP.Net. Đồng thời em cũng hiểu được các vấn đề trong thiết kế, cùng các khó khăn khi tìm hiểu công nghệ và áp dụng vào thực tế. Đó là những kinh nghiệm vô cùng quý báu giúp em có thêm nhiều cơ hội trong việc phát triển bản thân và công việc sau này.

Với hệ thống quản lý trạng thái ứng dụng đã đáp ứng đúng với yêu cầu hiện tại đặt ra, tuy nhiên hệ thống sẽ quản lý hiệu quả và sự trải nghiệm tốt từ người dùng khi phát triển sâu hơn các chức năng chính có thể kể đến như:

* Chức năng thêm mới và cập nhật script trong quản lý script cần lưu trữ thông tin người dùng nào thêm mới, cập nhật và thời gian cập nhật vào khi nào. Hiện tại, quản lý script chỉ đáp ứng lưu trữ các thông tin liên quan đến script.
* Thêm tính bảo mật đối với máy client khi người dùng thao tác với chức năng thực thi script. Hiện tại người dùng có thể thấy được toàn bộ đường dẫn vật lý của file trên máy client. Điều đó rất nguy hiểm nếu người dùng thực thi đoạn script xóa một file System trên máy client. Ý tưởng cho vấn đề này là chỉ cho phép người dùng truy xuất dữ liệu trên một thư mục đã cố định sẵn trên ứng dụng biên dịch.
* Phát triển thêm tính năng giúp người dùng có thể tự cấu hình thêm, bớt nút chức năng đối với mỗi ứng dụng được hệ thống quản lý, việc đó làm tăng tính linh hoạt đối với quản lý ứng dụng.

# TÀI LIỆU THAM KHẢO

[1] “ABP Documentation”. https://docs.abp.io/en/abp/latest (truy cập tháng 10 15, 2020).

[2] “Introducing the ABP vNext”. http://blog.abp.io/abp/Abp-vNext-Announcement (truy cập tháng 10 15, 2020).

[3] “Best Practices/Module Architecture | ABP Documentation”. https://docs.abp.io/en/abp/latest/Best-Practices/Module-Architecture (truy cập tháng 10 15, 2020).

[4] “Domain-Driven Design in ASP.NET Core applications”. [Online]. Available at: https://enlabsoftware.com/development/domain-driven-design-in-asp-net-core-applications.html

[5] “Implementing Domain Driven Design With ABP”. [Online]. Available at: https://docs.abp.io/en/abp/4.0/Domain-Driven-Design-Implementation-Guide

[6] “Introduction to ASP.NET Core SignalR”. [Online]. Available at: https://docs.microsoft.com/vi-vn/aspnet/core/signalr/introduction?view=aspnetcore-3.0

[7] “ClearScript FAQtorial”. [Online]. Available at: https://microsoft.github.io/ClearScript/Tutorial/FAQtorial.html

[8] “V8 Script Engine -Documentation Overview”. [Online]. Available at: https://v8.dev/docs

[9] “Entity Framework (EF) Core”. [Online]. Available at: https://docs.microsoft.com/en-us/ef/core/

[10] “Tài liệu yêu cầu hệ thống của Công ty Devmoba”.