BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM
KHOA CNTT

ĐỒ ÁN

Đề tài: Website quản lý công văn

Giáo viên hướng đẫn: Ths. Nguyễn Hữu Quang
Sinh viên thực hiện: Nguyễn Quang Thọ
Lớp: ĐHHTT 11
Mssv: 15086971

TP.HCM - 7/2020

LỜI CẢM ƠN

Để nhóm em có thể đi đến chặng đường hiện tại là nhờ sự giúp đỡ, động viên rất nhiều từ Thầy/Cô, gia đình và bạn bè xung quanh từ lúc bắt đầu đến lúc hoàn thành hiện thực đề tài.

Với tấm lòng biết ơn sâu sắc nhóm em xin gửi lời cảm ơn đến quý Thầy/Cô giảng dạy bộ môn Hệ Thống Thông Tin Trường Đại học Công Nghiệp Thành Phố Hồ Chí Minh đã truyền đạt cho nhóm em những kiến thức quý báu nhờ đó nhóm em mới có thể hiện thực được đề tài đưa ra. Đặc biệt em xin chân thành cảm ơn Thầy ThS. Nguyễn Hữu Quang đã giúp đỡ nhóm em rất nhiều trong suốt quá trình thực hiện đề tài. Qua những buổi trao đổi với thầy đã cho chúng em nhiều lời khuyên, giải pháp để có thể hoàn thiện đề tài tốt hơn.

Trong quá trình hiện thực đề tài chắc chắn nhóm em sẽ không thể tránh khỏi thiếu sót vì thế nhóm em rất mong nhận được sự góp ý từ phía Thầy/Cô để có thể nâng cao kiến thức, hiểu biết phục vụ cho phát triển công việc sau này.

LỜI MỞ ĐẦU

Đối với bất cơ quan, tổ chức nào thì việc quản ý, lưu trữ, chuyển giao công văn có vai trò rất quan trọng. Việc lưu trữ công văn bằng bảng cứng trên giấy tờ gây ra hao phí tài nguyên giấy, hao phí điện tích lưu trữ, dễ bị thất lạc tài liệu, hư hỏng bởi các yếu tố tự nhiên và con người. Việc quản lý và chuyển giao công văn cần nhiều chi phí về nhân lực và thời gian.

Với mong muốn tiết kiện chi phí về nhân lực và thời gian trong việc quản lý, lưu trữ công văn nhóm chúng em đưa ra ý tưởng "Xây dựng website quản lý công văn" cho đề tài khóa luận của mình.

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……...

Ngày … Tháng …. Năm…….

Giảng viên hướng dẫn

ThS. Nguyễn Hữu Quang

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN 1

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Ngày … Tháng …. Năm…….

Giảng viên phản biện 1

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN 2

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

Ngày … Tháng …. Năm…….

Giảng viên phản biện 2

ABSTRACT

For any agency or organization, the management, storage and transfer of official letters are very important. Storing official documents with hard boards on paper causes waste of paper resources, wastage of storage charge, and the loss of documents, damage by natural and human factors. The management and delivery of dispatch need a lot of manpower and time.

With the desire to save the cost of manpower and time in managing and archiving documents, our group came up with the idea of "Building official documents management website".

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU

1.1 Tổng quan và lý do chọn đề tài

Trong bất kì lĩnh vực nào thì công tác quản lý luôn giữ một vai trò vô cùng quan trọng, trong đó có công tác quản lý công văn.

Đối với các cơ quan, tổ chức, công tác văn thư lưu trữ có vai trò rất quan trọng đối với hoạt động của bất kỳ tổ chức nào vì nó đảm bảo thông tin dưới dạng công văn được cập nhật, lưu trữ và chuyển tới người xử lý một cách kịp thời và chính xác. Trước đây, công tác quản lý công văn, công văn được thực hiện thủ công dưới hình thức lưu trữ hồ sơ, sổ sách. Việc làm thủ công này đã gặp không ít khó khăn và hạn chế, đặc biệt trong việc tìm kiếm, sắp xếp. Các ông văn có thể bị thất lạc và rất kho khăn trong việc theo dõi tình trạng thực hiện công văn. Ngày nay, cùng với sự phát triển mạnh mẽ, len lỏi vào tất cả các lĩnh vực của công nghệ thông tin, cơ quan có thể dễ dàng quản lý công văn với các sản phẩm phần mềm ưu việt. Chính vì vậy, quản lý công văn là một trong những vấn đề quan trọng trong các cơ quan, công ty, … là một trong những yếu tố góp phần cải thiện và nâng cao hiệu quả công việc. Ứng dụng công nghệ thông tin trong công tác quản lý công văn là rất cần thiết và cần phải chú trọng.

Nhận thấy ở một số cơ quản, tổ chức chưa cón thiếu phần mềm hỗ trợ công tác quản lý công văn. Vì vậy, em chọn đề tài xây dựng website quản lý công văn. Mặc dù còn có những hạn chế về kiến thức cũng như kinh nghiệm thực tế nhưng em đã cố gắng tìm hiểu và vận dụng công nghệ mới để hoàn thành website này.

1.2 Mục tiêu đề tài

Mục tiêu phát triển một ứng dụng hệ thống thông tin cho các cơ quan, tổ chức hỗ trợ việc lưu trữ, quản lý công văn, phân công thực hiện công văn và theo đõi tình trạng thực hiện công văn.

1.3 Phạm vi đề tài

Người dùng có thể xem danh sách công văn đi mà người đó dược nhận và coi danh sách công văn đến mà người đó được giao thực hiện. Xem chi tiết công văn và file dính kèm. Theo dõi trạng thái, đề xuất hoàn thành đối với công văn đến.

Quản trị viên có thể tạo công văn đến phân công người thực hiện và tạo công văn đi phân công người nhận. Có thể sửa, xóa công văn khi tạo sai. Theo dõi trang thái, chấp nhận hoặc từ chối đề xuất hoàn thành công văn đến. Quản ý người dùng trong hệ thống.

Đề tài chỉ tập trung quản lý công văn trong nội bộ của một cơ quan, tổ chức. Các công văn đã được xử lý đầy đủ theo quy trình của cơ quan, tổ chức đó. Hệ thống được xây dựng và phát triển trên nền tảng web.

1.4 Mô tả yêu cầu phi chức năng

Tính dễ sử dụng: Trang web thân thiện với người dùng không quá khó học khi thực hiện các chức năng.

Hiệu suất / thời gian đáp ứng: hiệu suất ổn định giúp trải nghiệm của người dùng thoải mái hơn.

1.5 Mô tả yêu cầu chức năng

- Tạo công văn đến phân công người thực hiện, tạo công văn đi phân công người nhận.
- Quản lý danh sách công văn đi, quản lý danh sách công văn đến.
- Quản lý người dùng.
- Theo dõi trạng thái công văn đến

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Quy trình xử lý công văn

Công văn đến

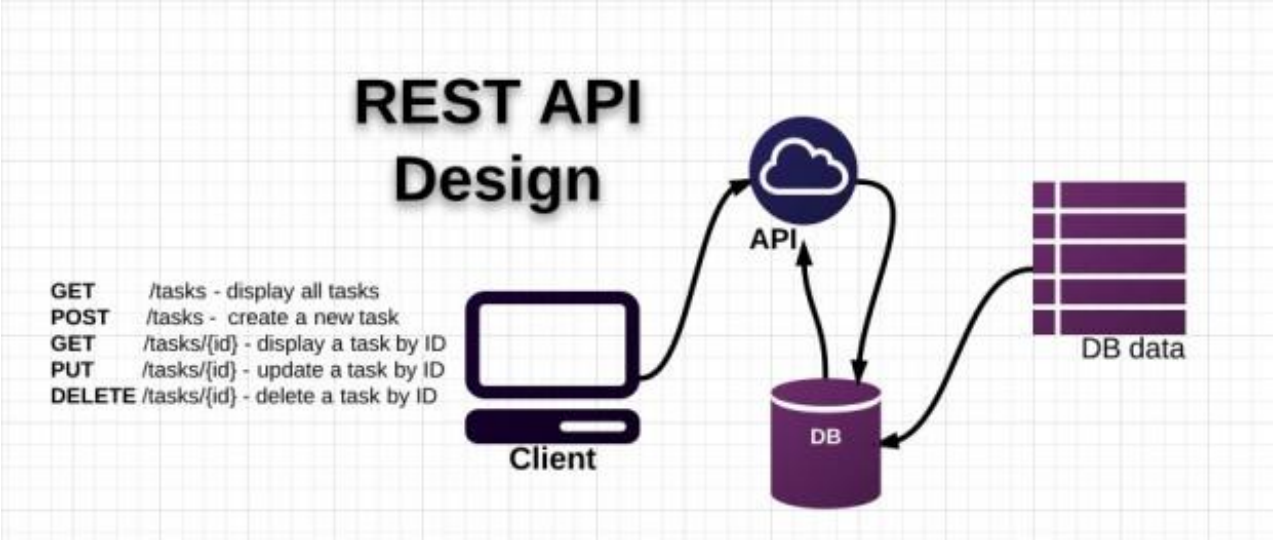
- Sau khi cơ quan tổ chức nhận được công văn đến, lãnh đạo sẽ là người phân công người thực hiện công văn đó.
- Các thông tin về công văn đến(số văn bản, loại văn bản, ngày đến, đơn vị gửi, nội dung, thời hạn) cùng với nội dung bút phê của lãnh đạo sẽ được quản trị viện lưu lại và gửi đến người thực hiện.
- Người thực hiện theo dõi và giải quyết công văn đó.

Công văn đi

- Sau khi công văn đi đươc soạn thảo sẽ được lãnh đạo kiểm tra và phê quyệt.
- Công văn đi được phê quyệt sẽ được gửi đi.
- Quản trị viên lưu trữ lại thông tin công văn đi(số văn bản, loại văn bản, ngày tháng, nội dung, người thực hiện, người nhận, tác giả)

2.2 RESTful API

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 2.1 REST API design

Hình 2.1 REST API design

Diễn giải các thành phần:

API (**A**pplication **P**rogramming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (**RE**presentational **S**tate **T**ransfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Ưu điểm của REST:

- Hiệu năng: các thành phần đảm bảo được việc giao tiếp theo đúng một quy ước giúp hệ thống có thể vận hành tốt hơn.
- Tính khả biến: với các hệ thống cần thay đổi các tài nguyên liên tục, sử dụng REST với việc tạo request đơn giản sẽ giúp mọi chuyển trở nên đơn giản hơn.
- Tính mở rộng: các hệ thống REST có khả năng mở rộng rất cao nhờ sự tách biệt giữa các thành phần và các quy ước giao tiếp được quy định sẵn.
- Tính linh hoạt: việc chuẩn hoá interface giúp hệ thống trở nên linh hoạt hơn, có thể sử dụng cho cho nhiều nền tảng khác nhau, mobile, web, ...
- Trong sáng: trong giao tiếp giữa các thành phần, các request trở nên rất rõ ràng, dễ hiểu.
- Đơn giản: xây dựng rất đơn giản, giúp xây dựng các uri cho resource.
- Tính tin cậy: khó để xảy ra lỗi trong giao tiếp giữa các thành phần gây sụp đổ hệ thống.

Nhược điểm của REST:

- Chỉ hoạt động trên các giao thức HTTP.

Lý do lựa chọn:

- Để dàng mở rộng, có thể tương tác với trên nhiều nền tảng web, mobile.

2.3 Môi trường Nodejs

Node.js là một JavaScript runtime được build dựa trên engine JavaScript V8 của Chrome. Node.js sử dụng kiến trúc hướng sự kiện event-driven, mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.

Đặc điểm của Node js:

- **Không đồng bộ và phát sinh sự kiện (Event Driven)**: Tất các các APIs của thư viện Node.js đều không đồng bộ, nghĩa là không blocking (khóa). Nó rất cần thiết vì Node.js không bao giờ đợi một API trả về dự liệu. Server chuyển sang một API sau khi gọi nó và có cơ chế thông báo về Sự kiện của Node.js giúp Server nhận đựa phản hồi từ các API gọi trước đó.
- **Chạy rất nhanh**: Dựa trên V8 Javascript Engine của Google Chrome, thư viện Node.js rất nhanh trong các quá trình thực hiện code.
- **Các tiến trình đơn giản nhưng hiệu năng cao**: Node.js sử dụng một mô hình luồng đơn (single thread) với các sự kiện lặp. Các cơ chế sự kiện giúp Server trả lại các phản hồi với một cách không khóa và tạo cho Server hiệu quả cao ngược lại với các cách truyền thống tạo ra một số lượng luồng hữu hạn để quản lý request. Nodejs sử dụng các chương trình đơn luồng và các chương trình này cung cấp các dịch vụ cho số lượng request nhiều hơn so với các Server truyền thống như Apache HTTP Server.

Lý do lựa chọn:

- Node.js chạy non-blocking việc hệ thống không phải tạm ngừng để xử lý xong một request sẽ giúp cho server trả lời client gần như ngay tức thì.
- Javascript là ngôn ngữ lập trình hướng sự kiện, mà trong lập trình thời gian thực, cách tiếp cận bằng lập trình sự kiện là cách tiếp cận khôn ngoan nhất.
- Hỗ trợ các framework để tạo API và websocket như: Express js và socket io.

2.4 MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql. MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ. Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Ưu điểm của mongoDB:

- Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.
- Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.
- MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:
- Trường dữ liệu "_id" luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
- Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
- Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.

Nhược điểm của mongoDB:

- Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.
- Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.

Lý do lựa chọn:

- Sử dụng dữ liệu dưới dạnh JSON nên để dành giao tiếp, sử dụng trên javascript và kiến trúc Restful Api (vì api cũng giao tiếp bằng các endpoint methol trả về kiểu dữ liệu JSON)
- Có tốc độ truy xuất nhanh phù hợp với để xây dựng các hệ thống realtime.

2.5 React

ReactJS là một thư viện JavaScript mã nguồn mở được thiết kế bởi Facebook để tạo ra những ứng dụng web hấp dẫn, nhanh và hiệu quả với mã hóa tối thiểu. Mục đích cốt lõi của ReactJS không chỉ khiến cho trang web phải thật mượt mà còn phải nhanh, khả năng mở rộng cao và đơn giản. Sức mạnh của nó xuất phát từ việc tập trung vào các thành phần riêng lẻ. Chính vì vậy, thay vì làm việc trên toàn bộ ứng dụng web, ReactJS cho phép một developer có thể phá vỡ giao diện người dùng phức tạp thành các thành phần đơn giản hơn.

Lý do lựa chọn:

- Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật len DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
- Render tầng server: Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
- Có nhiều công cụ phát triển giúp debug code dễ dàng hơn.
- Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

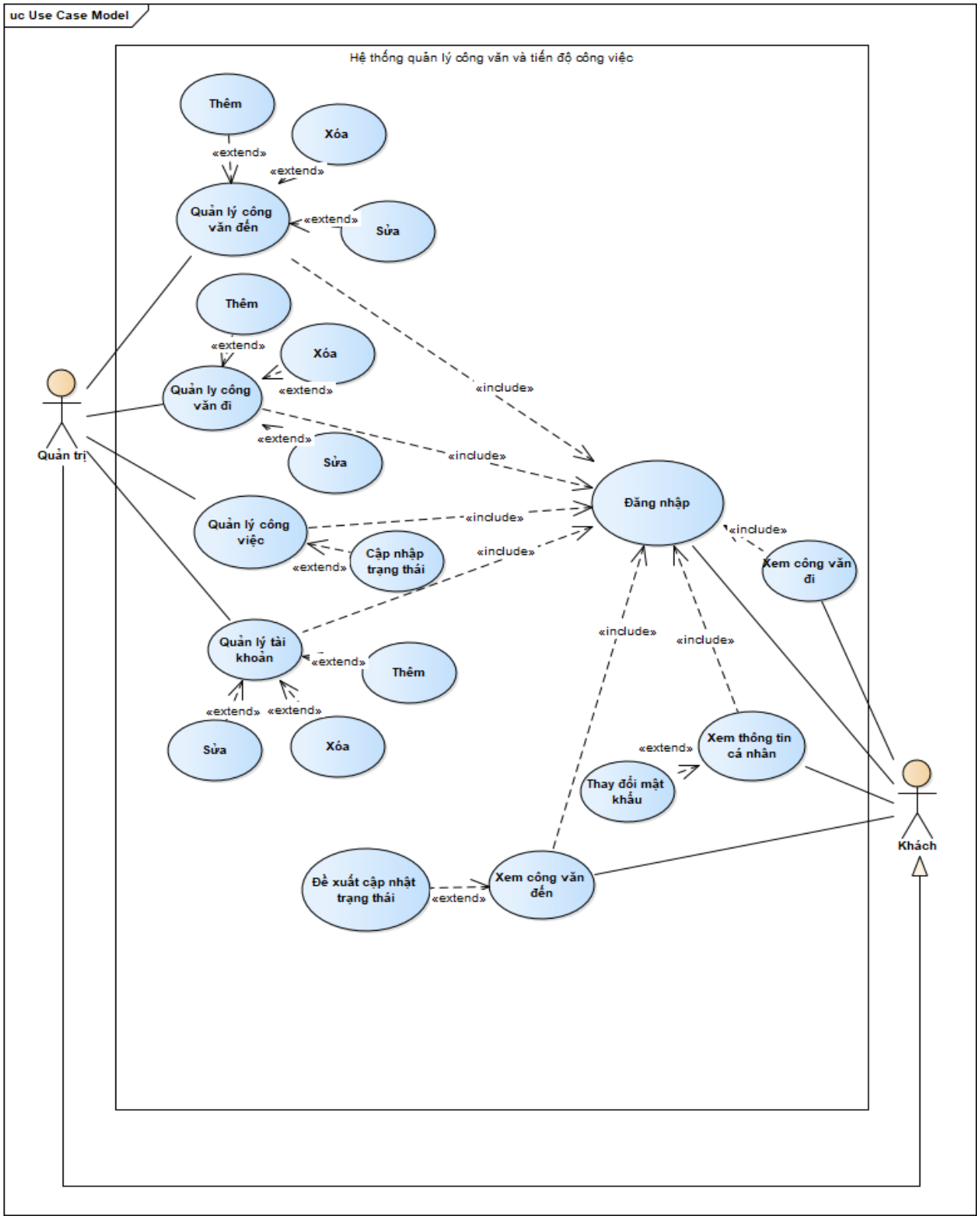
2.6 MERN Stack

MERN là từ viết tắt của MongoDB, Express JS, React JS và Node JS. MERN Stack là sự kết hợp của các công nghệ trên, tất cả dựa trên JavaScript, được sử dụng để xây dựng các ứng dụng web nâng cao. Nó là một full stack development framework mã nguồn mở, tức là nó cung cấp toàn bộ các thành phần phát triển front-end đến back-end.

Hình 2.2 MERN stack

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ

3.1 Use case diagram



Hình 3.1 Use case diagram

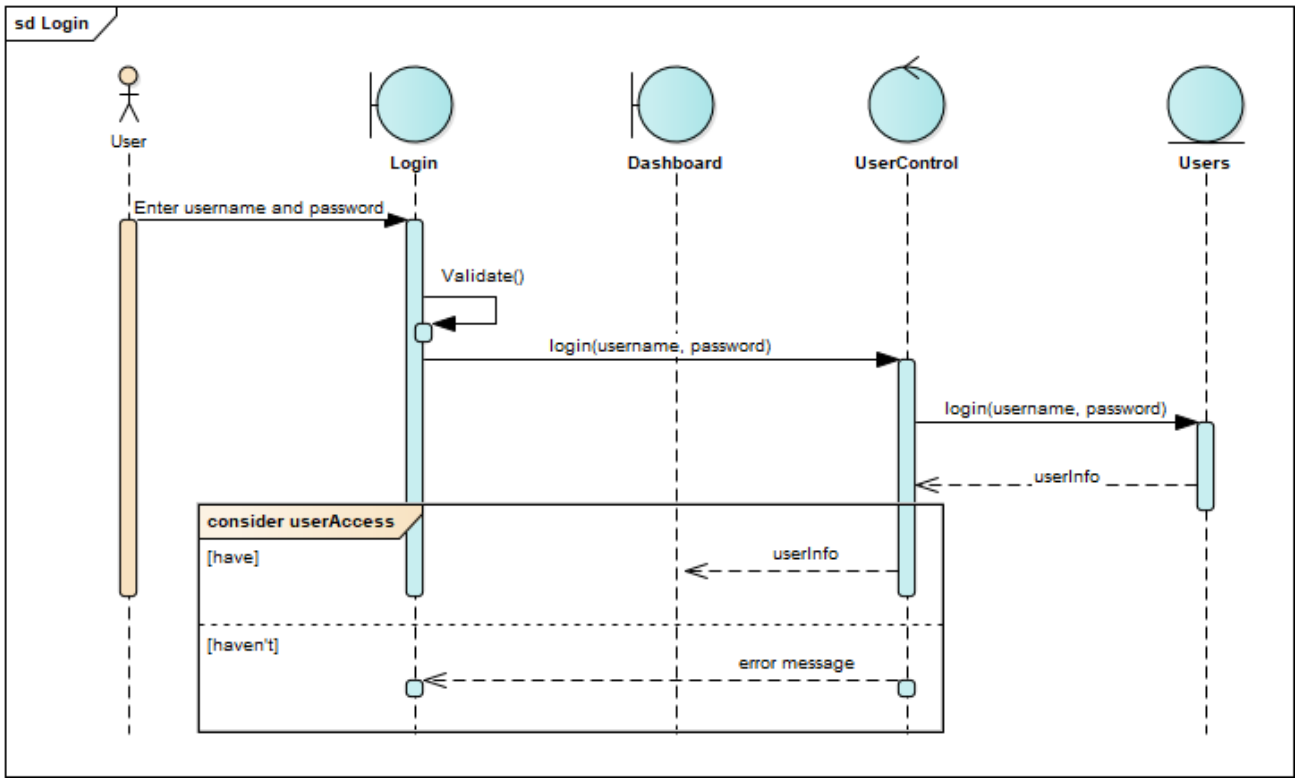
3.2 Đặc tả use case

3.2.1 Đăng nhập

- Đặc tả use case

Tên use case	Đăng nhập	
Tác nhân	Quản trị, khách	
Mô tả	Người dùng sử dụng username, password của mình để đăng nhập.	
Tiền điều kiện		
Luồng sự kiện		
Người dùng		Hệ thống
1. Truy cập hệ thống. 3. Điền username và password, nhấn nút đăng nhập.		2. Chuyển đến trang đăng nhập. 4. Xác minh username, password. 5. Chuyển hướng trang theo quyền truy cập của tài khoản.
Luồng sự kiện phụ		
Người dùng		Hệ thống
		4.1 Username, password không chính xác, hiển thị thông báo.
Hậu điều kiện	Người dùng đăng nhập thành công vào hệ thống để thực hiện các chức năng khác	

- Sequence diagram



Hình 3.1 Login sequence diagram

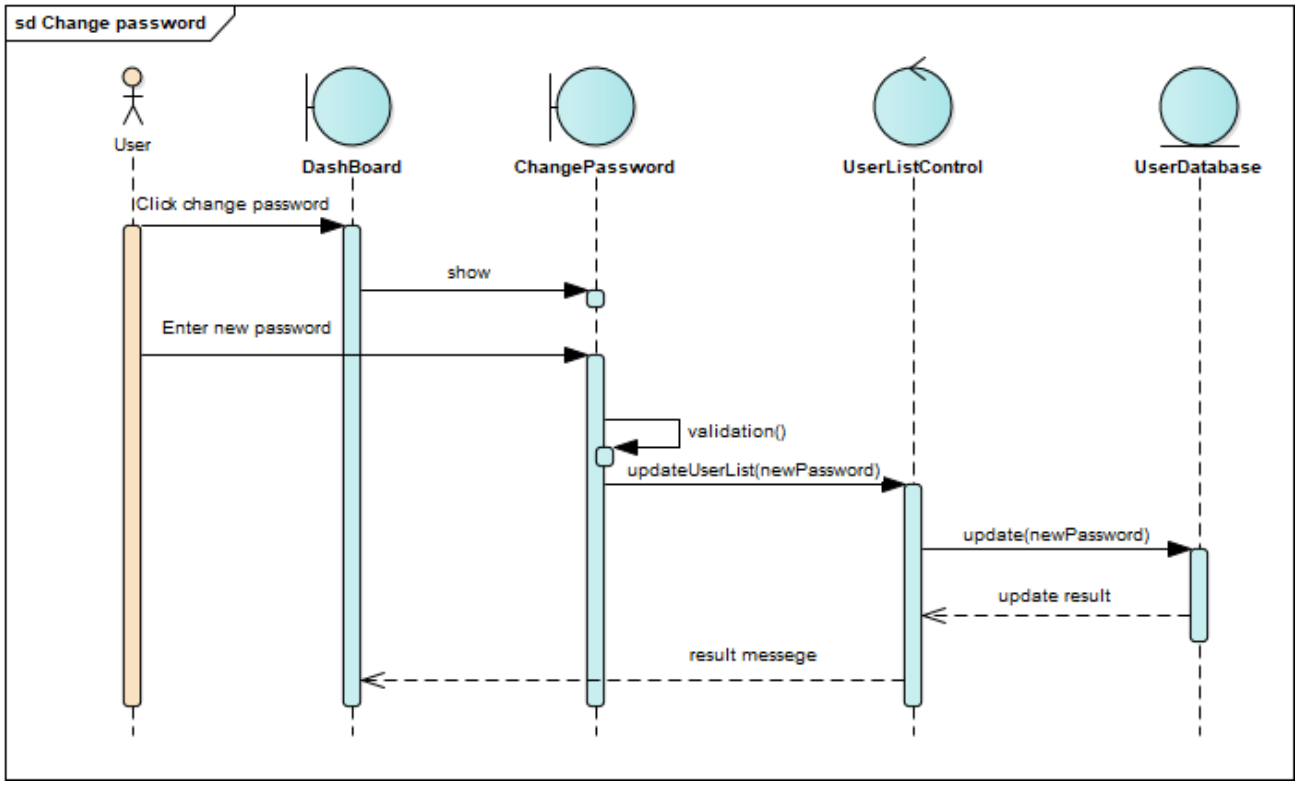
3.2.2 Thay đổi mật khẩu

- Đặc tả use case

Tên use case	Thay đổi mật khẩu
Tác nhân	Quản trị, khách
Mô tả	Khi nhười dùng muốn thay đổi mật khẩu.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Luồng sự kiện	
Người dùng	Hệ thống
1. Chọn xem thông tin cá nhân, chọn thay đổi mật khẩu. 3. Hoàn tất form thay đổi mật khẩu.	2. Hiển thị form thay đổi mật khẩu. 4. Kiểm tra password mới hợp lệ. 5. Thay đổi mật khẩu người dùng bằng mật khẩu mới. 6. Đăng xuất tài khoản hiện tại, chuyển hướng sang trang đăng nhập.
Luồng sự kiện phụ	

Người dùng	Hệ thống
	5.1 Password không hợp lệ. 6.1 Hiển thị thông báo từ chối thay đổi.
Hậu điều kiện	Mật khẩu được thay đổi thành mật khẩu mới.

- Sequence diagram



Hình 3.2 Change password sequence diagram

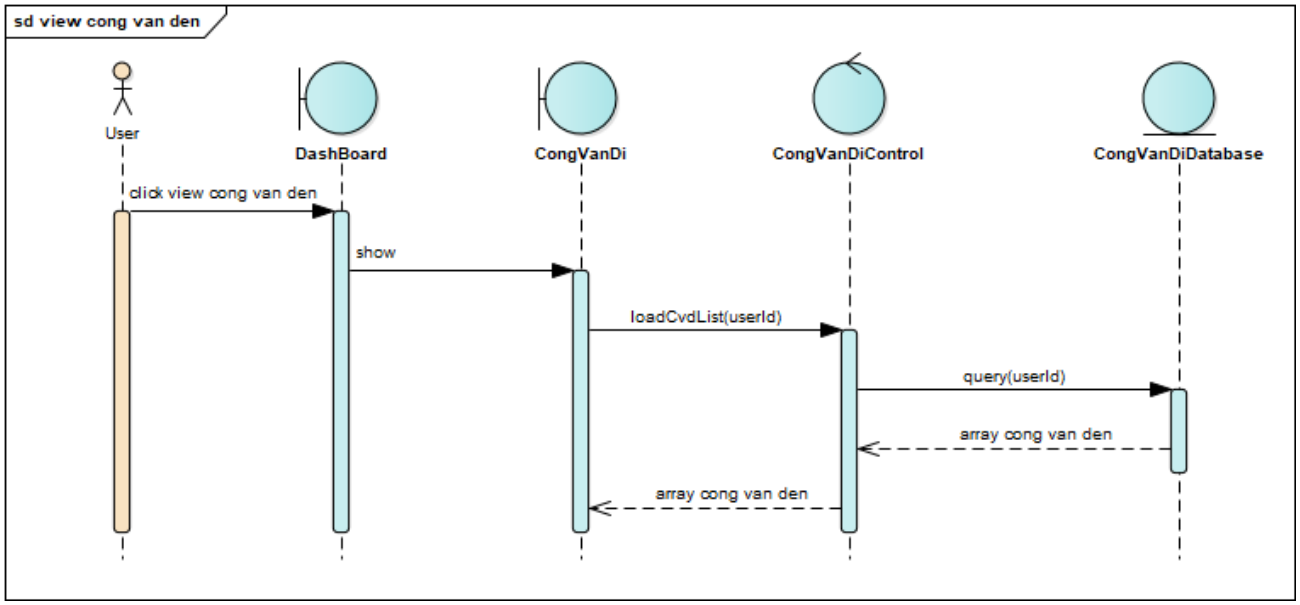
3.2.3 Xem công văn đến

- Đặt tả use case

Tên use case	Xem danh sách công văn đến.
Tác nhân	Quản trị, khách.
Mô tả	Quản trị có thể xem tất cả các công văn đến, khách có thể xem tất cả các công văn đến được giao cho mình.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Luồng sự kiện	
Người dùng	Hệ thống
1. Đăng nhập hệ thống. 2. Chọn công văn đến.	3. Hiển thị danh sách công văn đến theo quyền truy cập.

4. Double click vào công văn đến muốn xem chi tiết.	5. Hiển thị chi tiết công văn đến và file pdf kèm theo.
Luồng sự kiện phụ	
Người dùng	Hệ thống
Hậu điều kiện	Người dùng có thể xem danh sách công văn đến và có thể thao tác tạo, sắp xếp, lọc, xem chi tiết trên danh sách công văn đến.

- Sequence diagram



Hình 3.3 View cong van den sequence diagram

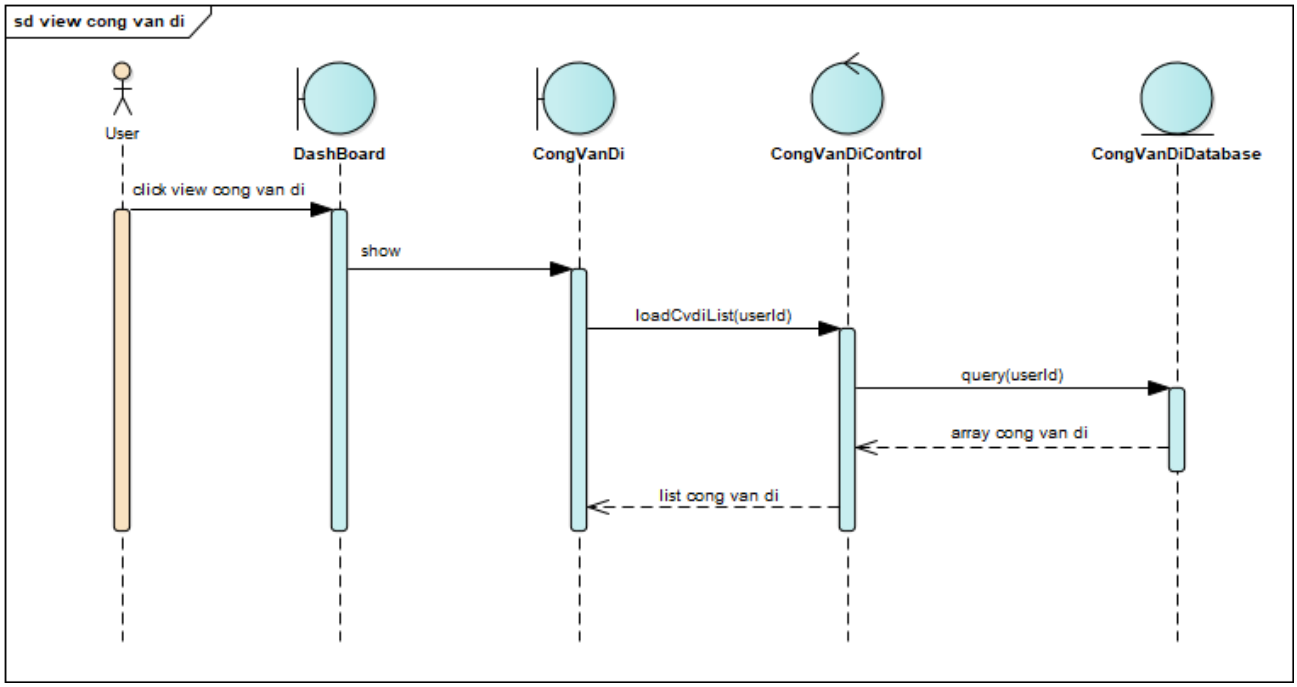
3.2.4 Xem công văn đi

- Đặt tả use case

Tên use case	Xem danh sách công văn đi.
Tác nhân	Quản trị, khách.
Mô tả	Quản trị có thể xem tất cả các công văn đi, khách có thể xem tất cả các công văn đi mà họ được nhận.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Luồng sự kiện	
Người dùng	Hệ thống
1. Đăng nhập hệ thống. 2. Chọn công văn đi.	3. Hiển thị danh sách công văn đi theo quyền truy cập.

4. Double click vào công văn đi muốn xem chi tiết.	5. Hiển thị chi tiết công văn đi và file pdf kèm theo.
Luồng sự kiện phụ	
Người dùng	Hệ thống
Hậu điều kiện	Người dùng có thể xem danh sách công văn đi và có thể thao tác tạo, sắp xếp, lọc, xem chi tiết trên danh sách công văn đi.

- Sequence diagram



Hình 3.4 View cong van di sequence diagram

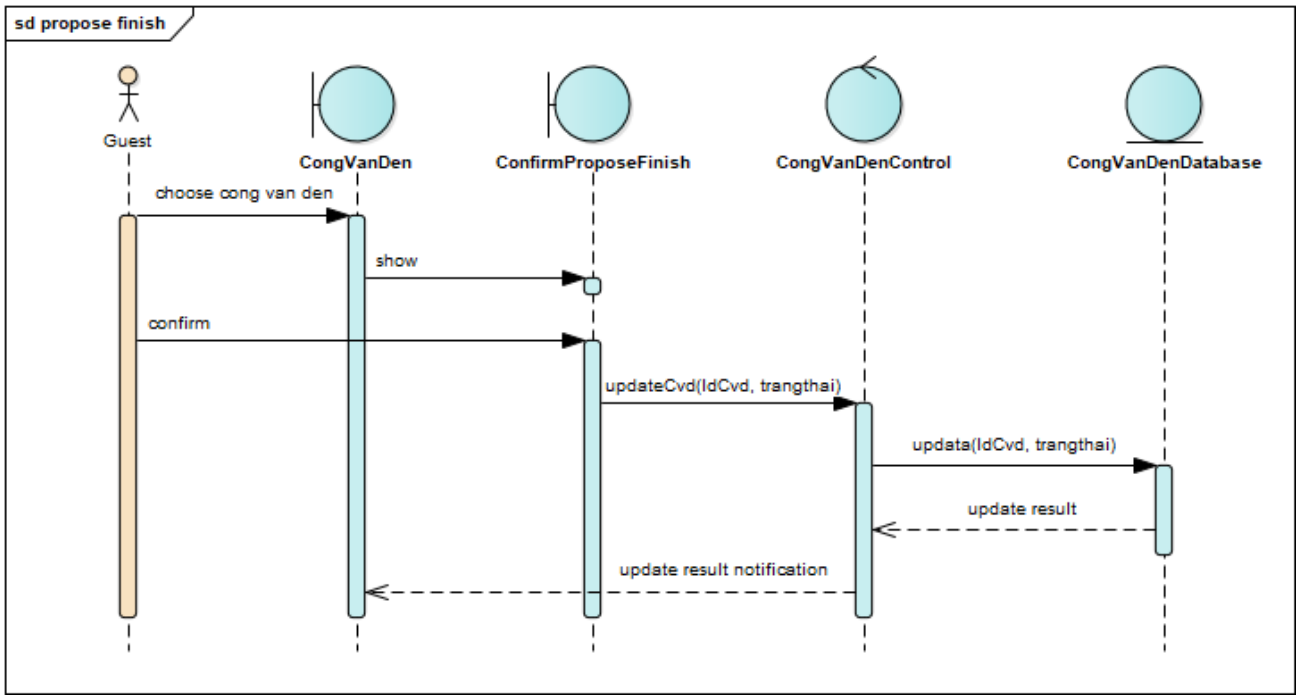
3.2.5 Đề xuất cập nhập trang thái

- Đặc tả use case

Tên use case	Đề xuất hoàn thành.
Tác nhân	Khách.
Mô tả	Sau khi khách hoàn tất công văn được giao người thực hiện sẽ gửi yêu cầu đề xuất cập trang thái đến quản trị viện.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống.
Luồng sự kiện	
Người dùng	Hệ thống
1. Chọn công văn đến.	

3. Chọn đề xuất trang thái hoàn thành ở công văn người thực hiện đã hoàn thành.	2. Hiển thị danh sách công văn đến theo quyền truy cập. 4. Chuyển trạng thái công văn đó sang đề xuất hoàn thành. 5. Gửi thông báo đề xuất hoàn thành cho quản trị viên.
Luồng sự kiện phụ	
Người dùng	Hệ thống
Hậu điều kiện	Trạng thái công văn chuyển thành đề xuất hoàn thành và quản trị viên nhận được thông báo đề xuất hoàn thành.

- Sequence diagram



Hình 3.5 Propose finish sequence diagram

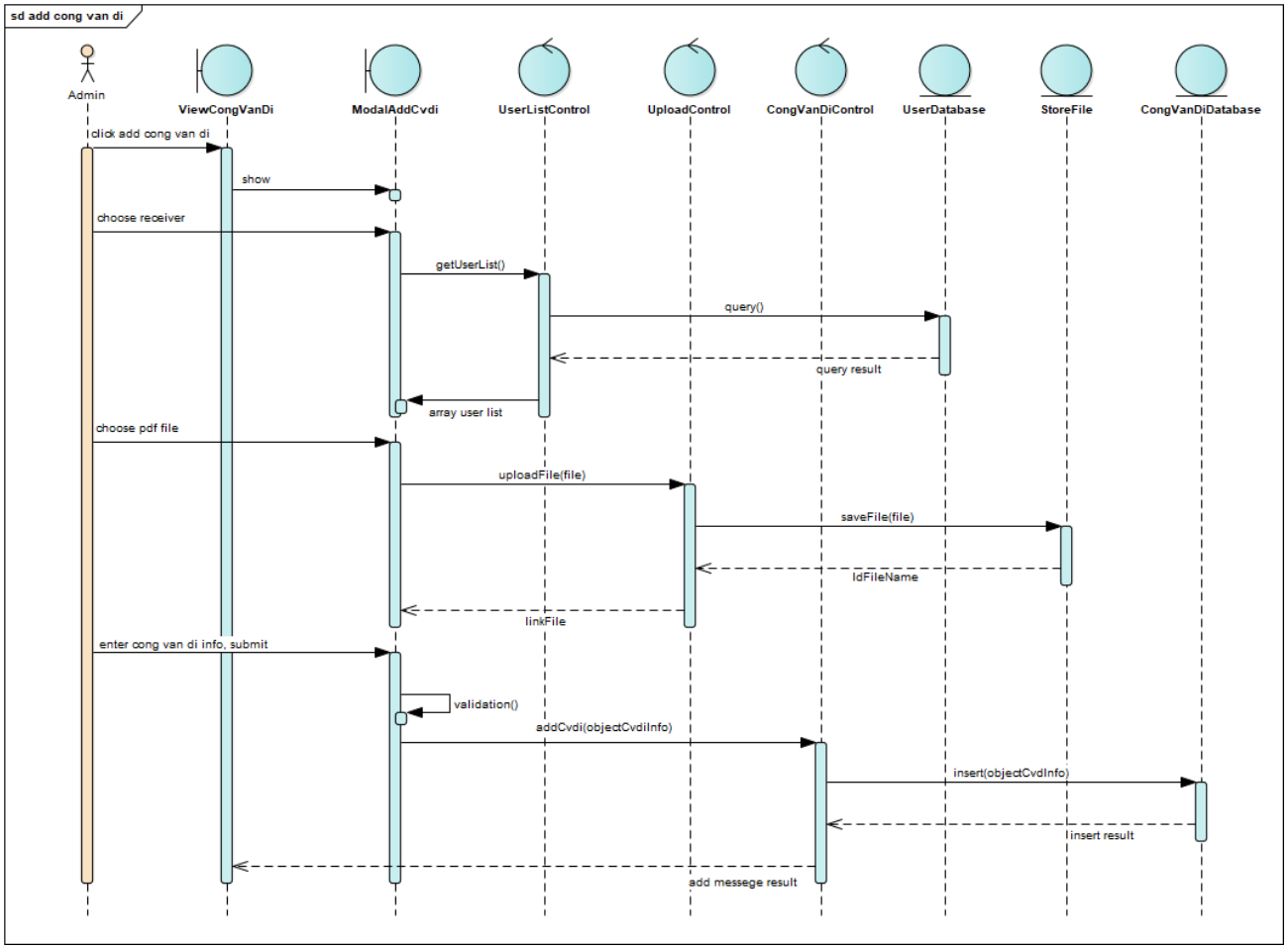
3.2.6 Thêm công văn đi

- Đặc tả use case

Tên use case	Thêm công văn đi
Tác nhân	Quản trị viên.
Mô tả	Cho phép người dùng có thể tạo công văn đi, công văn đi được tạo thành công sẽ chuyển đến người nhận.
Tiền điều kiện	Người dùng đăng nhập tài khoản với quyền quản trị viên.
Luồng sự kiện	

Người dùng	Hệ thống
1. Chọn công văn đi. 2. Chọn thêm công văn đi. 4. Hoàn tất form tạo công văn đi.	 3. Hiển thị form tạo công văn đi. 5. Kiểm tra dữ liệu hợp lệ. 6. Thêm vào danh sách công văn đi, gửi thông báo có công văn đi mới đến người được nhận.
Luồng sự kiện phụ	
Người dùng	Hệ thống
	5.1 Dữ liệu không hợp lệ. Hiển thị thông báo ở form không hợp lệ
Hậu điều kiện	Công văn tạo thành công sẽ thêm vào danh sách công văn đi. Người nhận sẽ nhận được thông báo có công văn đi mới.

- Sequence diagram



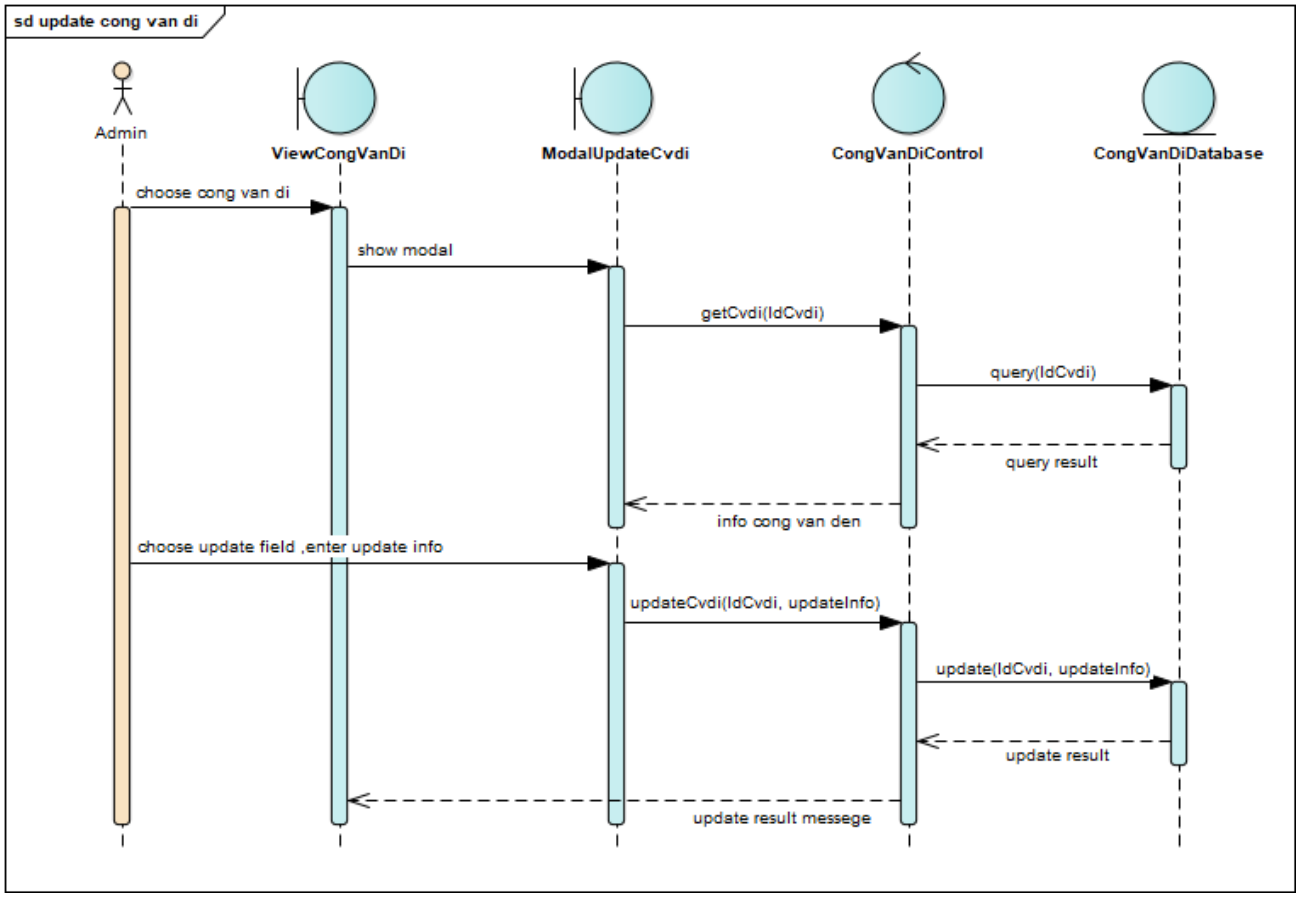
Hình 3.6 Add cong van di sequence diagram

3.2.7 Sửa công văn đi

- Đặc tả use case

Tên use case	Sửa công văn đi	
Tác nhân	Quản trị viên.	
Mô tả	Khi người dùng tạo sai có thể sử dụng sửa công văn đi để cập nhật lại công văn đi đã tạo.	
Tiền điều kiện	Người dùng phải đăng nhập tài khoản có quyền quản trị viên. Công văn cần sửa phải được tạo trước và có trong danh sách công văn đi.	
Luồng sự kiện		
Người dùng		Hệ thống
1. Chọn quản lý công văn đi. 3. Chọn biểu tượng edit ở công văn đi muốn sửa. 5. Chọn field muốn cập nhật thông tin, hoàn tất form.		2. Hiển thị danh sách công văn đi. 4. Hiển thị form cập nhật công văn đi. 6. Kiểm tra thông tin form hợp lệ. 7. Cập nhập danh sách công văn đi của quản trị viện và danh sách công văn đi của người dùng được nhận.
Luồng sự kiện phụ		
Người dùng		Hệ thống
		6.1. Dữ liệu không hợp lệ, hiển thi thông báo dưới form.
Hậu điều kiện	Thông tin công văn được cập nhật lại.	

- Sequence diagram



Hình 3.7 Update cong van di sequence diagram

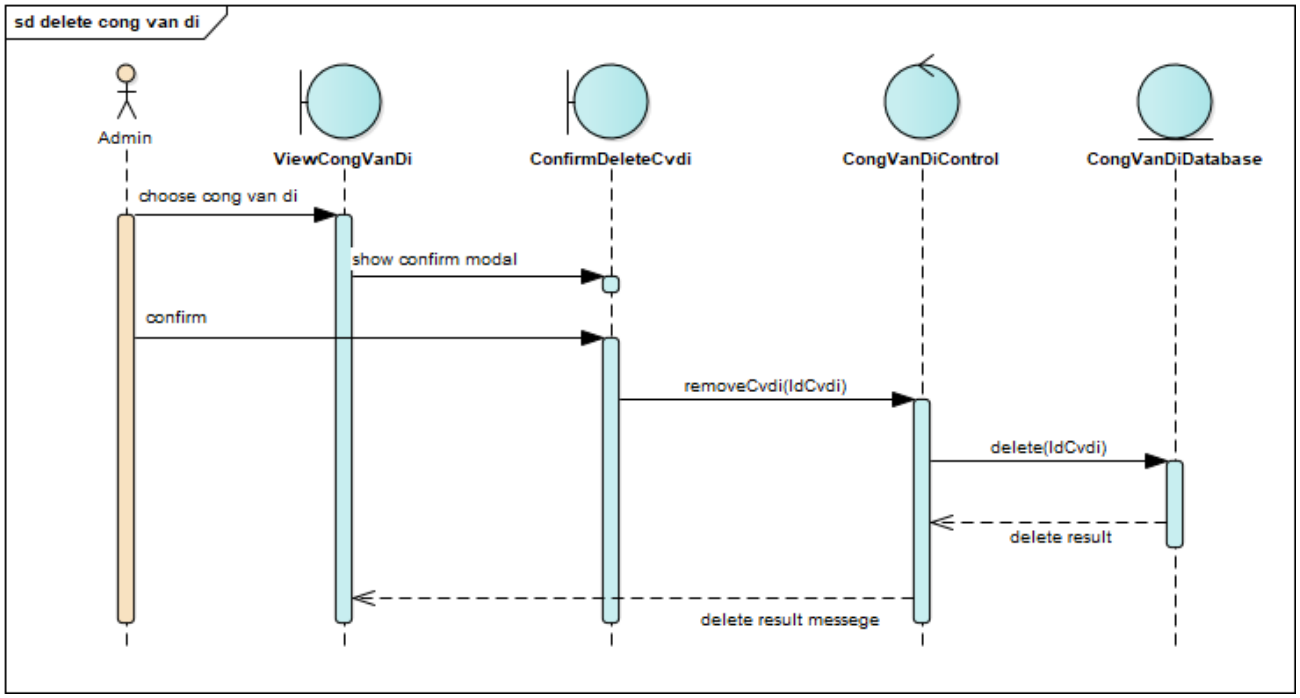
3.2.8 Xóa công văn đi

- Đặc tả use case

Tên use case	Xóa công văn đi
Tác nhân	Quản trị viên.
Mô tả	Khi người dùng tạo sai có thể sử dụng xóa công văn đi để xóa công văn đi ra khỏi danh sách.
Tiền điều kiện	Người dùng phải đăng nhập tài khoản có quyền quản trị viên. Công văn phải được tạo trước và có trong danh sách công văn.
Luồng sự kiện	
Người dùng	Hệ thống
1. Chọn quản lý công văn đi. 3. Chọn biểu tượng xóa ở công văn muốn xóa. 5. Chọn xóa.	2. Hiển thị danh sách công văn đi. 4. Hiển thị thông báo xác nhận xóa công văn. 5. Xóa khỏi danh sách công văn đi của quản trị viện và danh sách công văn đi của người dùng được nhận

Luồng sự kiện phụ	
Người dùng	Hệ thống
5.1 Chọn hủy.	5.1.1 Đóng thông báo xác nhận xóa.
Hậu điều kiện	Công văn được xóa khỏi danh sách công văn đi.

- Sequence diagram



Hình 3.8 Delete cong van di sequence diagram

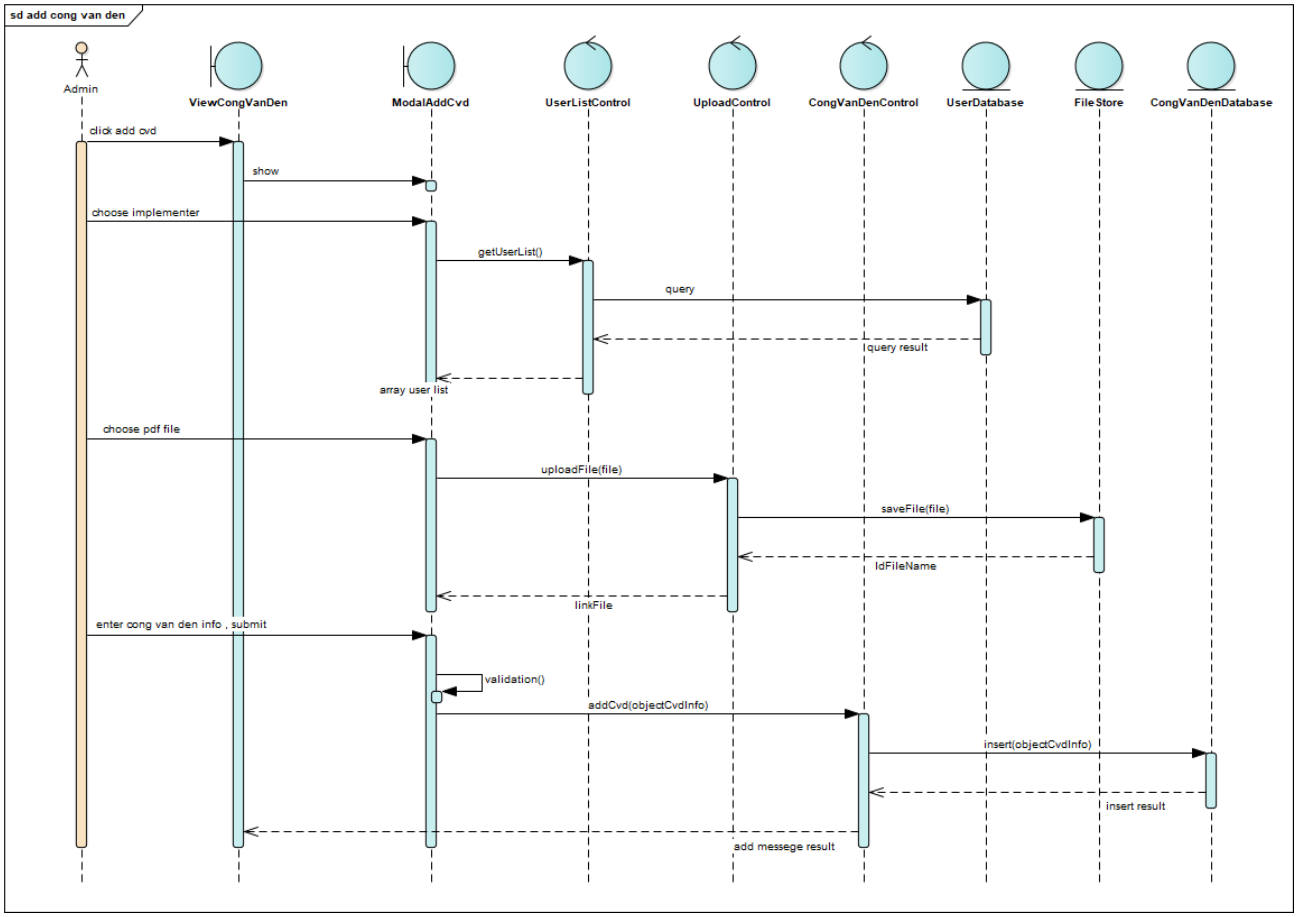
3.2.9 Thêm công văn đến

- Đặc tả use case

Tên use case	Thêm công văn đến
Tác nhân	Quản trị viên.
Mô tả	Cho phép người dùng có thể tạo công văn đến, công văn đến được tạo thành công sẽ chuyển đến người thực hiện.
Tiền điều kiện	Người dùng đăng nhập tài khoản với quyền quản trị viên.
Luồng sự kiện	
Người dùng	Hệ thống
1. Chọn công văn đến. 2. Chọn thêm công văn đến. 4. Hoàn tất form tạo công văn đến.	 3. Hiển thị form tạo công văn đến. 5. Kiểm tra dữ liệu hợp lệ. 6. Công văn đến không thời hạn: thêm vào danh sách công văn đến của quản trị viên và

	người thực hiện, thông báo có công văn đến mới được gửi tới người thực hiện.
Luồng sự kiện phụ	
Người dùng	Hệ thống
	6.1 Công văn đến có thời hạn: thêm vào danh sách công văn đến của quản trị viên, người thực hiện và danh sách quản lý công việc của quản trị, thông báo có công văn đến mới được gửi tới người thực hiện.
Hậu điều kiện	Công văn tạo thành công sẽ thêm vào danh sách công văn đến. Người thực hiện sẽ nhận được thông báo có công văn đến mới.

- Sequence diagram



Hình 3.9 Add cong van den sequence diagram

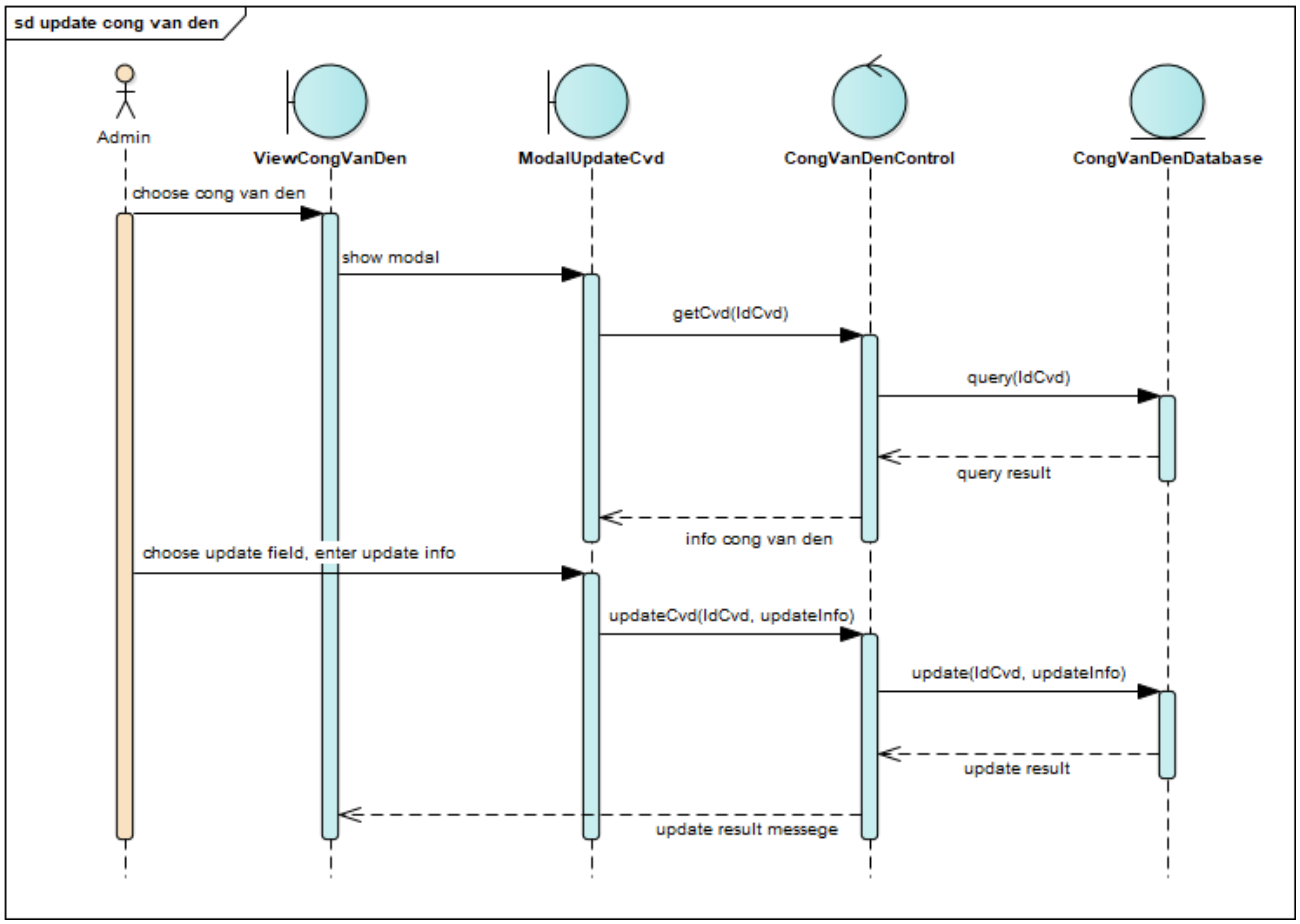
3.2.10 Sửa công văn đến

- Đặc tả use case

Tên use case	Sửa công văn đến
Tác nhân	Quản trị viên.

Mô tả	Khi người dùng tạo sai có thể sử dụng sửa công văn đến để cập nhật lại công văn đến đã tạo.
Tiền điều kiện	Người dùng phải đăng nhập tài khoản có quyền quản trị viên. Công văn cần sửa phải được tạo trước và có trong danh sách công văn đến.
Luồng sự kiện	
Người dùng	Hệ thống
1. Chọn quản lý công văn đến. 3. Chọn biểu tượng edit ở công văn đến muốn sửa. 5. Chọn field muốn cập nhật thông tin, hoàn tất form.	2. Hiển thị danh sách công văn đến. 4. Hiển thị form cập nhật công văn đến. 6. Kiểm tra thông tin form hợp lệ. 7. Công văn đến không thời hạn: cập nhập danh sách công văn đến của quản trị viên và người thực hiện.
Luồng sự kiện phụ	
Người dùng	Hệ thống
	7.1 Công văn đến có thời hạn: cập nhập danh sách công văn đến của quản trị viên, người thực hiện và danh sách quản lý công việc của quản trị.
Hậu điều kiện	Thông tin công văn đến được cập nhật lại.

- Sequence diagram



Hình 3.10 Update cong van den sequence diagram

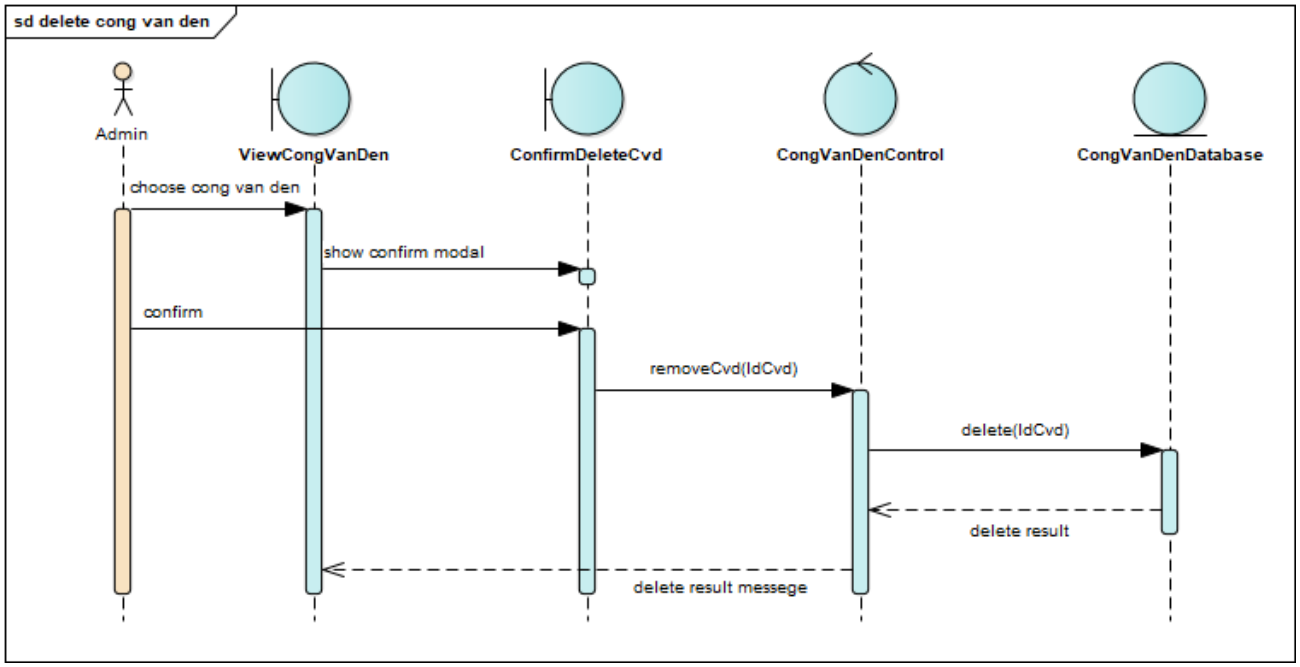
3.2.11 Xóa công văn đến

- Đặc tả use case

Tên use case	Xóa công văn đến
Tác nhân	Quản trị viên.
Mô tả	Khi người dùng tạo sai có thể sử dụng xóa công văn đến để xóa công văn đến ra khỏi danh sách công văn đến.
Tiền điều kiện	Người dùng phải đăng nhập tài khoản có quyền quản trị viên. Công văn phải được tạo trước và có trong danh sách công văn đến.
Luồng sự kiện	
Người dùng	Hệ thống
1. Chọn quản lý công văn đến. 3. Chọn biểu tượng xóa ở công văn muốn xóa. 5. Chọn xóa.	2. Hiển thị danh sách công văn đến. 4. Hiển thị thông báo xác nhận xóa công văn. 6. Công văn đến không thời hạn: xóa khỏi danh sách công văn đến của quản trị viên và người thực hiện.

Luồng sự kiện phụ	
Người dùng	Hệ thống
5.1 Chọn hủy.	5.1.1 Đóng thông báo xác nhận xóa. 6.1 Công văn đến có thời hạn: xóa khỏi danh sách công văn đến của quản trị viên, người thực hiện và danh sách quản lý công việc của quản trị.
Hậu điều kiện	Công văn được xóa khỏi danh sách công văn đến.

- Sequence diagram



Hình 3.11 Delete cong van den sequence diagram

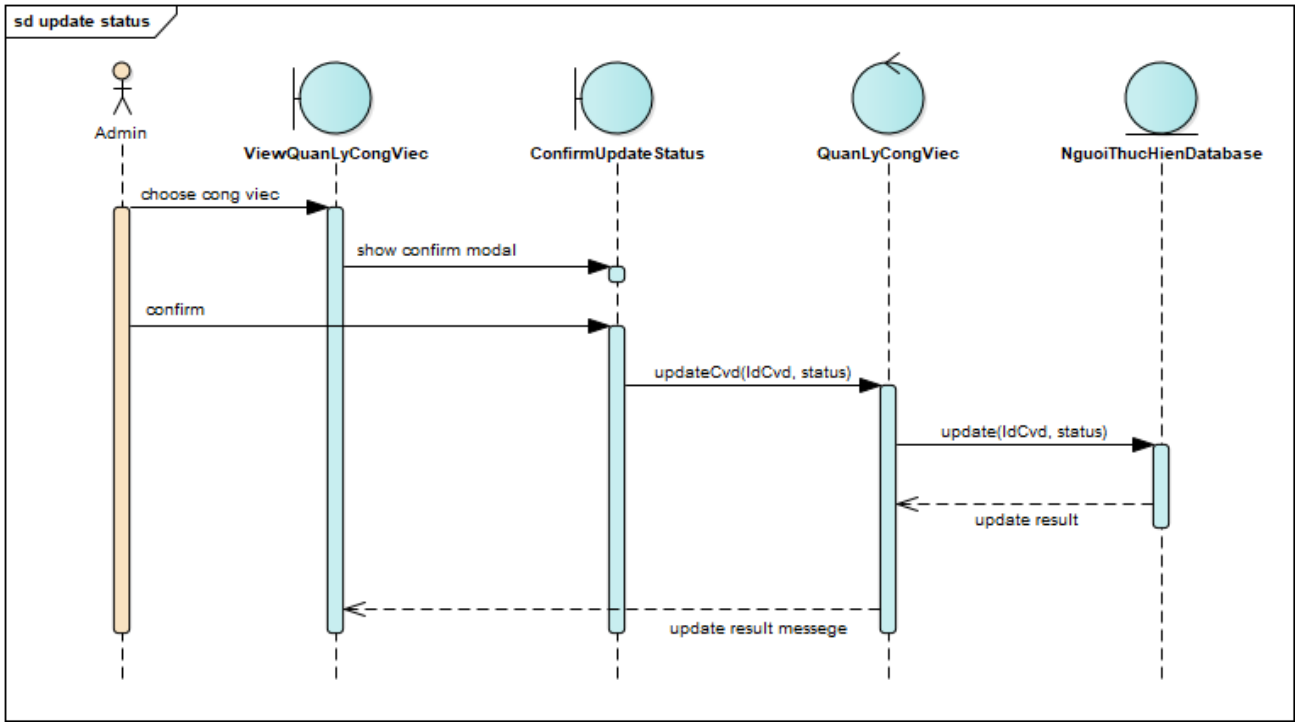
3.2.12 Cập nhập trạng thái

- Đặc tả use case

Tên use case	Cập nhập trạng thái.
Tác nhân	Quản trị viên.
Mô tả	Sau khi nhận được đề xuất cập nhật trạng thái hoàn thành từ người thực hiện, quản trị viên có thể xác nhận công văn đó hoàn thành bằng cách cập nhập trạng thái của công văn.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống với quyền quản trị, công văn đang ở trạng thái đề xuất hoàn thành.
Luồng sự kiện	
Người dùng	Hệ thống
1. Chọn quản lý công việc.	

3. Chọn xác nhận hoàn thành tại công văn muốn cập nhập trạng thái. 5. Chọn đồng ý.	2. Hiển thị danh sách công văn đến có thời hạn kèm trạng thái của công văn. 4. Hiển thị thông báo xác nhận hoàn thành. 6. Cập nhật trạng thái hoàn thành ở danh sách công việc và danh sách công văn đến. 7. Gửi thông báo công văn hoàn thành đến người thực hiện công văn.
Luồng sự kiện phụ	
Người dùng	Hệ thống
5.1 Chọn không đồng ý	5.1.1 Cập nhật trạng thái từ chối đề xuất ở danh sách công việc và danh sách công văn đến. 5.1.2 Gửi thông báo từ chối đề xuất đến người thực hiện công văn.
Hậu điều kiện	Trạng thái của công văn được cập nhập và người tạo công văn thực hiện được thông báo trạng thái được cập nhập.

- Sequence diagram



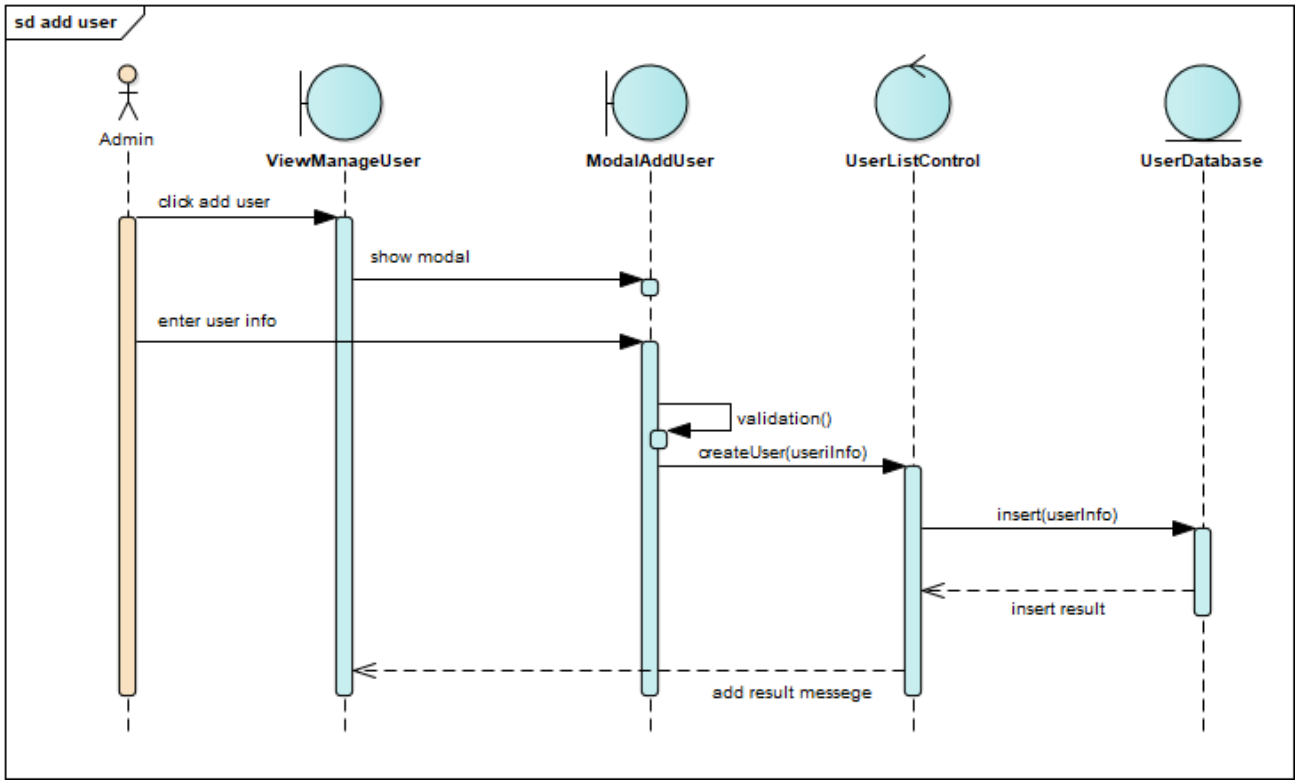
Hình 3.12 Update status sequence diagram

3.2.13 Thêm người dùng

- Đặc tả use case

Tên use case	Thêm người dùng	
Tác nhân	Quản trị.	
Mô tả	Quản trị có thể tạo tài khoản người dùng mới.	
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống với quyền quản trị.	
Luồng sự kiện		
Người dùng		Hệ thống
1. Chọn quản lý người dùng. 3. Chọn thêm người dùng. 5. Hoàn tất form thêm người dùng.		2. Hiển thị danh sách người dùng. 4. Hiển thị form thêm người dùng. 6. Kiểm tra thông tin form hợp lệ. 7. Thêm tài khoản mới tạo vào danh sách người dùng.
Luồng sự kiện phụ		
Người dùng		Hệ thống
		6.1 Thông tin form không hợp lệ. Hiển thị thông báo không hợp lệ dưới form.
Hậu điều kiện	Người dùng có đăng nhập bằng tài khoản mới.	

- Sequence diagram



Hình 3.13 Add user sequence diagram

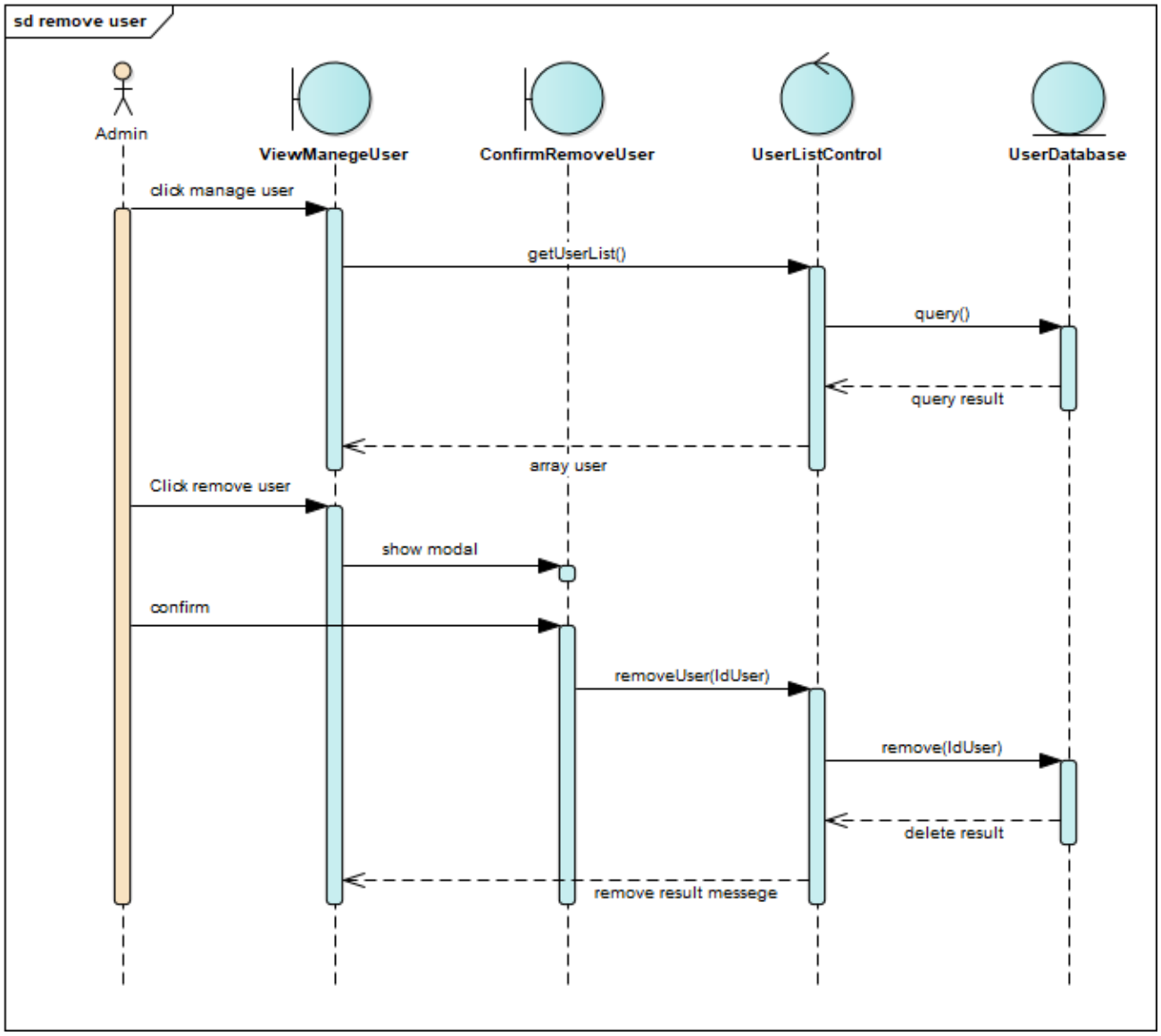
3.2.14 Xóa người dùng

- Đặc tả use case

Tên use case	Xóa người dùng
Tác nhân	Quản trị.
Mô tả	Quản trị có thể xóa tài khoản người dùng.
Tiền điều kiện	Người dùng phải đăng nhập vào hệ thống với quyền quản trị.
Luồng sự kiện	
Người dùng	Hệ thống
1. Chọn quản lý người dùng. 3. Chọn biểu tượng thùng rác ở người dùng muốn xóa. 5. Chọn đồng ý.	2. Hiển thị danh sách người dùng. 4. Hiển thị thông báo xác nhận xóa. 6. Xóa tài khoảng khỏi danh sách người dùng.
Luồng sự kiện phụ	
Người dùng	Hệ thống
5.1 Chọn hủy.	5.1.1 Đóng thông báo xác nhận xóa.

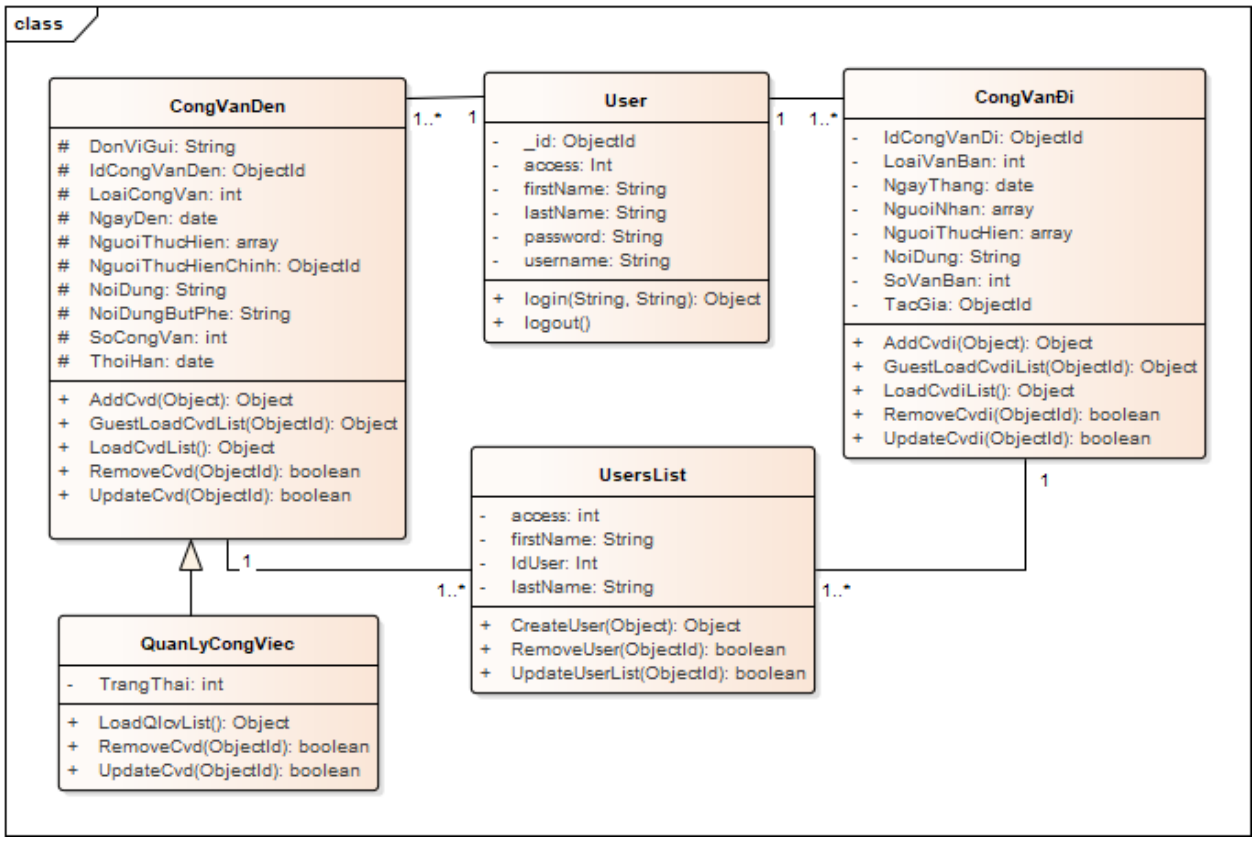
Hậu điều kiện	Tài khoản xóa khỏi danh sách người dùng và không thể sử dụng để đăng nhập.

- Sequence diagram



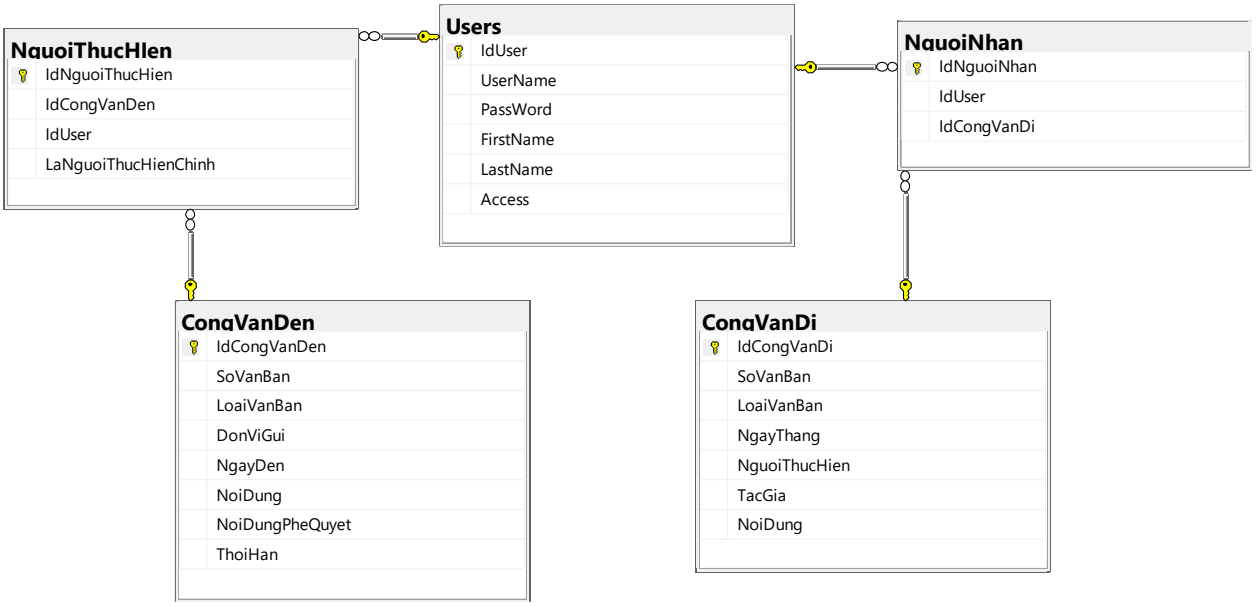
Hình 3.14 Remove user sequence diagram

3.3 Class diagram:



Hình 3.15 Class diagram

3.4 Data model



Hình 3.16 Data model diagram

CHƯƠNG 4. THIẾT KẾ VÀ THỰC HIỆN

4.1 Xây dựng ứng dụng trên nền tảng web

4.2 Kiểm thử phần mềm

CHƯƠNG 5. KẾT LUẬN

5.1 Kết quả đạt được

5.2 Hạn chế của đồ án

5.3 Hướng phát triển

TÀI LIỆU THAM KHẢO

[1] RESTful API (Book: "REST API Development with Node.js" author: Fernando Doglio, https://topdev.vn/blog/restful-api-la-gi/)

[2] Nodejs (Book: "Web Development with Node and Express: Leveraging the JavaScript Stack" author: Ethan Brown, https://freetuts.net/nodejs-la-gi-584.html)

[3] MongoDB (Book: "Beginning Node.js, Express & MongoDB Development" author: Greg Lim, https://topdev.vn/blog/mongodb-la-gi-co-so-du-lieu-phi-quan-he/)

[4] Reactjs (Book: "Learning React: Functional Web Development with React and Redux" author: Alex Banks, Eve Porcello, https://viblo.asia/p/reactjs-uu-diem-va-nhuoc-diem-V3m5WzexlO7)

[5] MERN stack (Book: "MERN Quick Start Guide" author: Eddy Wilson Iriarte Koroliova, https://viblo.asia/p/gioi-thieu-mern-stack-bWrZnv4vZxw)