# Tổng quan hệ thống

## Vấn đề

* Máy client là máy tính được sử dụng để chạy ứng dụng.
* Ứng dụng là các ứng dụng tự động chạy 24/7 trên nhiều máy client khác nhau để làm một công việc cụ thể trong linh vực kinh doanh của công ty.
* Vấn đề người quản lý gặp phải khi thực hiện các công việc:
* Xem các thông tin về tài nguyên của máy client;
* Xem file log của ứng dụng;
* Cài đặt một ứng dụng trên nhiều máy client cùng 1 lúc;

=> Thì đều phải remote vào từng máy client.

* Một vấn đề quan trọng đó là khi ứng dụng chạy trên máy client **ngừng hoạt động** do gặp sự cố lỗi mà không có sự phản hồi đến người quản lý.
* Các ứng dụng có thể liên quan đến nhau, nếu một ứng dụng ngừng hoạt động sẽ làm ảnh hưởng đến ứng dụng khác gây ảnh hưởng đến hoạt động kinh doanh của công ty.

=> Người quản lý thường xuyên phải rà soát tất cả các ứng dụng trên máy client.

=> Do thực hiện các việc lần lượt trên từng máy client sẽ tốn nhiều thời gian và công sức.

## Yêu cầu hệ thống

1. Quản lý máy client

* Các máy client được kết nối đến hệ thống;
* Người dùng có thể xem danh sách thông tin của máy client bao gồm: Địa chỉ IP LAN, địa chỉ IP thực, % CPU, %RAM, Trạng thái kết nối (Online/Offline);
* Hệ thống tự động cập nhật các thông tin đã kể trên theo thời gian thực.

1. Quản lý ứng dụng

* Người dùng có thể xem danh sách thông tin của ứng dụng bao gồm: Tên ứng dụng, ID thực của ứng dụng, version, máy client đang ứng dụng, trạng thái hoạt động (Đang hoạt động/ Ngừng hoạt động),…
* Người dùng có thể bật hoặc tắt một ứng dụng từ hệ thống mà không cần truy cập từ xa vào máy client mà có ứng dụng đó.
* Ứng dụng sẽ định kì gửi thông tin đến hệ thống theo thời gian người dùng cấu hình trong hệ thống.

1. Quản lý script

* Script là tập hợp các câu lệnh được viết bằng mã javascript. Trong một script có các hàm xử lý công việc theo mục đích của người dùng. Với mỗi hàm người dùng có thể viết mô tả cho hàm đó theo quy tắc hệ thống đã đặt ra.
* Người dùng có thể thêm mới, cập nhật và xóa script.
* Nội dung script được người dùng thêm vào hệ thống sẽ bao gồm: Có một hàm chứa các xử lý chính của script đó (gọi là hàm main), các hàm được người dùng định nghĩa, với mỗi hàm được định nghĩa người dùng cần mô tả cho hàm đó theo quy tắc mà hệ thống đã quy định;

1. Thực thi script

* Người dùng có thể thực thi một script trên nhiều máy client cùng một lúc;
* Người dùng có thể nhập vào một script tùy ý, sau đó thực thi script đó;
* Có thể xem được kết quả trả về sau khi thực thi thành công

1. Gửi email thông báo ứng dụng ngừng hoạt động

* Hệ thống tự động kiểm tra ứng dụng ngừng hoạt động mà gặp lỗi sau đó gửi email thống báo đến người quản lý.

## Hình dung tổng thể

* Môi trường chạy của hệ thống là môi trường web. Web Server trong sơ đồ chính là hệ thống kiểm soát trạng thái ứng dụng.
* Hệ thống nhận các yêu cầu của người dùng và xử lý trả về kết quả thông qua trình duyệt.
* Hệ thống có lưu thông tin của máy client, ứng dụng, script, và một số thông tin khác nên có database.
* Ứng dụng biên dịch là một ứng dụng windown cái trên máy client có nhiệm:
* Giữ kết nối với hệ thống;
* Nhận script gửi từ hệ thống;
* Thực thi script và trả về kết quả;
* Cập nhật trạng thái của máy client;
* Bật tắt ứng dụng đã được cài trên máy client;
* Ứng dụng mà hệ thống quản lý cập nhật trạng thái hoạt động đến hệ thống thông qua HTTP Request.

# Kiến trúc hệ thống

Để xây dựng hệ thống kiểm soát trạng thái em đã sử dụng Domain Driven Design. Thì Framework em lựa chọn sử dụng cũng theo hướng tiếp cận này. Em sẽ giới thiệu một chút về Domain Driven Design.

## Domain Driven Design

Các nghiệp vụ sau khi phân tích sẽ được triển khai trong các layer: (Mô tả hình kiến trúc layer)

* 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: Chịu trách nhiệm đối với các business object để thực hiện các tác vụ ứng dụng. Triển khai các xử lý logic của hệ thống.
* Domain Layer: Bao gồm các business object và các quy tắc nghiệp vụ chính.
* Infrastructure Layer: Bao gồm kết nối với cơ sở dữ liệu thông qua ORM

Ưu điểm:

* Tạo ra ngôn ngữ chung giúp cho việc giao tiếp giữa nhóm phát triển với các chuyên gia nghiệp vụ.
* Linh hoạt hơn để thay đổi vì các yêu cầu chức năng phụ hợp theocách tự nhiên.
* Có thể độc lập triển khai các chức năng cùng 1 lúc.
* 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, hạn chế trùng lặp code và giảm chi phí bảo trì.

Nhược điểm:

* Yêu cầu hiểu rõ nghiệp vụ của bài toán.
* Chí phát triển cao phù hợp với dự án ngắn hạn.