

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

MAI VĂN TUẤN – 1612781

CHÂU XUÂN TUẤN – 1712868

NGUYỄN THỌ TUẤN – 1712878

LÊ VĂN – 1712897

HOÀNG MINH VŨ - 1712918

**XÂY DỰNG HỆ THỐNG KÝ KẾT VĂN BẢN
TRỰC TUYẾN**

**THỰC TẬP DỰ ÁN TỐT NGHIỆP CỦ NHÂN CNTT
CHƯƠNG TRÌNH CHÍNH QUY**

Tp. Hồ Chí Minh, tháng 03/2022

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

MAI VĂN TUẤN – 1612781
CHÂU XUÂN TUẤN – 1712868
NGUYỄN THỌ TUẤN – 1712878
LÊ VĂN – 1712897
HOÀNG MINH VŨ - 1712918

**XÂY DỰNG HỆ THỐNG KÝ KẾT VĂN BẢN
TRỰC TUYẾN**

THỰC TẬP DỰ ÁN TỐT NGHIỆP CỦ NHÂN CNTT
CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN HƯỚNG DẪN
TS. NGÔ HUY BIÊN

Tp. Hồ Chí Minh, tháng 03/2022

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Tp. HCM, ngày ... tháng ... năm 2021
Giáo viên hướng dẫn

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Tp. HCM, ngày ... tháng ... năm 2021
Giáo viên phản biện

LỜI CẢM ƠN

Để hoàn thành Thực tập dự án tốt nghiệp này, đầu tiên chúng tôi xin gửi lời cảm ơn đến Khoa Công nghệ Thông tin - Trường Đại học Khoa học Tự nhiên, xin cảm ơn ban quản trị của nhà trường và toàn thể quý Thầy Cô giảng dạy. Các bài giảng của thầy cô trang bị cho chúng tôi những kiến thức, nền tảng vững chắc và quý báu trong những năm học vừa qua.

Đặc biệt, chúng tôi xin chân thành gửi lời cảm ơn đến thầy Ngô Huy Biên, thầy đã trực tiếp hướng dẫn, tận tình giải đáp thắc mắc, góp ý về nội dung, tạo điều kiện thoải mái và đưa ra những định hướng trong quá trình thực hiện Thực tập dự án tốt nghiệp.

Trong quá trình thực hiện Thực tập dự án tốt nghiệp, do những hạn chế về kinh nghiệm nên khó có thể tránh khỏi những thiếu sót và hạn chế. Vì vậy, chúng tôi hi vọng thầy cô và người đọc sẽ đưa ra những phản hồi, những đóng góp giúp đỡ tài dự án trở nên hoàn thiện hơn.

Cuối cùng, chúng tôi xin kính gửi đến thầy cô một lời cảm ơn sâu sắc, chúc các thầy cô sẽ gặp nhiều thành công trong cuộc sống.

Thành phố Hồ Chí Minh, ngày tháng 03 năm 2022

Nhóm sinh viên thực hiện

Mai Văn Tuấn

Châu Xuân Tuấn

Nguyễn Thọ Tuấn

Lê Văn

Hoàng Minh Vũ



fit@hcmus

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

XÂY DỰNG HỆ THỐNG KÝ KẾT VĂN BẢN TRỰC TUYẾN

(*BUILDING E-SIGNATURE SYSTEM*)

1 THÔNG TIN CHUNG

Người hướng dẫn:

TS. Ngô Huy Biên (Khoa Công nghệ Thông tin)

Nhóm sinh viên thực hiện:

1. Mai Văn Tuấn (MSSV: 1612781)
2. Châu Xuân Tuấn (MSSV: 1712868)
3. Nguyễn Thọ Tuấn (MSSV: 1712878)
4. Lê Văn (MSSV: 1712897)
5. Hoàng Minh Vũ (MSSV: 1712918)

Loại đề tài: Ứng dụng

Thời gian thực hiện: Từ 09/2021 đến 03/2022

2 NỘI DUNG THỰC HIỆN

2.1 Giới thiệu về đề tài

Hiện nay, khi mà mọi vấn đề trong cuộc sống đều có thể giải quyết bằng ứng dụng công nghệ và số hóa thông tin. Chữ ký điện tử được sử dụng rộng rãi và nhiều người biết đến, cụ thể trong các giao dịch điện tử. Các ứng dụng hiện có trên thị trường rất đa dạng, với nhiều tính năng kèm theo và có thể mang nặng tính chất kinh doanh quảng cáo. Bên cạnh đó, chức năng và nhu cầu của người sử dụng không nhiều, các hệ thống chữ ký điện tử phổ biến hầu hết tập trung vào thị trường Mỹ và nước ngoài. Đến đến bất cập không hỗ trợ ngôn ngữ tiếng Việt, nhiều chức năng thừa và không được sử dụng tại quốc gia Việt Nam, yêu cầu thẻ thanh toán trực tuyến quốc tế khi đăng ký, bản quyền giá cao khi quy đổi ngoại tệ tiền Việt Nam. Vì thế chúng tôi muốn tạo ra một hệ thống ký kết văn bản trực tuyến, tập trung thị trường trong nước, ưu tiên sự tối giản, nhanh gọn. Tên là VTSign – Hệ thống ký kết văn bản trực tuyến (Building e-signature system).

2.2 Mục tiêu đề tài

Thiết kế, xây dựng, kiểm thử, và triển khai hệ thống.

- Back-end: Cung cấp các API bao gồm: cho phép người dùng đăng ký, và thanh toán chi phí sử dụng dịch vụ, cho phép người quản trị quản lý người sử dụng dịch vụ, xử lý việc quản lý và ký kết văn bản, cung cấp các báo cáo cho admin về quá trình truy cập và báo cáo cho người dùng về quá trình truy cập, dung lượng sử dụng của từng người dùng sử dụng, dung lượng sử dụng của tài khoản. Áp dụng Microservices, mô phỏng, thiết kế, thực hiện, báo cáo kết quả các kịch bản kiểm thử khả năng tải mong muốn (load tests), khả năng chịu tải tối đa (stress tests), khả năng xâm nhập hệ thống (penetration testing).
- Front-end: Các giao diện kết nối với các tính năng tương ứng với back-end, đơn giản dễ dàng thao tác và phù hợp với nhiều đối tượng người dùng sử dụng.

Viết 80 trang Đồ án.

2.3 Phạm vi của đề tài

Các tính năng không thực hiện

- Các chức năng tự động hóa được thiết lập riêng. Thanh toán trực tuyến thông qua ngân hàng. Bản quyền và giới hạn sử dụng.
- Đồng bộ vào trên các phần mềm bên thứ ba ví dụ như Dropbox, Word, Adobe. Triển khai trên tất cả nền tảng khác.
- Chức năng phức tạp chuyên môn cao như công chứng và chống giả mạo, mã hóa tài liệu và bảo mật cao. Chức năng kiểm toán thống kê đánh giá đặc thù.
- Giao diện có nhiều tùy chọn thay đổi như quốc gia và vùng, ngôn ngữ. Tùy chọn ẩn hay hiện mục và các chức năng trên hệ thống.
- Thêm các thương hiệu logo được cá nhân hóa. Xác định các loại chữ ký được phép. Hệ thống live chat trò chuyện trực tiếp trên hệ thống.

2.4 Cách tiếp cận dự kiến

***Bản mẫu:** demo các trang chức năng chính của website.

The screenshot shows the VTSign software interface. At the top, there is a blue header bar with the VTSign logo, navigation links for 'Trang chủ' (Home), 'Kí kết' (Sign), 'Quản lý' (Management), 'Bản mẫu' (Template), and language settings ('Vi'). Below the header, the main window has a left sidebar with options: 'Hủy' (Cancel), 'Thêm tài liệu' (selected), 'Chọn người nhận và cài đặt' (Select recipient and set up), 'Ký tên và các thông tin khác' (Sign name and other information), and 'Kiểm tra và gửi file' (Check and send file). The central area is titled 'Thêm tài liệu' and features a large dashed rectangular area for file upload, a blue cloud icon with an upward arrow, and the text 'Tải tài liệu'. To the right, there is a vertical panel with a document icon and the text 'Chưa có tài liệu' (No documents). At the bottom, there are two buttons: 'Chỉ mình tôi ký' (I sign myself) and a blue 'Tiếp tục' (Continue) button. A small note at the bottom states: 'Bản quyền © 2021 VTSign, Inc. Bảo lưu mọi quyền.'

VTSign

- [Trang chủ](#)
- [Kí kết](#)
- [Quản lý](#)
- [Bản mẫu](#)

VI EN A ▾

X Hủy

Thêm tài liệu

Chọn người nhận và cài đặt

Ký tên và các thông tin khác

Kiểm tra và gửi file

Ký tên và các thông tin khác

Tôi

Ký tên

Ngày ký

Email

Họ Tên

Công ty

Vẽ

Chữ

ĐIỀU 8 : CAM KẾT CHUNG

- Hai bên cam kết thực hiện nghiêm túc và đầy đủ các điều khoản và điều kiện quy định trong hợp đồng này. Mọi thay đổi, hủy bỏ hoặc bổ sung một hay nhiều điều khoản, điều kiện của hợp đồng này phải được cả 2 bên thỏa thuận bằng văn bản và lập thành phụ lục hợp đồng.
- Trường hợp các cơ quan có thẩm quyền của Việt Nam ban hành các văn bản pháp lý liên quan đến việc cho thuê Nhà, Hợp đồng này sẽ được điều chỉnh cho phù hợp với những quy định của Pháp luật Việt Nam.
- Tranh chấp phát sinh liên quan đến hợp đồng này hoặc việc vi phạm hợp đồng sẽ được giải quyết trước hết bằng thương lượng trên tinh thần thiện chí, hợp tác. Nếu thương lượng không thành thì vụ việc sẽ được đưa ra to án có thẩm quyền giải quyết. Quyết định của to án là chung và có hiệu lực cưỡng chế thi hành với các bên có liên quan. Bên thuê phải chịu toàn bộ án phí và các chi phí khác (nếu có), trừ khi có thỏa thuận khác.
- Hợp đồng này có hiệu lực pháp lý từ ngày ký đến hết ngày:
Hợp đồng này được lập thành 03 bản và có giá trị pháp lý như nhau, mỗi bản giữ 01 (một) bản.

ĐẠI DIỆN BÊN CHO THUÊ / BÊN A

Nguyễn Văn A

ĐẠI DIỆN BÊN THUÊ / BÊN B

Ký tên

Trở lại Tiếp tục

Bản quyền © 2021 VTSign, Inc. Bảo lưu mọi quyền.

VTSign

X Hủy

ĐIỀU 8 : CAM KẾT CHUNG

- Hai bên cam kết thực hiện nghiêm túc và đầy đủ các điều khoản và điều kiện quy định trong hợp đồng này. Mọi thay đổi, hủy bỏ hoặc bổ sung một hay nhiều điều khoản, điều kiện của hợp đồng này phải được cả 2 bên thỏa thuận bằng văn bản và lập thành phụ lục hợp đồng.
- Trường hợp các cơ quan có thẩm quyền của Việt Nam ban hành các văn bản pháp lý liên quan đến việc cho thuê Nhà, Hợp đồng này sẽ được điều chỉnh cho phù hợp với những quy định của Pháp luật Việt Nam.
- Tranh chấp phát sinh liên quan đến hợp đồng này hoặc việc vi phạm hợp đồng sẽ được giải quyết trước hết bằng thương lượng trên tinh thần thiện chí, hợp tác. Nếu thương lượng không thành thì vụ việc sẽ được đưa ra to án có thẩm quyền giải quyết. Quyết định của to án là chung và có hiệu lực cưỡng chế thi hành với các bên có liên quan. Bên thuê phải chịu toàn bộ án phí và các chi phí khác (nếu có), trừ khi có thỏa thuận khác.
- Hợp đồng này có hiệu lực pháp lý từ ngày ký đến hết ngày:
Hợp đồng này được lập thành 03 bản và có giá trị pháp lý như nhau, mỗi bản giữ 01 (một) bản.

ĐẠI DIỆN BÊN CHO THUÊ / BÊN A

Nguyễn Văn A

ĐẠI DIỆN BÊN THUÊ / BÊN B

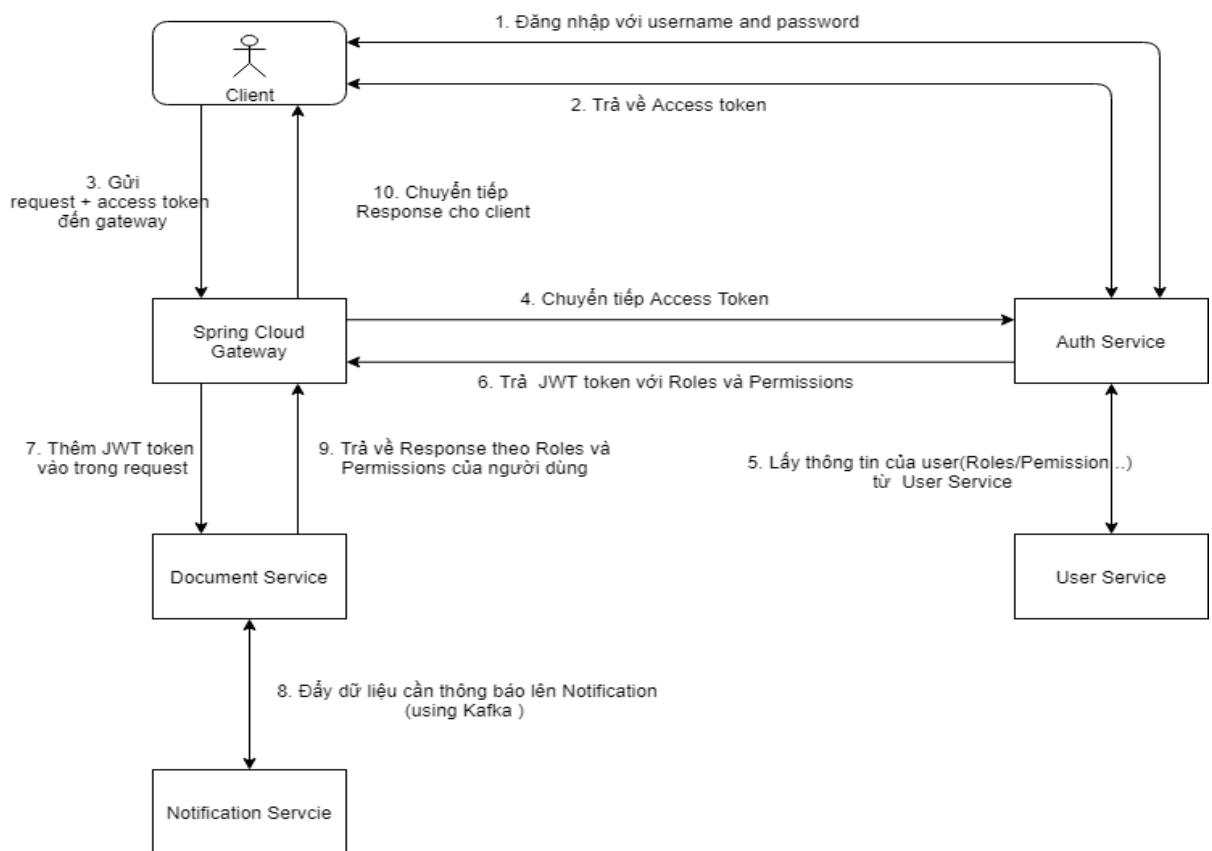
Ký tên

Kết thúc

Bản quyền © 2021 VTSign, Inc. Bảo lưu mọi quyền.

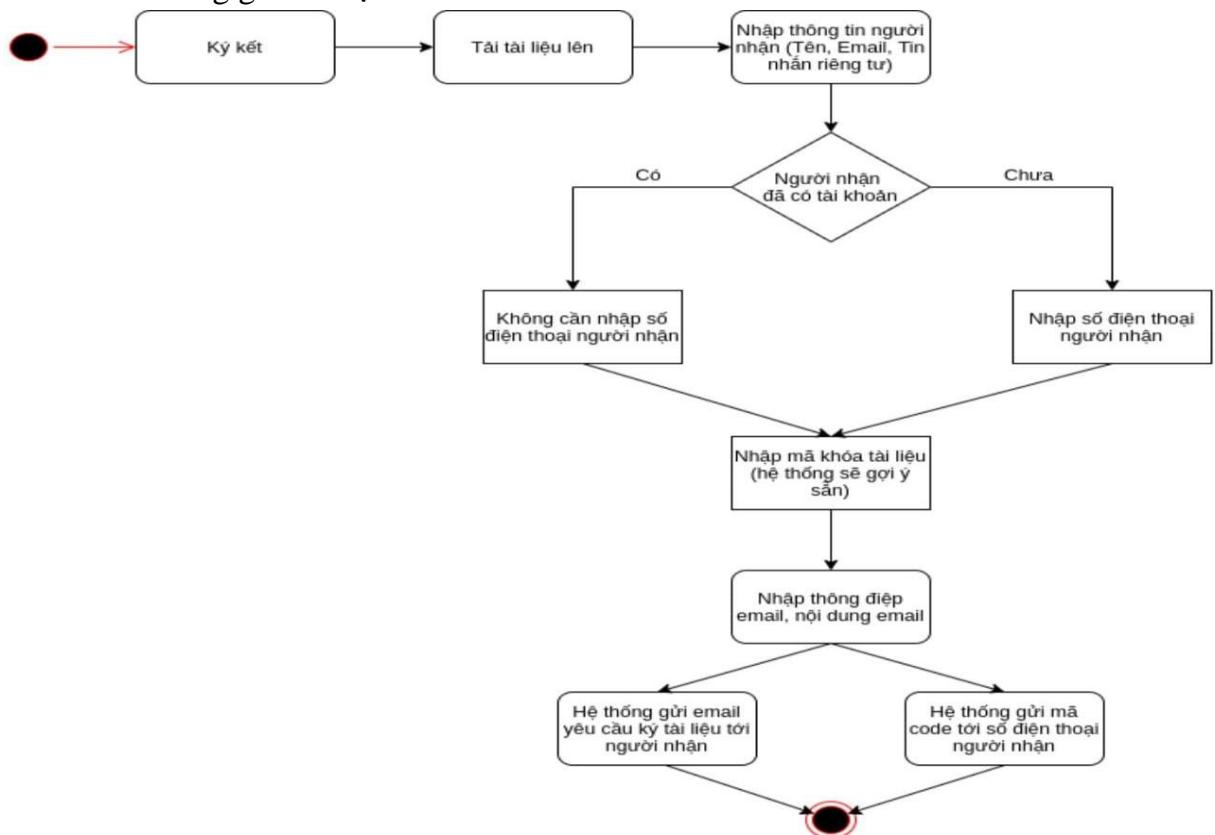
*Kiến trúc:

Kiến trúc Process:

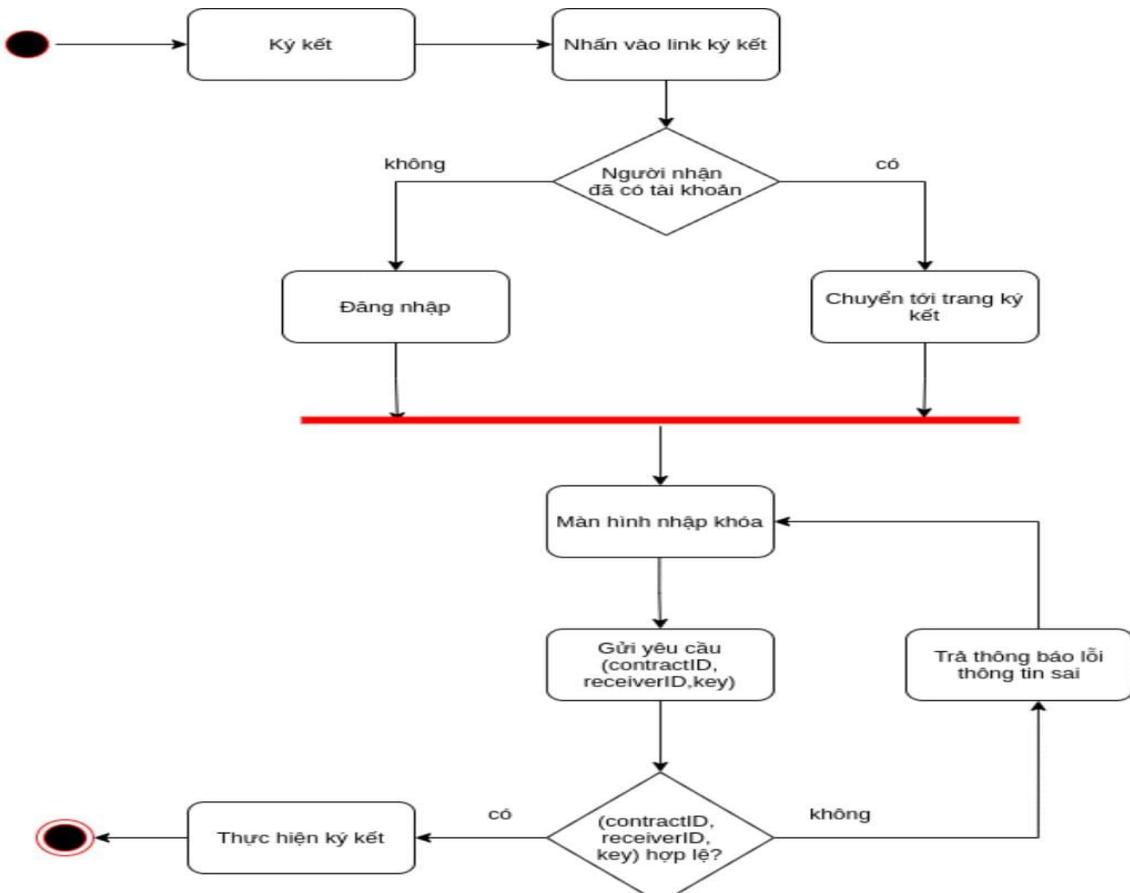


- Hình trên thể hiện quy trình khi thực hiện yêu cầu thì yêu cầu sẽ đi qua các service. Yêu cầu được gửi lên từ user (đã có tài khoản) thông qua phải có AccessToken. Sau đó Spring Cloud Gateway sẽ gửi AccessToken lên AuthService và lấy roles và permissions. Tạo ra Access Token mới và trả về cho gateway gắn vào request và tiếp tục đến service đích. Người chưa có tài khoản cần đăng ký, đăng nhập để có Access Token.

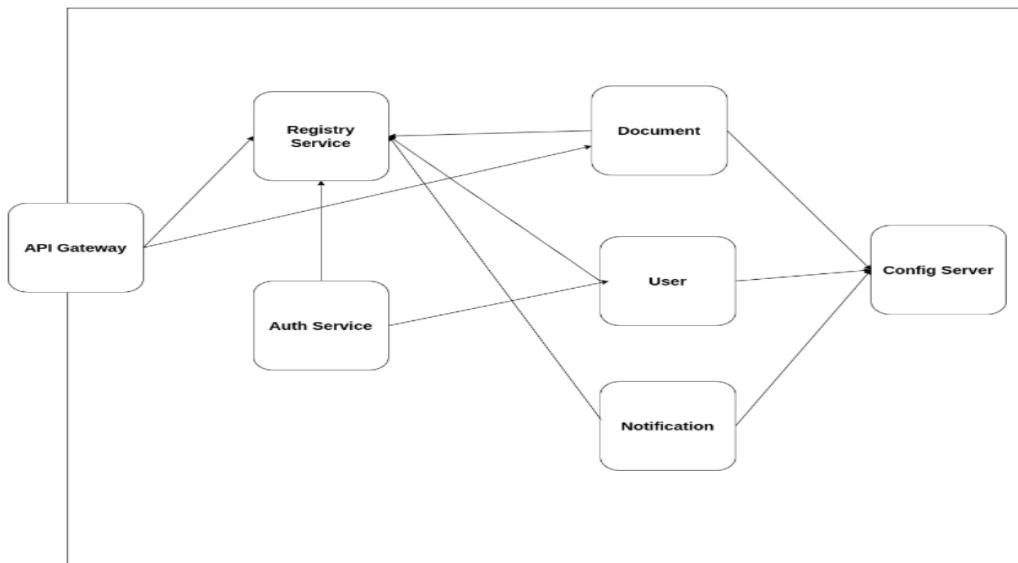
- Mô tả luồng gửi tài liệu:



- Mô tả luồng ký tài liệu:



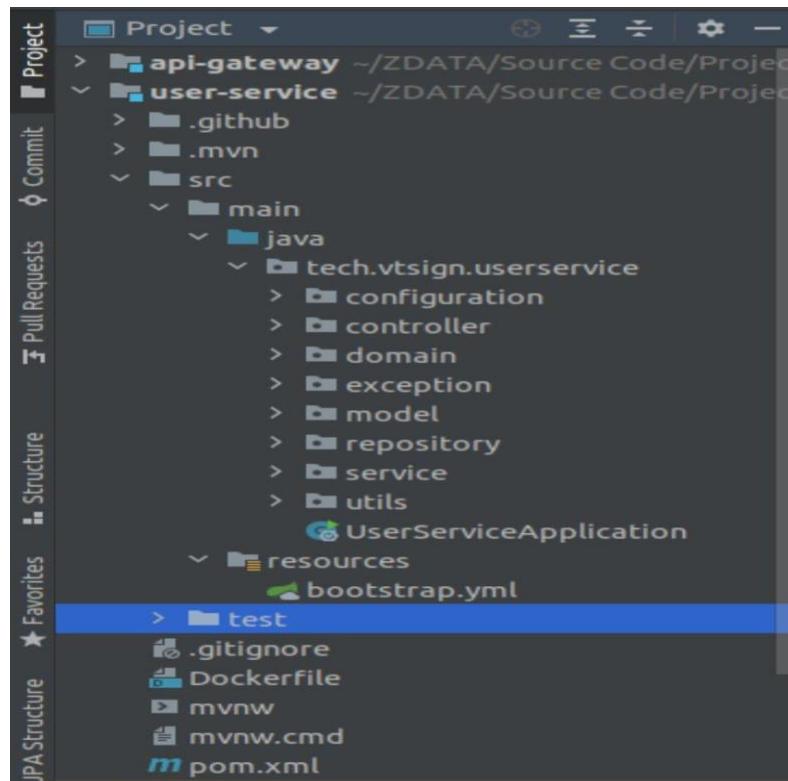
Kiến trúc Logical:



- Logical view của nhóm thể hiện cấu trúc và tổ chức thiết kế giữa các service trong hệ thống, logical view được dùng để phân tích và thiết kế.

- Miêu tả các lớp đối tượng và mối quan hệ, gửi thông điệp cho nhau để cung cấp các thiết lập cấu hình được cài sẵn.

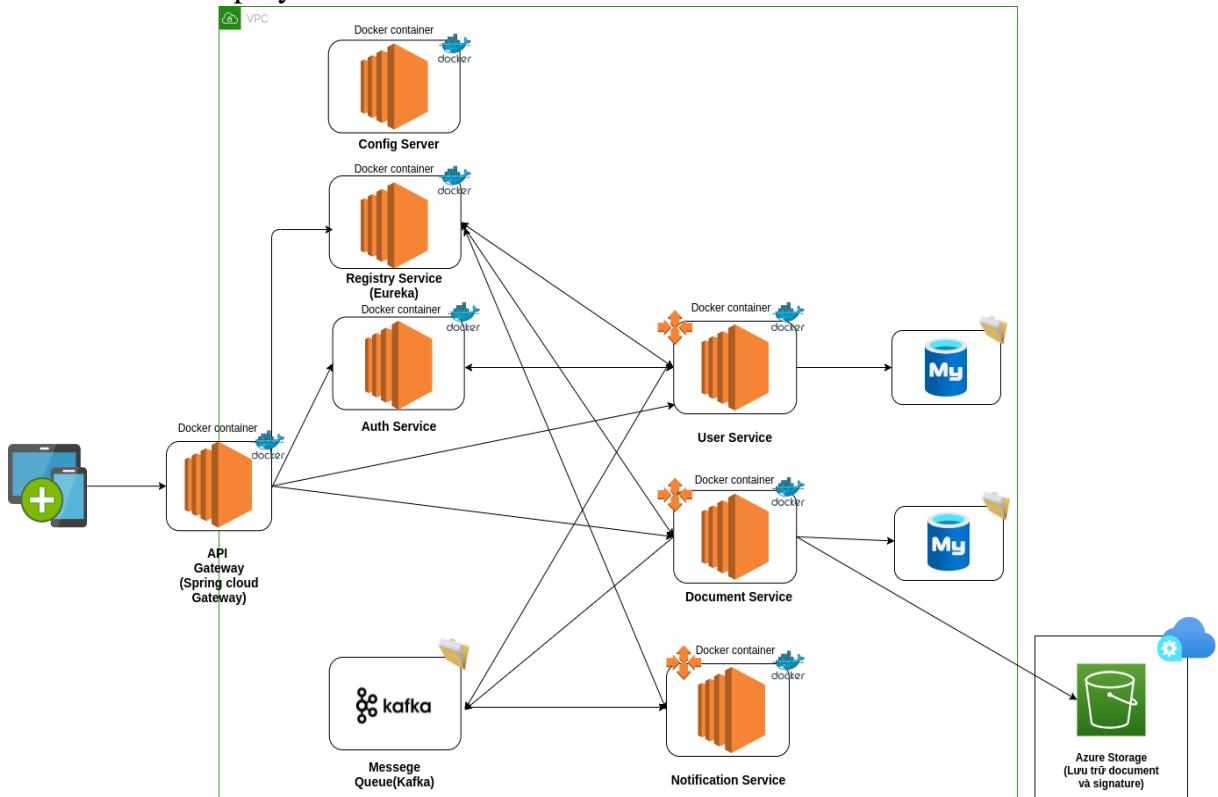
Kiến trúc Development:



- Phát triển theo kiến trúc microservices
- Mỗi service backend gồm có các packages chính sau:

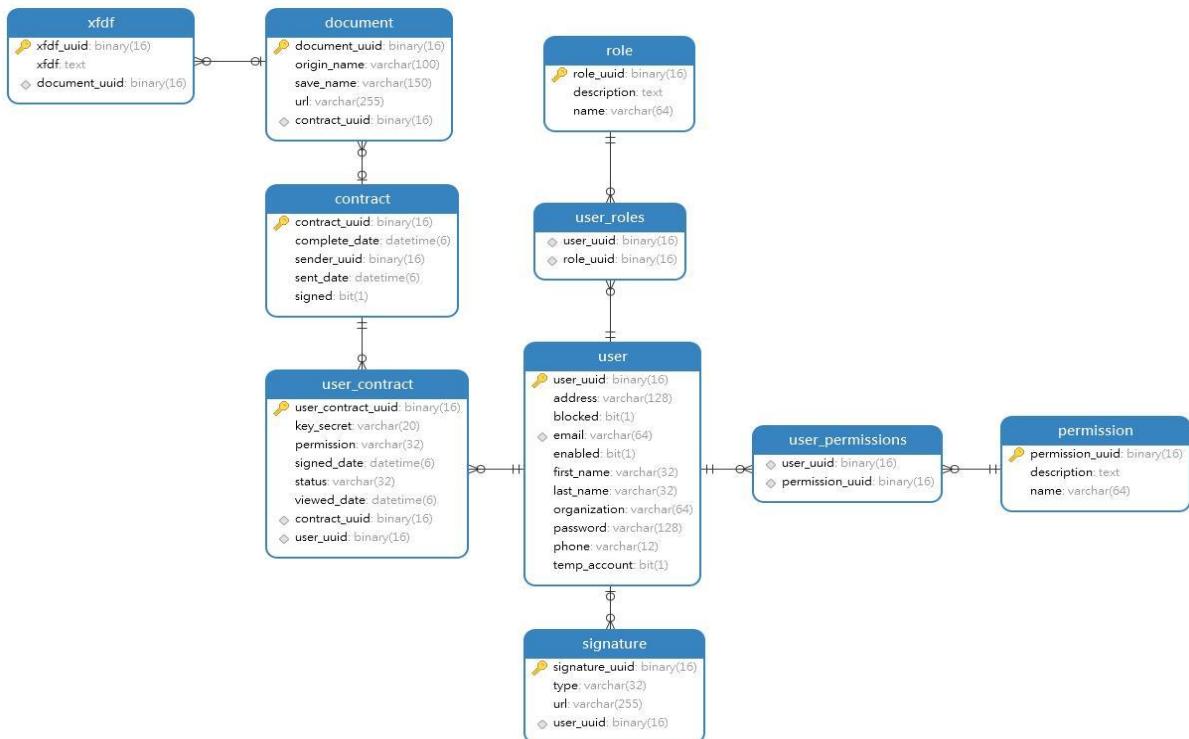
- + configuration: Các cấu hình cho service
- + controller: Tiếp nhận các yêu cầu gửi tới
- + domain: Transfer dữ liệu với database
- + exception: Xử lý các lỗi và ngoại lệ
- + model: Transfer dữ liệu với các services khác
- + repository: Thao tác dữ liệu với database
- + service: Xử lý logic
- + utils: Các hàm xử lý tiện ích
- + test: Viết các unit test
- + resources: chứa các file cấu hình biến môi trường cho service
- Một số file và thư mục khác:
 - + github: thư mục viết các cấu hình chạy hệ thống CI/CD tự động với Github Actions
 - + Dockerfile: cấu hình để xây dựng docker image để tiện hơn cho việc triển khai
 - + pom.xml: quản lý các thư viện phụ thuộc

Kiến trúc Deployment:



- Thể hiện cách triển khai hệ thống trên quy trình thực tế
 - Hệ thống phát triển theo hướng microservices:
 - + Mỗi service có thể có nhiều thể hiện (instance), được triển khai bên trong các Docker Container và kết nối tới một database độc lập.
 - + Hệ quản trị CSDL được sử dụng là MySQL.
 - + Api Gateway sử dụng Spring Cloud Gateway để nhận các yêu cầu từ người dùng.
 - + Các service đăng ký với nhau thông qua Registry Service (sử dụng Eureka Server), có thể gọi nhau bằng service-id.
 - + Message Queue sử dụng Kafka.
 - + Config Server sử dụng Spring Cloud Config Server để cấu hình các service trên Github repository.
 - + Tài liệu được lưu trữ trên Microsoft Azure Storage.
 - + Có thể triển khai trên các hệ thống máy chủ chạy độc lập với nhau (loosely coupled)
 - + Docker Image của mỗi service được build và đẩy lên Docker Hub một cách tự động thông qua hệ thống build tự động (Github Actions).

***Mô hình dữ liệu:**



***Các mục tiêu kiểm thử:**

- Load, Stress testing: Apache Jmeter là một dự án Apache có thể được sử dụng như một công cụ kiểm tra tải để phân tích và đo lường hiệu suất của nhiều loại dịch vụ, tập trung vào các ứng dụng web.
- Penetration Testing: Nmap - Network Mapper là một công cụ bảo mật được phát triển bởi Gordon Lyon. Nmap có mã nguồn mở, miễn phí, dùng để quét cổng và lỗ hổng bảo mật.

***Dự kiến phương pháp so sánh, đánh giá hệ thống:**

Bảng so sánh các tính năng ký kết cơ bản của 4 hệ thống

Tính năng	SignNow	AdobeSign	DocuSign	VTSign
Gửi tài liệu cần ký	✓	✓	✓	✓
Mời ký số lượng lớn	✓	✓	✓	✓
Chỉnh sửa tài liệu trước khi gửi	✓	✓	✗	✗
Tạo các mẫu có thể sử dụng lại	✓	✓	✓	✓
Thêm thương hiệu được cá nhân hóa	✓	✓	✓	✗
Gửi lời mời qua liên kết	✓	✓	✓	✓
Ký kết trực tiếp	✓	✗	✗	✓
Xác định các loại chữ ký được phép	✓	✗	✗	✗
Trò chuyện trực tiếp	✓	✓	✗	✗
Bảo vệ môi đe dọa nâng cao	✓	✗	✓	✓
Quản lý tài liệu có thời hạn	✓	✗	✓	✓

Bảng ưu điểm khuyết điểm của 4 hệ thống

Hệ thống	Ưu điểm	Khuyết điểm
SignNow	<ul style="list-style-type: none"> - Đăng nhập/dăng ký nhanh bằng tài khoản Facebook, Google. - Hệ thống sử dụng eSignature đơn giản, giao diện hiện đại 	<ul style="list-style-type: none"> - Hỗ trợ ít ngôn ngữ, tập trung thị trường nước ngoài. - Giao diện trình duyệt chưa có chế độ nền tối.

	<p>dễ sử dụng, có LiveChat hỗ trợ khách hàng.</p> <ul style="list-style-type: none"> - Bản dùng thử không giới hạn nhiều chức năng, bản quyền giá hợp lý, phù hợp nhiều đối tượng. 	<ul style="list-style-type: none"> - Không dùng phương pháp Digital Signature, bảo mật còn hạn chế.
DocuSign	<ul style="list-style-type: none"> - Sử dụng phương pháp định danh Self-Signed Digital Signature, đòi hỏi người dùng trả tiền mua Chứng thực (Certificate). - Sử dụng các công nghệ mã hóa và bảo mật dữ liệu, thông báo qua tin nhắn (SMS Delivery). - Phù hợp với các công ty quy mô lớn, đặt nặng về bảo mật chuyên môn. - Nhiều công cụ hỗ trợ kèm theo, đáp ứng nhiều tiêu chuẩn bảo mật. 	<ul style="list-style-type: none"> - Không có đăng nhập/đăng ký nhanh bằng tài khoản Facebook, Google. - Hệ thống nặng, truy cập chậm ở khu vực Việt Nam, không có LiveChat. - Không được chỉnh sửa tài liệu trước khi gửi và ký kết trực tiếp. - Hỗ trợ ít ngôn ngữ, tập trung thị trường Mỹ, bản quyền giá cao.
AdobeSign	<ul style="list-style-type: none"> - Sử dụng phương pháp Digital Signature, đòi hỏi người dùng trả tiền mua 1 Chứng thực (Certificate). - Đóng bộ và cá nhân hóa với phần mềm liên quan của Adobe. 	<ul style="list-style-type: none"> - Đăng nhập/đăng ký phức tạp, hệ thống khá chậm và tải nhiều thông tin và thông báo. - Giao diện cũ khó nhìn, nhiều tính năng thừa và khó sử dụng, Không ký trực tiếp.

		<ul style="list-style-type: none"> - Bản quyền và phí sử dụng cao, bản dùng thử nhiều ràng buộc chức năng.
VTSign	<ul style="list-style-type: none"> - Hệ thống xử lý và hiển thị nhanh, không hiện các thông báo gây nhiễu. - Bảo mật tài liệu, xác thực người dùng bằng email và số điện thoại cá nhân. - Giao diện thao tác đơn giản dễ sử dụng, phù hợp nhiều lứa tuổi. - Hỗ trợ ngôn ngữ tiếng việt, bản quyền sử dụng phù hợp người Việt Nam. 	<ul style="list-style-type: none"> - Không có đăng nhập/đăng ký nhanh bằng tài khoản Facebook, Google. - Không chỉnh sửa tài liệu trước khi gửi, không LiveChat, không hỗ trợ nhiều ngôn ngữ. - Phù hợp với người dùng cá nhân, các tổ chức nhỏ, không chuyên môn cao.

Đánh giá

- Các hệ thống trên thị trường rất đa dạng, với nhiều tính năng kèm theo. Thực tế chức năng và nhu cầu của người sử dụng không nhiều, các hệ thống phổ biến đa số tập trung vào thị trường Mỹ và nước ngoài. Dẫn đến bất cập không hỗ trợ ngôn ngữ tiếng việt, nhiều chức năng thừa và không được sử dụng tại Việt Nam, yêu cầu thẻ thanh toán trực tuyến quốc tế khi đăng ký, bản quyền giá cao khi quy đổi ngoại tệ tiền Việt Nam.
- Qua so sánh, đánh giá hệ thống của nhóm với các hệ thống tương tự. Các thành viên nhóm đã đưa ra thống nhất tổng quan về hệ thống VTSign. Thực hiện đúng các kế hoạch đề ra theo các mục 2.2 Mục tiêu đề tài và 2.3 Phạm vi của đề tài.

***Danh sách các công nghệ, công cụ sử dụng:**

- Mô hình Kanban: để thiết kế và triển khai đồ án.
- Kiến trúc Microservice: một kỹ thuật phát triển phần mềm, với nhiều lợi ích mang lại về khả năng mở rộng và bảo trì.
- Trello: để chia việc theo danh sách và các thẻ (kiểu Kanban).
- Figma: thiết kế giao diện bản mẫu các chức năng, luồng hệ thống.
- ReactJs: một thư viện JavaScript front-end mã nguồn mở miễn phí.
- Material-UI: một thư viện các React Component.
- Intellij: một IDE Java để phát triển các phần mềm máy tính.
- Visual Studio Code: một trình biên tập mã được phát triển bởi Microsoft.
- Postman: công cụ thao tác với API như call và test API
- PhpMyAdmin: quản lý Cơ sở dữ liệu
- Microsoft Azure: quản lý, lưu trữ các tài liệu của user
- Github: một dịch vụ cung cấp kho lưu trữ mã nguồn
- Digitalocean: thiết lập quản lý các Cloud máy chủ chạy các Service.

2.5 Kết quả dự kiến của đề tài

- Hệ thống ký kết văn bản trực tuyến hoàn chỉnh với các chức năng đặt ra.
- Mã nguồn ứng dụng, trang website hệ thống ký kết văn bản VTSign.
- Tài liệu báo cáo chi tiết mà nhóm đã tìm hiểu trong suốt quá trình thực hiện đồ án.
Kinh nghiệm tích lũy đạt được khi thực hiện một đồ án thực tế.

2.6 Kế hoạch thực hiện

Thời gian	Công việc	Người thực hiện
15/08/2021 – 31/08/2021	<ul style="list-style-type: none">- Liên hệ giảng viên hướng dẫn xem xét, bàn luận để thống nhất nhận thực hiện đề tài.- Tìm hiểu thêm về đề tài. Nghiên cứu quy trình thực hiện đồ án của giảng viên hướng dẫn.- Nộp đơn đăng ký thực tập dự án tốt nghiệp.	Tất cả thành viên

01/09/2021 – 30/09/2021	<ul style="list-style-type: none"> - Giai đoạn khởi tạo dự án, khảo sát thị trường với các hệ thống tương tự. - Chuẩn bị bản mẫu Prototype và Proof of Concept. - Khởi tạo và hoàn thành chương 1 báo cáo. Khởi tạo đề cương chi tiết, kế hoạch sơ bộ. 	Tất cả thành viên
01/10/2021 – 31/10/2021	<ul style="list-style-type: none"> - Thiết kế luồng hoạt động dự kiến của hệ thống. - Tìm hiểu và lựa chọn các công cụ, công nghệ, thư viện hỗ trợ xây dựng hệ thống. - Cập nhật chương 2 báo cáo và đề cương. 	Tất cả thành viên
01/11/2021 – 30/11/2021	<ul style="list-style-type: none"> - Tổ chức mã nguồn, thiết kế giao diện trang chủ và đăng nhập. - Hoàn tất khái niệm chương 2, cập nhật chương 5 và đề cương. Gửi giảng viên góp ý để chỉnh sửa tài liệu. - Hoàn tất và nộp đề cương chi tiết cho khoa 10/11. 	Tất cả thành viên
01/12/2021 – 31/12/2021	<ul style="list-style-type: none"> - Phát triển thiết kế kiến trúc hệ thống. Triển khai CI/CD. Xây dựng các chức năng đã đặt ra. - Thực hiện triển khai chức năng xử lý dữ liệu người dùng trên máy khách - chủ. - Hoàn tất cơ bản giao diện hệ thống. Triển khai phiên bản thử nghiệm. - Hoàn tất chương 5, cập nhật chương 4 và thảo luận với giảng viên. 	Tất cả thành viên
01/01/2022 – 31/01/2022	<ul style="list-style-type: none"> - Tiếp tục xử lý các vấn đề còn lại của hệ thống, đánh giá chung và cải tiến. - Hoàn tất chương 4, cập nhật chương 3 và thảo luận với giảng viên. 	Tất cả thành viên
01/02/2022 – 28/02/2022	<ul style="list-style-type: none"> - Thực hiện kiểm thử, triển khai phiên bản chính thức đầu tiên. - Hoàn tất chương 3, cập nhật làm xong chương 2, kiểm tra toàn bộ báo cáo. - Chuẩn bị và nộp đơn đăng ký bảo vệ đồ án 23/02. 	Tất cả thành viên
01/03/2021 – Thời gian còn lại	<ul style="list-style-type: none"> - Cập nhật, kiểm tra hệ thống và máy chủ lần cuối. Hoàn tất báo cáo đề tài. - Thực hiện chỉnh sửa báo cáo đề tài lần cuối. Chuẩn bị tài liệu buổi bảo vệ đề tài. 	Tất cả thành viên

Tài liệu

- [1] N. X. Sơn, Chữ ký điện tử và ứng dụng. PhD thesis, Trường Đại học Bách Khoa Hà Nội, 2015.
- [2] J. Katz, Digital signatures. Springer Science & Business Media, 2010.
- [3] C. Richardson, Microservices patterns: with examples in Java. Simon and Schuster, 2018.
- [4] X. WANG, Y. WANG, and F. WANG, “The implement of a pair of secret key of digital signature algorithm by using java programming language,” Journal of Logistical Engineering University, vol. 3, 2006.
- [5] C. Adams and S. Lloyd, Understanding PKI: concepts, standards, and deployment considerations. Addison-Wesley Professional, 2003.

XÁC NHẬN CỦA
GIẢNG VIÊN HƯỚNG DẪN
(Ký và ghi rõ họ tên)

TP.Hồ Chí Minh, ngày 10 / tháng 11 / năm 2021
NHÓM SINH VIÊN THỰC HIỆN
(Ký và ghi rõ họ tên)

MỤC LỤC

Chương 1	GIỚI THIỆU	1
1.1	Giới thiệu đề tài.....	1
1.2	Khảo sát thị trường.....	1
1.2.1	“SignNow” do airSlate Inc phát triển	1
1.2.2	“DocuSign” do DocuSign Inc phát triển.....	2
1.2.3	“AdobeSign” do Adobe Inc phát triển	3
1.2.4	So sánh nhận xét 4 hệ thống ký kết văn bản	3
1.2.5	Tình hình sử dụng chữ ký điện tử ở Việt Nam	5
1.3	Lý do lựa chọn đề tài.....	6
1.4	Mục tiêu thực hiện	7
1.5	Yêu cầu chức năng.....	7
1.6	Phạm vi đề tài.....	15
Chương 2	LÝ THUYẾT NỀN TẢNG	16
2.1	Các khái niệm liên quan tới Chữ ký số Chữ ký điện tử.....	16
2.1.1	Chữ ký số - Digital signatures.....	16
2.1.2	Chữ ký điện tử - Electronic Signatures	18
2.1.3	Sự khác biệt giữa chữ ký số và chữ ký điện tử	18
2.2	Ký kết văn bản trong đời sống thực	19
2.2.1	Quy trình ký kết	19
2.2.2	Các vấn đề thường gặp.....	21
2.2.3	Các phương pháp xử lý	22
2.3	Ký kết văn bản bằng máy tính	22
2.3.1	Các vấn đề thường gặp.....	22
2.3.2	Các phương pháp xử lý	22

2.4	Phương pháp ký kết bằng Digital Signature	23
2.4.1	Quy trình ký kết	23
2.4.2	Xác thực người ký	23
2.4.3	Ưu điểm và khuyết điểm	24
2.5	Phương pháp ký kết bằng cách định danh	24
2.5.1	Quy trình ký kết	24
2.5.2	Định danh người ký	25
2.5.3	Ưu điểm và khuyết điểm	25
Chương 3	GIẢI PHÁP ĐỀ TÀI.....	26
3.1	Giải pháp tổng quát.....	26
3.1.1	Giải pháp ký kết	26
3.1.1.1	Quy trình chung của việc gửi tài liệu	26
3.1.1.2	Quy trình chung của việc ký tài liệu	27
3.1.2	Giải pháp quản lý tài liệu	27
3.2	Thiết kế hệ thống.....	29
3.2.1	Thiết kế giao diện hệ thống.....	29
3.2.1.1	Giao diện trang chủ hệ thống	29
3.2.1.2	Giao diện trang ký kết	30
3.2.1.3	Giao diện trang quản lý tài liệu	32
3.2.1.4	Giao diện trang quản trị thông kê	33
3.2.1.5	Giao diện trang quản trị người dùng	34
3.2.2	Thiết kế kiến trúc hệ thống.....	35
3.2.2.1	Giải pháp kiến trúc Microservice	35
3.2.2.2	Giải pháp mô hình API Gateway	38
3.2.3	Thiết kế giải pháp lưu trữ dữ liệu.....	39

3.3	Tổng kết	43
Chương 4	CÀI ĐẶT VÀ TRIỂN KHAI	44
4.1	Spring Framework.....	44
4.1.1	Giới thiệu tổng quan.....	44
4.1.2	Một số dự án của Spring được sử dụng trong đồ án	44
4.2	ReactJs	45
4.2.1	Tạo ứng dụng web động trở nên dễ dàng hơn.....	45
4.2.2	Các thành phần có thể tái sử dụng	45
4.2.3	Nâng cao hiệu suất	46
4.2.4	Các phần mềm sử dụng ReactJs hiện nay	46
4.3	Cài đặt kiến trúc	46
4.3.1	Config Server	46
4.3.2	Registry Service	47
4.3.3	API Gateway	49
4.3.4	Notification service	50
4.3.5	Auth Service.....	51
4.3.6	Document Service	52
4.3.7	User Service	52
4.4	Cài đặt chức năng.....	53
4.4.1	Đăng ký, đăng nhập.....	53
4.4.1.1	Đăng ký	53
4.4.1.2	Đăng nhập.....	54
4.4.2	Ký kết hợp đồng	56
4.4.2.1	Gửi hợp đồng.....	56
4.4.2.2	Ký hợp đồng	57

4.4.2.3	Thanh toán chi phí sử dụng dịch vụ	59
4.4.3	Quản lý hợp đồng	60
4.4.4	Quản lý người sử dụng dịch vụ	64
4.4.4.1	Thêm người dùng	64
4.4.4.2	Cập nhật người dùng	65
4.4.4.3	Xem danh sách người dùng đang hoạt động	66
4.4.4.4	Xem danh sách người dùng đã bị khóa	66
4.4.4.5	Xem danh sách người dùng đã bị xóa	67
4.4.4.6	Khóa và mở người dùng	68
4.4.4.7	Xóa người dùng	69
4.4.5	Báo cáo cho quản trị viên về tổng quan hệ thống	70
4.4.5.1	Thông kê tổng quan	70
4.4.5.2	Vẽ biểu đồ số lượng người đăng ký sử dụng	72
4.4.5.3	Vẽ biểu đồ tổng quan về tiền nạp vào hệ thống	73
4.4.5.4	Vẽ biểu đồ tổng quan về số lượng tài liệu đã gửi hoặc hoàn thành ..	74
4.4.6	Báo cáo cho quản trị viên về quá trình truy cập và dung lượng của từng người dùng	75
4.4.6.1	Báo cáo về lịch sử giao dịch người dùng	75
4.4.6.2	Báo cáo về số lượng tài liệu của người dùng	76
4.4.7	Báo cáo cho người dùng về quá trình truy cập và dung lượng sử dụng của tài khoản	77
4.4.7.1	Báo cáo về lịch sử giao dịch	77
4.4.7.2	Báo cáo về số lượng tài liệu	78
Chương 5	TỔNG KẾT VÀ ĐÁNH GIÁ	80
5.1	Kiến thức thu được	80

5.2	Sản phẩm thu được.....	81
5.2.1	Môi trường phát triển	81
5.2.2	Môi trường triển khai	81
5.2.3	Hiệu năng hệ thống	82
5.2.4	Các chức năng đã cài đặt.....	85
5.2.5	So sánh tính năng với các hệ thống khác trên thị trường	88
5.3	So sánh kết quả thu được với mục tiêu ban đầu	89
5.4	Định hướng phát triển và nghiên cứu	91
5.5	Lời kết	92
PHỤ LỤC 1: GIAO DIỆN CHỨC NĂNG VTSIGN.....		93
TÀI LIỆU THAM KHẢO.....		102

DANH SÁCH CÁC HÌNH

Hình 1-1: Màn hình sau khi đăng nhập hệ thống SignNow	2
Hình 1-2: Màn hình sau khi đăng nhập hệ thống DocuSign	2
Hình 1-3: Màn hình sau khi đăng nhập hệ thống AdobeSign	3
Hình 2-1: Cách thức hoạt động của Chữ ký số	17
Hình 2-2: Quy trình ký kết văn bản đòi sống thực – Bước 1	19
Hình 2-3: Quy trình ký kết văn bản đòi sống thực – Bước 2	19
Hình 2-4: Quy trình ký kết văn bản đòi sống thực – Bước 3	20
Hình 2-5: Quy trình ký kết văn bản đòi sống thực – Bước 4	20
Hình 2-6: Quy trình ký kết văn bản đòi sống thực – Bước 5	20
Hình 2-7: Quy trình ký kết văn bản đòi sống thực – Bước 6	21
Hình 2-8: Quy trình ký kết văn bản đòi sống thực – Bước 7	21
Hình 2-9: Quy trình ký kết văn bản bằng Digital Signature – Bước 1	23
Hình 2-10: Quy trình ký kết văn bản bằng Digital Signature – Bước 2	23
Hình 2-11: Quy trình ký kết văn bản bằng Digital Signature – Bước 3	23
Hình 2-12: Quy trình ký kết văn bản bằng Digital Signature – Xác thực	24
Hình 2-13: Quy trình ký kết văn bản bằng định danh – Bước 1	24
Hình 2-14: Quy trình ký kết văn bản bằng định danh – Bước 2	25
Hình 2-15: Quy trình ký kết văn bản bằng định danh – Bước 3	25
Hình 3-1: Giải pháp quản lý tài liệu – Phần 1	28
Hình 3-2: Giải pháp quản lý tài liệu – Phần 2	29
Hình 3-3: Giao diện trang chủ hệ thống	30
Hình 3-4: Giao diện trang ký kết – Thêm tài liệu	30
Hình 3-5: Giao diện trang ký kết – Chọn người nhận và cài đặt	31
Hình 3-6: Giao diện trang ký kết – Ký tên và thông tin khác	31
Hình 3-7: Giao diện trang ký kết – Kiểm tra và gửi file	32
Hình 3-8: Giao diện trang quản lý tài liệu	32
Hình 3-9: Giao diện trang quản trị - Thông kê phần 1	33
Hình 3-10: Giao diện trang quản trị - Thông kê phần 2	33

Hình 3-11: Giao diện trang quản trị - Thống kê phần 3	34
Hình 3-12: Giao diện trang quản trị - Thống kê phần 4	34
Hình 3-13: Giao diện trang quản trị người dùng – Phần 1	35
Hình 3-14: Giao diện trang quản trị người dùng – Phần 2	35
Hình 3-15: Kiến trúc Monolithic	36
Hình 3-16: Kiến trúc Microservice.....	37
Hình 3-17: Cách hoạt động của Direct Pattern.....	38
Hình 5-1: Kết quả kiểm thử với 250.000 người dùng truy cập	82
Hình 5-2: Kết quả kiểm thử với 100.000 người dùng truy cập	82
Hình 5-3: Kết quả kiểm thử với 500.000 người dùng truy cập	83
Hình 5-4: Kết quả kiểm thử với 1.000.000 người dùng truy cập	84
Hình 5-5: Kết quả kiểm thử với 750.000 người dùng truy cập	84

DANH SÁCH CÁC BẢNG

Bảng 1-1: So sánh các tính năng ký kết cơ bản của 4 hệ thống	3
Bảng 1-2: Ưu điểm và khuyết điểm của 4 hệ thống ký kết	4
Bảng 1-3: Bảng yêu cầu chức năng đặt ra của đề tài.....	7
Bảng 2-1: So sánh chữ ký điện tử và chữ ký số	18
Bảng 3-1: Bảng người dùng – Table User.....	40
Bảng 3-2: Bảng hợp đồng – Table Contract.....	41
Bảng 3-3: Bảng hợp đồng người dùng – Table User Contract.....	42
Bảng 4-1: Bảng trạng thái HTTP cho việc đăng ký tài khoản	54
Bảng 4-2: Bảng trạng thái HTTP cho việc đăng nhập	55
Bảng 4-3: Bảng trạng thái HTTP cho việc gửi hợp đồng.....	56
Bảng 4-4: Bảng trạng thái HTTP cho việc mở khóa tài liệu	57
Bảng 4-5: Bảng trạng thái HTTP cho việc ký kết văn bản.....	58
Bảng 4-6: Bảng trạng thái HTTP cho việc thanh toán chi phí	59
Bảng 4-7: Bảng trạng thái HTTP chov việc lọc hợp đồng	61
Bảng 4-8: Bảng trạng thái HTTP cho việc xóa hợp đồng	61
Bảng 4-9: Bảng trạng thái HTTP cho việc khôi phục hợp đồng	62
Bảng 4-10: Bảng trạng thái HTTP cho việc ẩn hợp đồng	63
Bảng 4-11: Bảng trạng thái HTTP cho việc thêm người dùng.....	64
Bảng 4-12: Bảng trạng thái HTTP cho việc cập nhật người dùng	65
Bảng 4-13: Bảng trạng thái HTTP cho việc xem danh sách người dùng hoạt động.....	66
Bảng 4-14: Bảng trạng thái HTTP cho việc xem danh sách người dùng đã bị khóa	67
Bảng 4-15: Bảng trạng thái HTTP cho việc xem danh sách người dùng đã bị xóa	67
Bảng 4-16: Bảng trạng thái HTTP cho việc khóa và mở người dùng	68
Bảng 4-17: Bảng trạng thái HTTP cho việc xóa người dùng	69
Bảng 4-18: Bảng trạng thái HTTP cho việc thống kê tổng quan	71
Bảng 4-19: Bảng trạng thái HTTP cho việc vẽ biểu đồ số lượng người đăng ký sử dụng	72

Bảng 4-20: Bảng trạng thái HTTP cho việc vẽ biểu đồ tổng quan về tiền nạp vào hệ thống	73
Bảng 4-21: Bảng trạng thái HTTP cho việc vẽ biểu đồ tổng quan về số lượng tài liệu đã gửi	74
Bảng 4-22: Bảng trạng thái HTTP cho việc báo cáo lịch sử giao dịch người dùng	75
Bảng 4-23: Bảng trạng thái HTTP cho việc báo cáo số lượng tài liệu người dùng	77
Bảng 4-24: Bảng trạng thái HTTP cho việc báo cáo lịch sử giao dịch	78
Bảng 4-25: Bảng trạng thái HTTP cho việc báo cáo số lượng tài liệu.....	79
Bảng 5-1: Các thư viện và nền tảng sử dụng.....	81
Bảng 5-2: Bảng chức năng đã hoàn thành	85
Bảng 5-3: Bảng so sánh tính năng của VTSign với hệ thống khác	88
Bảng 5-4: Bảng so sánh mục tiêu ban đầu với kết quả thu được	89

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Ký hiệu	Điễn giải
1	FAQ	Frequently Asked Questions
2	e-Signature	Electronic Signature
3	PKI	Public key Infrastructure
4	CA	Certificate Authority
5	SMS OTP	Short Message Service One Time Password

TÓM TẮT BÁO CÁO

Báo cáo đề tài tốt nghiệp là tài liệu chính của thực tập dự án tốt nghiệp, nội dung của báo cáo sẽ đề cập tới các kiến thức, kỹ thuật liên quan trong quá trình thực hiện đồ án. Báo cáo bao gồm các phần như sau:

Chương 1 – Giới thiệu: Giới thiệu về đề tài, trình bày lý do xây dựng hệ thống ký kết văn bản trực tuyến. Trình bày nhận xét 3 hệ thống ký kết văn bản tiêu biểu. Mô tả chi tiết các chức năng chính của hệ thống. Nêu ra mục tiêu, phạm vi phát triển đề tài về hệ thống của nhóm.

Chương 2 – Lý thuyết nền tảng: Trình bày các khái niệm về Chữ ký số, Chữ ký điện tử. Tìm hiểu các quy trình ký kết văn bản trong đời sống thực, bằng máy tính. Bên cạnh đó đề cập đến phương pháp ký kết văn bằng bằng Chữ ký số, bằng cách định danh.

Chương 3 – Giải pháp đề tài: Nêu ra các công cụ và giải pháp tổng quát cho các chức năng của ứng dụng ký kết văn bản và lý giải cho việc chọn phương án. Ngoài ra, còn trình bày một số phương hướng thiết kế hệ thống và thiết kế giao diện để cài đặt và triển khai được hiệu quả.

Chương 4 – Cài đặt và triển khai: Trình bày cách hoạt động, cách cài đặt chi tiết của một số chức năng. Bên cạnh đó, chương 4 còn đề cập đến một số khó khăn trong quá trình cài đặt và giải pháp cho chúng.

Chương 5 – Tổng kết và đánh giá: Phần này sẽ liệt kê các kiến thức mà nhóm đạt được, sản phẩm, hiệu năng hệ thống và những chức năng thu được từ quá trình thực hiện. Trong chương này còn so sánh ứng dụng của luận văn với một số ứng dụng hiện có trên thị trường, phương hướng phát triển trong tương lai trong ứng dụng.

Chương 1

GIỚI THIỆU

1.1 Giới thiệu đề tài

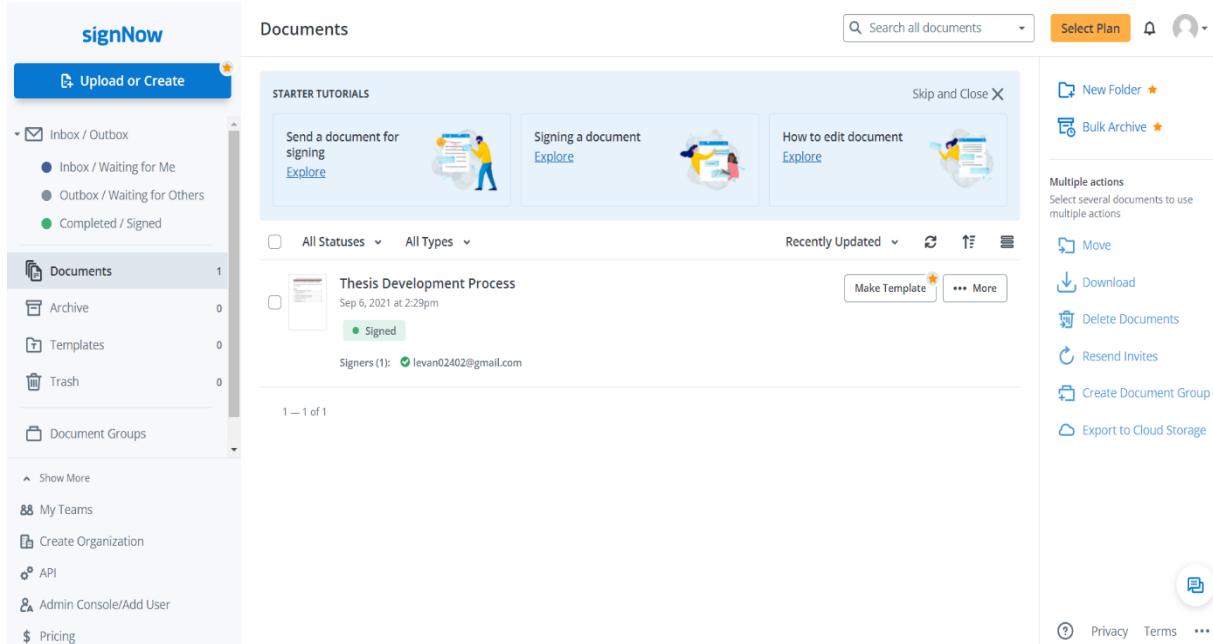
Hiện nay, khi mà mọi vấn đề trong cuộc sống đều có thể giải quyết bằng ứng dụng công nghệ và số hóa thông tin. Chữ ký điện tử được sử dụng rộng rãi và nhiều người biết đến, cụ thể trong các giao dịch điện tử. Các ứng dụng hiện có trên thị trường rất đa dạng, với nhiều tính năng kèm theo và có thể mang nặng tính chất kinh doanh quảng cáo. Bên cạnh đó, chức năng và nhu cầu của người sử dụng không nhiều, các hệ thống chữ ký điện tử phổ biến hầu hết tập trung vào thị trường Mỹ và nước ngoài. Dẫn đến bất cập không hỗ trợ ngôn ngữ tiếng Việt, nhiều chức năng thừa và không được sử dụng tại quốc gia Việt Nam, yêu cầu thẻ thanh toán trực tuyến quốc tế khi đăng ký, bản quyền giá cao khi quy đổi ngoại tệ tiền Việt Nam. Vì thế chúng tôi muốn tạo ra một hệ thống ký kết văn bản trực tuyến, tập trung thị trường trong nước, ưu tiên sự tối giản, nhanh gọn. Tên là VTSign – Hệ thống ký kết văn bản trực tuyến (Building e-signature system).

1.2 Khảo sát thị trường

1.2.1 “SignNow” do airSlate Inc phát triển

Giới thiệu [1]

SignNow là nhà cung cấp công nghệ chữ ký điện tử được phát triển tại Hoa Kỳ. Nền tảng phần mềm dưới dạng dịch vụ của công ty cho phép các cá nhân và doanh nghiệp ký, quản lý tài liệu từ bất kỳ máy tính nào. Sản phẩm được cung cấp miễn phí trên các thiết bị iPhone, iPad và Android, cho phép tải lên tài liệu từ tài khoản email, máy ảnh hoặc Dropbox của điện thoại thông minh và nhấn để chèn chữ ký.



Hình 1-1: Màn hình sau khi đăng nhập hệ thống SignNow

1.2.2 “DocuSign” do DocuSign Inc phát triển

Giới thiệu [2]

DocuSign là hãng công nghệ tiên phong và đứng số 1 trong mảng chữ ký điện tử trên thế giới. Cung cấp giải pháp e-signature cho 500.000 doanh nghiệp và hàng trăm triệu người dùng tại 180 quốc gia toàn cầu. Cung cấp chữ ký an toàn và đơn giản cho các văn bản điện tử và thu thập chữ ký từ những tài liệu khác. Ứng dụng loại bỏ hết sự phức tạp, chi phí và thiểu an toàn trong in ấn, fax, scan các tài liệu cho việc ký kết.

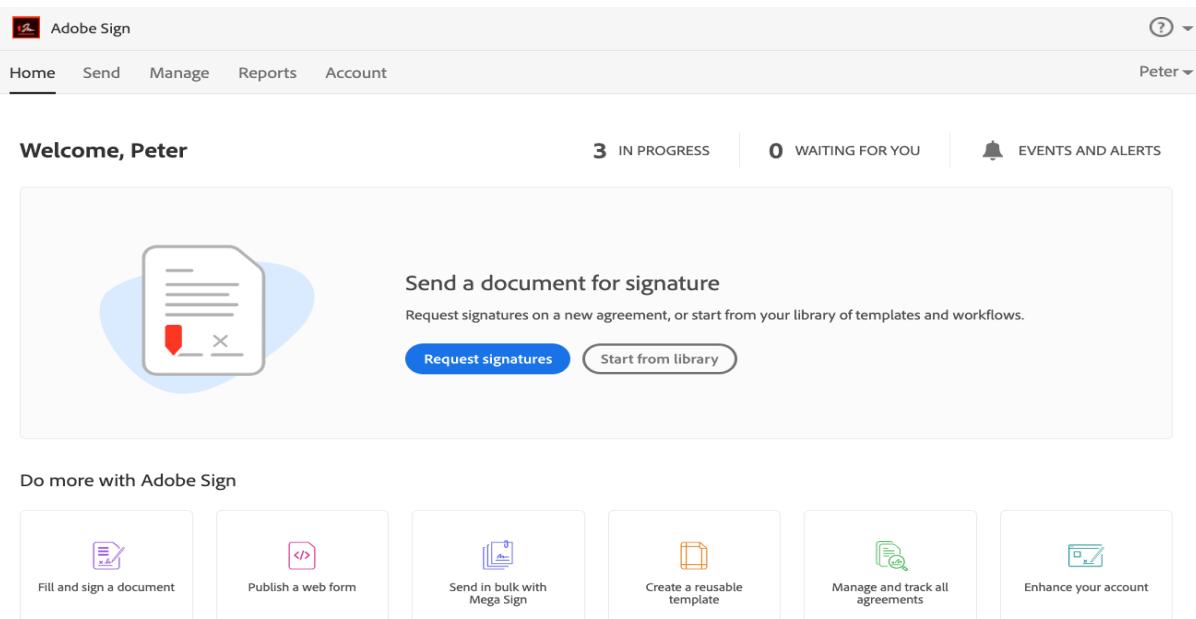
Subject	Status	Last Change
Please DocuSign: Thesis Development Process.pdf To: Van Le, Van Le	Completed	10/3/2021 08:51:00 pm

Hình 1-2: Màn hình sau khi đăng nhập hệ thống DocuSign

1.2.3 “AdobeSign” do Adobe Inc phát triển

Giới thiệu [3]

AdobeSign là một dịch vụ dựa trên đám mây cho phép một cá nhân hoặc tổ chức gửi, bảo mật, theo dõi và quản lý các chữ ký điện tử có quy trình (từ khi gửi tài liệu cho đến khi kết thúc bằng chữ ký). Dịch vụ này nhằm thay thế chữ ký giấy và mực in vật lý bằng một giải pháp thay thế điện tử hoàn toàn tự động. AdobeSign được phát triển bởi nền tảng EchoSign.



Hình 1-3: Màn hình sau khi đăng nhập hệ thống AdobeSign

1.2.4 So sánh nhận xét 4 hệ thống ký kết văn bản

Bảng 1-1: So sánh các tính năng ký kết cơ bản của 4 hệ thống

Tính năng	SignNow	AdobeSign	DocuSign	VTSign
Gửi tài liệu cần ký	✓	✓	✓	✓
Mời ký số lượng lớn	✓	✓	✓	✓
Chỉnh sửa tài liệu trước khi gửi	✓	✓	✗	✗
Tạo các mẫu có thể sử dụng lại	✓	✓	✓	✓
Thêm thương hiệu được cá nhân hóa	✓	✓	✓	✗

Gửi lời mời qua liên kết	✓	✓	✓	✓
Ký kết trực tiếp	✓	✗	✗	✓
Xác định các loại chữ ký được phép	✓	✗	✗	✗
Trò chuyện trực tiếp	✓	✓	✗	✗
Bảo vệ môi đe dọa nâng cao	✓	✗	✓	✓
Quản lý tài liệu có thời hạn	✓	✗	✓	✓

Bảng 1-2: Ưu điểm và khuyết điểm của 4 hệ thống ký kết

Hệ thống	Ưu điểm	Khuyết điểm
SignNow	<p>Đăng nhập/đăng ký nhanh bằng tài khoản Facebook, Google.</p> <p>Hệ thống sử dụng eSignature đơn giản, giao diện hiện đại dễ sử dụng, LiveChat hỗ trợ khách hàng.</p> <p>Bản dùng thử không giới hạn nhiều chức năng, bản quyền giá hợp lý, phù hợp nhiều đối tượng.</p>	<p>Hỗ trợ ít ngôn ngữ, tập trung thị trường nước ngoài.</p> <p>Giao diện trình duyệt chưa có chế độ nền tối.</p> <p>Không dùng phương pháp Digital Signature, bảo mật còn hạn chế.</p>
DocuSign	<p>Sử dụng phương pháp định danh Self-Signed Digital Signature, không đòi hỏi người dùng trả tiền mua Chứng thực (Certificate).</p> <p>Sử dụng các công nghệ mã hóa và bảo mật dữ liệu, thông báo qua tin nhắn (SMS Delivery).</p> <p>Phù hợp với các công ty quy mô lớn, bảo mật chuyên môn cao.</p>	<p>Không đăng nhập/đăng ký nhanh bằng tài khoản Facebook, Google.</p> <p>Hệ thống truy cập chậm ở khu vực Việt Nam, không LiveChat.</p> <p>Không được chỉnh sửa tài liệu trước khi gửi và ký kết trực tiếp.</p> <p>Hỗ trợ ít ngôn ngữ, tập trung thị trường Mỹ, bản quyền giá cao.</p>

	Nhiều công cụ hỗ trợ kèm theo, đáp ứng nhiều tiêu chuẩn bảo mật.	
AdobeSign	<p>Sử dụng phương pháp Digital Signature, đòi hỏi người dùng trả tiền mua 1 Chứng thực.</p> <p>Đồng bộ và cá nhân hóa với phần mềm liên quan của Adobe.</p>	<p>Đăng nhập/đăng ký phức tạp, hệ thống khá chậm và tải nhiều thông tin và thông báo.</p> <p>Giao diện cũ khó nhìn, nhiều tính năng thừa và khó sử dụng, Không ký trực tiếp.</p> <p>Bản quyền phí sử dụng cao, bản dùng thử nhiều ràng buộc.</p>
VTSign	<p>Hệ thống xử lý hiển thị nhanh, không hiện thông báo gây nhiễu.</p> <p>Bảo mật tài liệu, xác thực người dùng bằng email và số điện thoại.</p> <p>Giao diện thao tác đơn giản dễ sử dụng, phù hợp nhiều lứa tuổi.</p> <p>Hỗ trợ ngôn ngữ tiếng việt, bản quyền sử dụng phù hợp người Việt</p>	<p>Không đăng nhập/đăng ký nhanh bằng tài khoản Facebook, Google.</p> <p>Không chỉnh sửa tài liệu trước khi gửi, không LiveChat, không hỗ trợ nhiều ngôn ngữ.</p> <p>Phù hợp với người dùng cá nhân, tổ chức nhỏ, không chuyên môn.</p>

1.2.5 Tình hình sử dụng chữ ký điện tử ở Việt Nam

Nafosted) sẽ áp dụng phương thức ký điện tử đối với các hợp đồng thực hiện đề tài, nhiệm vụ hoặc hoạt động do Quỹ tài trợ, hỗ trợ.

Theo Bộ phận Công nghệ thông tin của Nafosted [4]:

- Chữ ký số/con dấu số (token) là một thiết bị đã được mã hóa tất cả dữ liệu, thông tin của tổ chức/cá nhân, dùng ký thay cho chữ ký/con dấu thông thường(ký tay, ký/đóng dấu trực tiếp) trên các loại văn bản và tài liệu số thực hiện qua mạng internet hoặc các giao dịch điện tử.

- Để có chữ ký số/con dấu số, tổ chức/cá nhân cần đăng ký với các tổ chức cung cấp dịch vụ chứng thực chữ ký số được cấp phép bởi Bộ Thông tin và Truyền thông.
- Ký số/đóng dấu số là thực hiện gán chữ ký số/con dấu số (đã đăng ký với nhà cung cấp dịch vụ chữ ký số) trên văn bản, tài liệu số (việc ký hoặc đóng dấu trực tiếp, sau đó scan văn bản không được gọi là ký số/đóng dấu số).
- Thông tin chi tiết về chữ ký số và các nội dung liên quan tới văn bản ký số được quy định tại Nghị định 130/2018/NĐ-CP của Chính phủ ban hành ngày 27/9/2018 quy định chi tiết thi hành Luật Giao dịch điện tử về chữ ký số và dịch vụ chứng thực chữ ký số.

1.3 Lý do lựa chọn đề tài

Hiện nay, xã hội và công nghệ đang ngày càng phát triển không ngừng, kéo theo nhu cầu rất cao của con người hiện đại. Tần suất xử lý các công việc và tác vụ ngày càng lớn, luôn yêu cầu phải áp dụng công nghệ và số hóa mọi thứ có thể, chúng tôi muốn tạo ra một hệ thống ký kết văn bản trực tuyến trong việc quản lý thao tác thực hiện chữ ký điện tử vào các giao dịch và hợp đồng. Trong quá trình thực hiện đề tài, các thành viên nhóm cũng có cơ hội kiểm tra xem tự mình có thể thực hiện một dự án thực tế từ khi bắt đầu đến khi kết thúc hay không. Đồng thời đánh giá được khả năng sẽ hợp với vai trò nào trong một dự án thực tế giúp ích cho các dự án thương mại sau khi tốt nghiệp. Nâng cao kỹ năng giải quyết vấn đề, kỹ năng tự nghiên cứu và tìm hiểu, viết tài liệu báo cáo một cách bài bản. Đồ án tốt nghiệp này là sự tổng hợp và ứng dụng các kiến thức của nhóm tích lũy được trong suốt thời gian học tập và nghiên cứu tại trường. Quá trình thực hiện đề tài cũng hỗ trợ chúng tôi làm quen với mô hình quản lý dự án Kanban. Bên cạnh đó, các hệ thống chữ ký điện tử nổi tiếng trên thị trường rất nhiều, các chức năng thừa ít sử dụng, kèm theo tính chất kinh doanh quảng cáo sản phẩm đặc thù riêng của mỗi công ty. Tập trung nhiều vào thị trường nước ngoài, với người Việt Nam nói riêng hay người dùng cơ bản nói chung rất khó sử dụng thành thạo và nhanh chóng. Vì thế kết quả hướng tới của nhóm là tạo ra được một hệ thống đơn giản dễ sử dụng, tiếp cận được với nhiều người hơn.

1.4 Mục tiêu thực hiện

Thiết kế, xây dựng, kiểm thử, và triển khai hệ thống.

- Back-end: Cung cấp các API bao gồm: cho phép người dùng đăng ký, và thanh toán chi phí sử dụng dịch vụ, cho phép người quản trị quản lý người sử dụng dịch vụ, xử lý việc quản lý và ký kết văn bản, cung cấp các báo cáo cho admin về quá trình truy cập và báo cáo cho người dùng về quá trình truy cập, dung lượng sử dụng của từng người dùng sử dụng, dung lượng sử dụng của tài khoản. Áp dụng Microservices, mô phỏng, thiết kế, thực hiện, báo cáo kết quả các kịch bản kiểm thử khả năng tải mong muốn (load tests), khả năng chịu tải tối đa (stress tests), khả năng xâm nhập hệ thống (penetration testing).
- Front-end: Các giao diện kết nối với các tính năng tương ứng với back-end, đơn giản dễ dàng thao tác và phù hợp với nhiều đối tượng người dùng sử dụng.

Viết 80 trang Đồ án.

1.5 Yêu cầu chức năng

Nhóm đề xuất các chức năng mà một hệ thống ký kết văn bản trực tiếp cần phải đáp ứng như sau. Bao gồm người gửi, người nhận, hành động thực hiện và mô tả.

Bảng 1-3: Bảng yêu cầu chức năng đặt ra của đề tài

Đối tượng	Thao tác	Mô tả
Khách hàng xem trang web	Đăng ký	Lưu trữ thông tin cá nhân, đăng nhập vào hệ thống
Người đã có tài khoản	Đăng nhập	Tôi bắt đầu sử dụng tài khoản của mình

Người đã có tài khoản	Tạo tài liệu, tải lên một tài liệu để ký kết	Để có thể xem tài liệu và gửi tài liệu cho việc ký kết
Chủ tài liệu	Nhập thông tin người nhận	Gửi tài liệu đúng người tôi cần
Chủ tài liệu	Nhập tin nhắn riêng tư cho từng người nhận	<i>Để rồi có thể đảm bảo thông tin riêng tư cho từng người nhận.</i>
Chủ tài liệu	Cài đặt quyền cho người nhận	Có người nhận có thể ký và có người nhận chỉ được phép xem tài liệu
Chủ tài liệu	Ký tài liệu đã tải lên	Tôi ký tên vào tài liệu của mình
Chủ tài liệu	Cài đặt tài liệu	Đánh dấu đối tác của tôi cần ký và các thông tin khác
Chủ tài liệu	Nhập chủ đề, thông điệp của tài liệu	Nhận dạng tài liệu cũng như gửi tin nhắn chung cho 1 hoặc nhiều đối tác
Chủ tài liệu	Gửi lời mời ký kết cho các đối tác qua email	Để tôi có thể thông báo về tài liệu cho đối tác
Chủ tài liệu	Gửi tin nhắn chứa mã khóa (code) cho đối tác qua SMS OTP	Để bảo mật và đảm bảo rằng chỉ có đối tác của tôi có thể ký tài liệu

Người nhận tài liệu	Nhận được email có chứa đường dẫn đến tài liệu cần ký	Xem được tài liệu cần ký
Chủ tài liệu	Thông báo qua mail cho tôi khi đối tác đã xem tài liệu	Tôi biết được đối tác của mình đã xem tài liệu
Nhà đầu tư	Nhập mã Code trước khi ký tài liệu.	Việc ký kết trở nên bảo mật hơn
Người nhận tài liệu	Ký tài liệu được nhận mà không cần phải tạo tài khoản hệ thống	thuận tiện cho người dùng không cần tạo tài khoản hệ thống để ký kết.
Người nhận tài liệu	Ký tài liệu được nhận trong mục quản lý	Xem và ký các tài liệu đang chờ tôi ký
Người nhận tài liệu	Nhận được email thông báo đối tác đã ký tài liệu	Tôi biết tài liệu của tôi đã được ký
Người đã có tài khoản	Quản lý các tài liệu đã hoàn thành	Tôi có thể xem, sắp xếp, thao tác nhanh(xem chi tiết, tải tài liệu, xóa) các tài liệu đã hoàn thành
Người đã có tài khoản	Xem chi tiết tài liệu hoàn thành	Tôi có thể xem thông tin người nhận, gửi, tin nhắn, lịch sử, xóa các tài liệu đã hoàn thành

Người đã có tài khoản	Quản lý các tài liệu chờ ký	Tôi có thể xem, sắp xếp, thao tác nhanh(xem chi tiết, tải tài liệu, xóa) các tài liệu chờ ký
Người đã có tài khoản	Xem chi tiết tài liệu đang chờ ký	Tôi có thể xem thông tin người nhận, gửi, tin nhắn, trạng thái, lịch sử, xóa các tài liệu đang chờ ký.
Người đã có tài khoản	Quản lý các tài liệu cần ký	Tôi có thể xem, sắp xếp, thao tác nhanh(ký tài liệu,xem chi tiết, tải tài liệu, xóa) các tài liệu cần ký.
Người đã có tài khoản	Xem chi tiết tài liệu cần ký	Tôi có thể ký tài liệu, xem thông tin người nhận, tin nhắn, lịch sử, xóa tài liệu cần ký
Người đã có tài khoản	Quản lý các tài liệu đã xóa	Tôi có thể xem, sắp xếp, thao tác nhanh(xem chi tiết, tải tài liệu, hoàn tác) các đã xóa.
Người đã có tài khoản	Xem chi tiết tài liệu đã xóa	Tôi có thể hoàn tác, xem thông tin người nhận, tin nhắn, lịch sử.
Người đã có tài khoản	Xem tổng quát các tài liệu yêu cầu của mình về số lượng	Có cái nhìn tổng quát và có thể truy cập tới những tài liệu đó dễ dàng

Người đã có tài khoản	Đăng xuất	Để đảm bảo không ai có thể sử dụng tài khoản của tôi khi tôi không dùng đến
Người đã có tài khoản	Xem thông tin tài khoản của tôi	Để tôi có thể xem các thông tin của mình đã đăng ký với hệ thống
Người đã có tài khoản	Thay đổi thông tin tài khoản	Để thay đổi các thông tin của mình: thông tin cơ bản, mật khẩu
Người đã có tài khoản	Nạp tiền vào tài khoản theo nhiều hình thức khác nhau	Để thực hiện việc ký kết
Nhà đầu tư	Cho phép người sử dụng dùng thử dịch vụ	Để cho người dùng thử nghiệm hệ thống trước khi tiến hành nạp tiền để sử dụng dịch vụ
Người đã có tài khoản	Hiển thị số tiền còn lại của tôi	Để tôi có thể biết được số tiền còn lại trong tài khoản của mình
Người đã có tài khoản	Hiện thị lịch sử giao dịch của tôi	Để tôi có thể tra soát lại lịch sử thanh nạp và sử dụng tiền của mình

Người quản trị hệ thống	Tạo mới một tài khoản cho hệ thống	Để người quản trị có thể tạo ra một người dùng với quyền user hoặc admin
Người quản trị hệ thống	Xem danh sách tất cả người sử dụng	Để người quản trị có thể xem tổng quan danh sách người đang sử dụng hệ thống
Người quản trị hệ thống	Khóa/Mở khóa người sử dụng	Để người quản trị có thể chủ động khóa/mở khóa 1 tài khoản cụ thể trong trường hợp tài khoản đó có trực trặc hoặc các vấn đề khác liên quan
Người quản trị hệ thống	Xóa/Khôi phục người sử dụng	Để người quản trị có thể chủ động xóa/khôi phục 1 tài khoản cụ thể trong trường hợp tài khoản đó có trực trặc hoặc các vấn đề khác liên quan
Người quản trị hệ thống	Cập nhật thông tin cho từng người dùng (thay đổi quyền, thay đổi thông tin cơ bản)	Để người quản trị có thể thay đổi thông tin hoặc quyền hạn(user, admin) cho 1 tài khoản bất kỳ
Người quản trị hệ thống	Xem chi tiết người dùng (thông tin, số lượng tài liệu, quá trình thanh toán)	Để người quản trị có thể xem tổng quát thông tin và quá trình sử dụng hệ thống của người dùng bất kỳ
Người quản trị hệ thống	Thống kê lượng tiền nạp vào hệ thống theo (ngày, tuần, tháng, năm, tất cả)	Để người quản trị có thể xem tổng quát doanh thu của hệ thống theo từng mốc thời gian

Người quản trị hệ thống	Vẽ biểu đồ tổng quan về tiền nạp vào hệ thống theo (tuần, tháng, năm, tất cả)	Để người quản trị có thể so sánh tổng quát doanh thu của hệ thống theo từng mốc thời gian
Người quản trị hệ thống	Thống kê số lượng người đăng ký sử dụng hệ thống theo (ngày, tuần, tháng, năm, tất cả)	Để người quản trị có thể xem tổng quát số lượng người dùng đăng kí theo thời gian
Người quản trị hệ thống	Vẽ biểu đồ tổng quan về số lượng người đăng ký sử dụng hệ thống (tuần, tháng, năm, tất cả)	Để người quản trị có thể so sánh tổng quát số lượng người dùng đăng kí của hệ thống theo từng mốc thời gian
Người quản trị hệ thống	Thống kê số lượng tài liệu đã gửi/hoàn thành theo (ngày, tuần, tháng, năm, tất cả)	Để người quản trị có thể xem tổng quát tổng số lượng tài liệu đăng kí theo thời gian
Người quản trị hệ thống	Vẽ biểu đồ tổng quan về số lượng tài liệu đã gửi/hoàn thành theo (tuần, tháng, năm, tất cả)	Để người quản trị có thể so sánh tổng quát số lượng tài liệu của hệ thống theo từng mốc thời gian
Khách hàng xem trang web	Thông tin giới thiệu website	Tôi biết được trang web hỗ trợ mình những gì
Nhà đầu tư	Cải thiện UI/UX trang ký kết	Để giao diện đẹp hơn

Nhà đầu tư	Hiển thị trang thông báo người dùng vào mail xác nhận tài khoản đã đăng ký của họ	Người dùng biết được quá trình giải quyết vấn đề ký kết với hệ thống
Người đã có tài khoản	Lấy lại mật khẩu khi quên	Tôi có thể lấy lại mật khẩu của mình phòng trường hợp bị quên
Nhà đầu tư	Kiểm tra lại sự hợp lệ của các trường dữ liệu ở trang đăng ký	Để đảm bảo thông tin người dùng chính xác trước khi lưu vào hệ thống
Nhà đầu tư	Hiện thị lỗi cho người dùng	Để người dùng biết mình thao tác bị lỗi hay sai gì
Nhà đầu tư	Chỉnh sửa footer trang chính	Để thể hiện các thông tin liên lạc của sản phẩm cho người dùng
Nhà đầu tư	Chỉnh sửa giao diện (tất cả các trang) cho phù hợp với kích thước màn hình điện thoại, máy tính bảng	Để người dùng có thể xem trang web trên thiết bị cầm tay như điện thoại, Ipad
Người đã có tài khoản	Xem lịch sử ký kết	Để xem lại quá trình ký của tài liệu diễn ra như thế nào
Nhà đầu tư	Admin xem chi tiết user nút riêng	Xem chi tiết phải có 1 nút riêng để xem chi tiết thông tin người dùng

Người đã có tài khoản	Ký nhiều hợp đồng liên tiếp	Để có thể kí nhiều tài liệu nhiều lần liên tiếp
Người đã có tài khoản	Gửi lại liên kết kích hoạt tài khoản	Để người dùng chưa nhận được mail kích hoạt có thể yêu cầu gửi lại

1.6 Phạm vi đề tài

Các tính năng không thực hiện

- Các chức năng tự động hóa được thiết lập riêng. Thanh toán trực tuyến thông qua ngân hàng. Bản quyền và giới hạn sử dụng.
- Đồng bộ vào trên các phần mềm bên thứ ba ví dụ như Dropbox, Word, Adobe. Triển khai trên tất cả nền tảng khác.
- Chức năng phức tạp chuyên môn cao như công chứng và chống giả mạo, mã hóa tài liệu và bảo mật cao. Chức năng kiểm toán thống kê đánh giá đặc thù.
- Giao diện có nhiều tùy chọn thay đổi như quốc gia và vùng, ngôn ngữ. Tùy chọn ẩn hay hiện mục và các chức năng trên hệ thống.
- Thêm các thương hiệu logo được cá nhân hóa. Xác định các loại chữ ký được phép. Hệ thống live chat trò chuyện trực tiếp trên hệ thống.

Chương 2

LÝ THUYẾT NỀN TẢNG

2.1 Các khái niệm liên quan tới Chữ ký số Chữ ký điện tử

2.1.1 Chữ ký số - Digital signatures

Định nghĩa

- Chữ ký số là một loại của chữ ký điện tử (electronic signatures), một tập hợp các ký tự (characters) được thêm vào cuối tài liệu hoặc phần nội dung của thông điệp (message) bằng cách áp dụng các thuật toán mật hóa (cryptographic algorithms) để xác nhận hoặc thể hiện tính hợp lệ và bảo mật. Được sử dụng để xác định người đưa ra thông điệp và tính xác thực tài liệu không bị sửa đổi so với bản gốc.

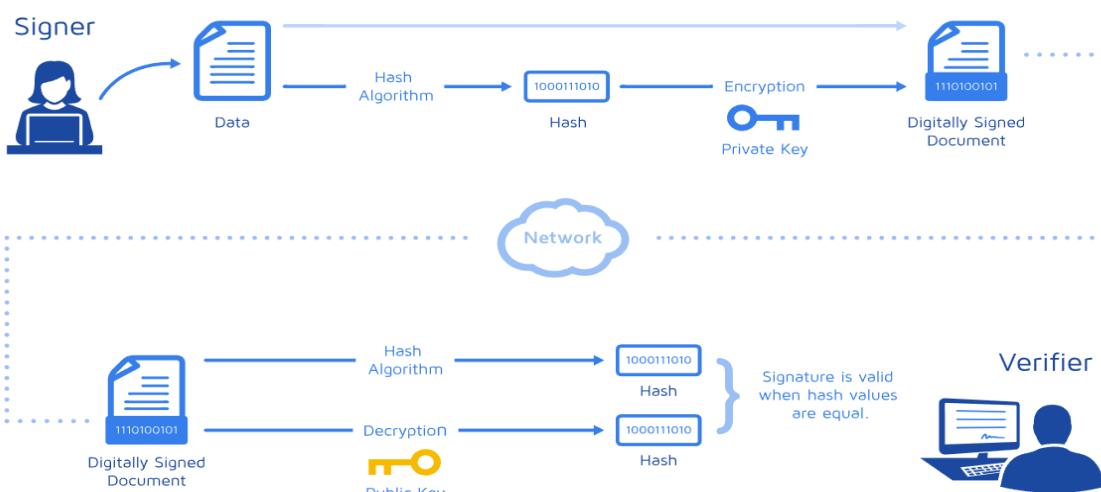
Các thành phần [5]

- Key generation: khóa công khai và khóa cá nhân tương quan của người dùng.
- Signing: tin nhắn tương ứng được ký bởi người dùng bằng khóa riêng của họ.
- Verification: chữ ký cho một thông điệp (message) được cung cấp dựa trên khóa công khai được xác minh.
- Hash Function: hàm băm đại diện cho một tập hợp các số và chữ cái được tạo ra từ một thuật toán được sử dụng bởi phần mềm chữ ký điện tử là duy nhất cho một tài liệu. Xây dựng theo một chiều (one-way), không thể đảo ngược (reversed) để tìm các tệp khác sử dụng các giá trị tương tự.
- Key cryptology: đại diện cho phương pháp mật mã (cryptographic) được sử dụng để tạo tập hợp các khóa công khai và riêng tư được liên kết với một tài liệu.
- Public Key Infrastructure (PKI): cơ sở hạ tầng khóa công khai là một tập hợp các yêu cầu cho phép tạo ra chữ ký số. Thông qua PKI, mỗi giao dịch chữ ký số bao gồm một cặp khóa: khóa riêng tư và khóa công khai. Khóa riêng tư không được chia sẻ và chỉ được sử dụng bởi người ký để ký điện tử vào các tài liệu. Khóa công khai có sẵn và được sử dụng bởi những người cần xác thực chữ ký điện tử của người ký.

PKI thực thi các yêu cầu bổ sung, như Tổ chức phát hành chứng chỉ (Certificate Authority - CA), chứng chỉ số, phần mềm đăng ký người dùng cuối và các công cụ để quản lý, gia hạn và thu hồi khóa và chứng chỉ [6].

- Certificate Authority (CA): chữ ký số dựa trên các khóa công khai và riêng tư. Các khóa đó phải được bảo vệ để đảm bảo an toàn và tránh bị giả mạo hoặc sử dụng với mục đích xấu. Khi gửi hoặc ký một tài liệu, cần đảm bảo rằng tài liệu và khóa được tạo một cách an toàn và đang sử dụng khóa hợp lệ. Nhà cung cấp dịch vụ tin cậy CA là các tổ chức bên thứ ba đã được chấp nhận rộng rãi là đáng tin cậy để đảm bảo bảo mật khóa và có thể cung cấp các chứng chỉ kỹ thuật số cần thiết. Cả người gửi và người nhận ký tài liệu phải đồng ý sử dụng một CA nhất định.

Cách thức hoạt động



Hình 2-1: Cách thức hoạt động của Chữ ký số

- Bước 1: Hệ thống tạo ra cặp chìa khóa (key pair) cho từng tài khoản của người dùng.
- Bước 2: Người gửi sẽ ký và tải (upload) tài liệu lên hệ thống.
- Bước 3: Sử dụng thuật toán để mã hóa tài liệu thành bản tóm tắt (digest).
- Bước 4: Hệ thống tiếp tục dùng bản tóm tắt (digest) và khóa riêng tư (private key) của người gửi để tiến hành tạo thành chữ ký số (digital signature).
- Bước 5: Người nhận xác nhận danh tính với hệ thống để nhận được tài liệu + khóa công khai (public key) + chữ ký số (digital signature) của người gửi.

- Bước 6: Hệ thống giải mã chữ ký điện tử của người nhận cung cấp bằng khóa công khai (public key) → bản tóm tắt (digest) 1 (nếu không giải mã được thì chữ ký điện tử này là giả mạo).
- Bước 7: Hệ thống cũng mã hóa tài liệu của người gửi → bản tóm tắt (digest) 2.
- Bước 8: Hệ thống so sánh 2 bản tóm tắt (digest) 1 và 2 để chứng thực.

2.1.2 Chữ ký điện tử - Electronic Signatures

- Chữ ký điện tử là một phần dữ liệu để cập đến dữ liệu điện tử khác và được sử dụng để xác minh người ký tài liệu, rằng danh tính của người ký đã được xác minh và tài liệu đó không thay đổi sau khi chữ ký được thêm vào. Các phương pháp khác nhau để ghi lại chữ ký gồm nhập tên của người ký vào khung chữ ký bằng cách sử dụng máy tính hoặc ứng dụng di động để chụp ảnh chữ ký viết tay, xác minh qua email, bằng ID công ty hoặc mã PIN điện thoại [7].

2.1.3 Sự khác biệt giữa chữ ký số và chữ ký điện tử

Bảng 2-1: So sánh chữ ký điện tử và chữ ký số

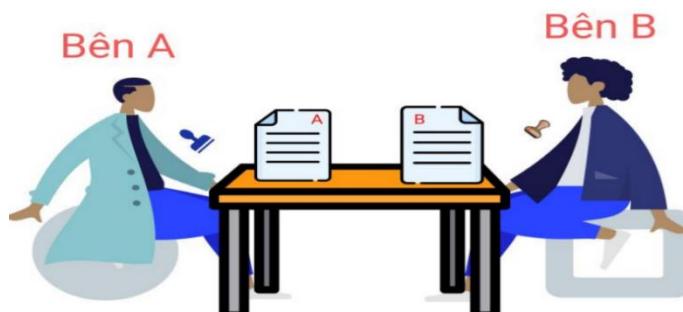
	Chữ ký số	Chữ ký điện tử
Tính chất	Được mã hóa và chứng thực danh tính người ký.	Biểu tượng, hình ảnh, được đính kèm với tin nhắn hoặc tài liệu chứng thực danh tính của người ký và hành động.
Tính năng	Bảo mật một tài liệu	Xác minh một tài liệu
Tiêu chuẩn	Sử dụng các phương thức mã hóa mật mã dựa trên PKI và đảm bảo danh tính người ký, tính toàn vẹn dữ liệu.	Không phụ thuộc vào các tiêu chuẩn. Không sử dụng mã hóa.
Xác thực	Kỹ thuật số dựa trên chứng chỉ.	Xác thực danh tính người ký thông qua email, mã pin

Bảo mật	An toàn bảo mật cao, khó giả mạo hoặc thay đổi.	Chữ ký dễ bị giả mạo.
Độc quyền	Được xác nhận bởi bất kỳ người nào mà không cần phần mềm xác minh độc quyền.	Chữ ký điện tử không được ràng buộc về mặt pháp lý và sẽ yêu cầu phần mềm độc quyền để xác nhận chữ ký điện tử.
Xác nhận	Được xác nhận bởi các cơ quan chứng nhận hoặc nhà cung cấp dịch vụ tin cậy.	Không có quy trình xác nhận cụ thể rõ ràng nào.

2.2 Ký kết văn bản trong đời sống thực

2.2.1 Quy trình ký kết

Bước 1: Bên A và bên B đều được chuẩn bị 1 văn bản / hợp đồng cho mỗi bên, cùng với con dấu (nếu có). Cả 2 sẽ hẹn một ngày gặp mặt để ký kết văn bản / hợp đồng.



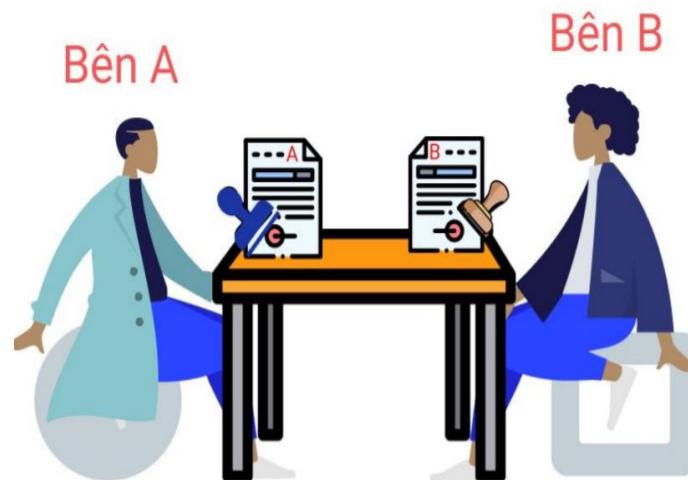
Hình 2-2: Quy trình ký kết văn bản đời sống thực – Bước 1

Bước 2: Bên A và bên B ký lên văn bản / hợp đồng của mình.



Hình 2-3: Quy trình ký kết văn bản đời sống thực – Bước 2

Bước 3: Bên A và bên B đóng dấu lên văn bản / hợp đồng của mình.



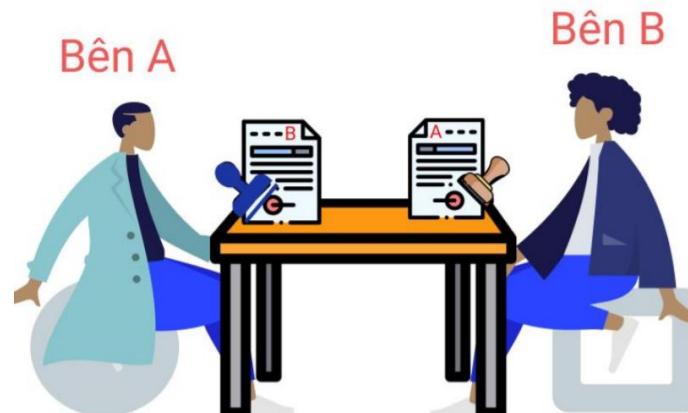
Hình 2-4: Quy trình ký kết văn bản đòn sống thực – Bước 3

Bước 4: Bên A và bên B ký tên lên văn bản / hợp đồng của đối tác.



Hình 2-5: Quy trình ký kết văn bản đòn sống thực – Bước 4

Bước 5: Bên A và bên B đóng dấu lên văn bản / hợp đồng của đối tác.



Hình 2-6: Quy trình ký kết văn bản đòn sống thực – Bước 5

Bước 6: Văn bản / hợp đồng sẽ được mang đến văn phòng công chức và 1 số giấy tờ theo yêu cầu. Công chứng viên kiểm tra xem đã đầy đủ giấy tờ theo yêu cầu hay chưa.



Hình 2-7: Quy trình ký kết văn bản đòi sống thực – Bước 6

Bước 7: Công chứng viên sẽ đóng dấu lên văn bản / hợp đồng đó để hoàn tất. Văn bản / hợp đồng này đã có hiệu lực pháp lý.



Hình 2-8: Quy trình ký kết văn bản đòi sống thực – Bước 7

2.2.2 Các vấn đề thường gặp

- Vị trí địa lý của bên A và bên B xa nhau, gây bất lợi cho 2 bên
- Thời gian ký kết và công chứng trong toàn bộ quá trình kéo dài, phức tạp
- Nhiều tài liệu giấy phải in và đem theo, bất tiện trong việc di chuyển và lưu trữ
- Bên A hoặc bên B hay cả 2 bên huỷ hoặc soạn lại văn bản / hợp đồng
- Bên A hoặc bên B không thừa nhận việc ký kết, không làm theo văn bản / hợp đồng

2.2.3 Các phương pháp xử lý

- Thông nhất địa điểm thuận tiện cho cả 2 bên, đặt lịch hẹn trước với phòng công chứng hoặc làm việc với công chứng viên trước, dùng người đại diện như luật sư để thay thế xử lý các vấn đề.
- Nên có phương án dự phòng như văn bản / hợp đồng thay thế. Chuẩn bị tài liệu đầy đủ và lưu trữ khoa học với bộ phận phụ trách.
- Trường hợp 1 trong 2 bên không thừa nhận việc ký kết, có thể dùng hợp đồng có chữ ký đôi bên và công chứng làm bằng chứng pháp lý trước toà. Bên A hoặc bên B không thể phủ nhận việc ký kết của mình.

2.3 Ký kết văn bản bằng máy tính

2.3.1 Các vấn đề thường gặp

- Bên A hoặc bên B không biết sử dụng hệ thống ký kết, không biết sử dụng máy tính, không chấp nhận việc ký kết văn bản bằng máy tính. Các sự cố không mong muốn bao gồm: đường truyền dữ liệu, kết nối mạng của người sử dụng.
- Các hệ thống trong quy trình bị tấn công hoặc chiếm đoạt bao gồm: hệ thống ký kết X (chính hệ thống hoặc tài khoản người dùng), tài khoản hộp thư (email) của bên A hoặc bên B, đường dẫn (link) trong hộp thư.
- Bên A hoặc bên B phủ nhận việc ký kết văn bản / hợp đồng. không thừa nhận ký kết, không làm theo văn bản / hợp đồng, không chịu pháp lý trước toà.

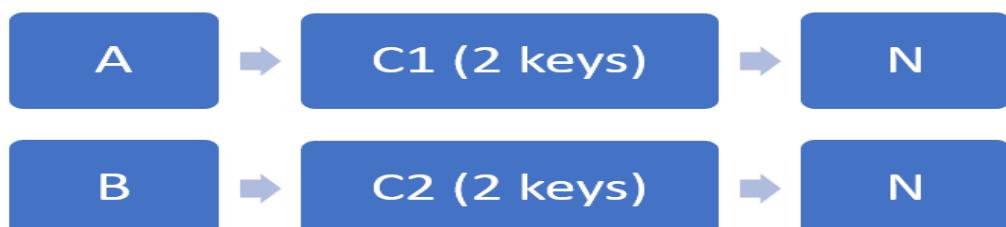
2.3.2 Các phương pháp xử lý

- Xử lý bằng cách đào tạo thực hành bởi công ty tổ chức. Hệ thống ký kết văn bản trên thị trường đều có hướng dẫn sử dụng và thao tác đơn giản dễ tiếp cận.
- Xử lý bằng cách lựa chọn hệ thống ký kết tin cậy và phổ biến trên thị trường. Áp dụng phương pháp Chữ ký số hoặc Chữ ký điện tử.
- Xác thực người dùng bằng phương pháp Digital Signature, định danh người dùng bằng phương pháp định danh, bên A hoặc bên B không thể phủ nhận việc ký kết.

2.4 Phương pháp ký kết bằng Digital Signature

2.4.1 Quy trình ký kết

- Các đối tượng trong quy trình gồm: Người gửi (A), người nhận (B), bên cấp chứng thực (N), tài liệu gửi đi (PDF), tài liệu gốc (D), chứng thực (C), hệ thống ký kết (X). Yêu cầu mỗi người dùng phải trả tiền mua một chứng thực (C) từ một bên cung cấp chứng thực (N) tin cậy.
- Bước 1: bên A và bên B thông nhất sử dụng chung một bên cấp chứng thực N, mỗi bên sẽ nhận được chứng thực C1 và C2 gồm 2 khoá (keys) kèm theo.



Hình 2-9: Quy trình ký kết văn bản bằng Digital Signature – Bước 1

- Bước 2: bên A gửi văn bản / hợp đồng gồm chứng thực C1 và tài liệu gốc D sử dụng trên hệ thống ký kết X. Hệ thống X sẽ đính kèm thành 1 tập tin tài liệu PDF1 gửi tới hộp thư (email) của B.



Hình 2-10: Quy trình ký kết văn bản bằng Digital Signature – Bước 2

- Bước 3: bên B nhận được tài liệu PDF1 của bên A gửi, bên B ký kết lên tập tin PDF1 với chứng thực C2. Hệ thống ký kết X sẽ đính kèm thành 1 tập tin tài liệu PDF2 rồi gửi kết quả ký kết tới hộp thư (email) của B. Hoàn tất quá trình.



Hình 2-11: Quy trình ký kết văn bản bằng Digital Signature – Bước 3

2.4.2 Xác thực người ký

- Bên A hoặc bên B phủ nhận việc ký kết. Sử dụng chứng thực mà bên cung cấp chứng thực đã đưa cho cả A và B. Mục đích định danh xác thực người ký tài liệu là ai, không thể phủ nhận việc ký kết.

- Bên cung cấp chứng thực N dựa vào Khoá công khai (public key) của A để xác định tài liệu sử dụng (PDF1). Từ đó dựa vào bên A mới có Khoá bí mật (private key) để sử dụng làm bằng chứng pháp lý.



Hình 2-12: Quy trình ký kết văn bản bằng Digital Signature – Xác thực

2.4.3 Ưu điểm và khuyết điểm

Ưu điểm

Độ tin cậy bảo mật an toàn cao, chứng thực và pháp lý không phụ thuộc hoàn toàn vào hệ thống ký kết sử dụng. Định danh xác thực người ký kết và mặt pháp lý đều dựa vào bên cung cấp chứng thực tin cậy. Loại bỏ việc phải gửi chứng thực của người dùng cho từng đối tác thương mại.

Khuyết điểm

Chứng thực phải được mua từ nhà cung cấp bên thứ ba, dẫn đến tốn nhiều chi phí. Khi xác minh chữ ký đối tác, người dùng phải tải xuống chứng thực có chữ ký của bên cung cấp, để xác minh chữ ký số khoá công khai của đối tác.

2.5 Phương pháp ký kết bằng cách định danh

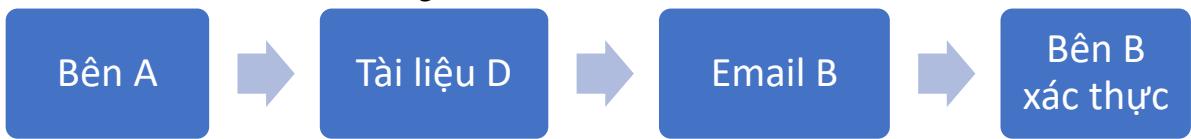
2.5.1 Quy trình ký kết

- Các đối tượng trong quy trình gồm: Người gửi (A), người nhận (B), tài liệu gửi đi (D), hệ thống ký kết (X). Không yêu cầu mỗi người dùng phải trả tiền mua một chứng thực từ một bên cung cấp chứng thực tin cậy.
- Bước 1: A đăng nhập vào hệ thống X bằng tài khoản đã đăng ký, gồm email và password. Hệ thống X xác thực người dùng A thông qua tài khoản và ID. A tải tài liệu lên và ký kết.



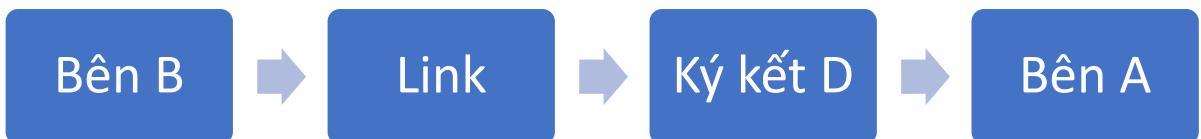
Hình 2-13: Quy trình ký kết văn bản bằng định danh – Bước 1

- Bước 2: A gửi tài liệu D đến email của B. Gồm đường dẫn ký kết (link), email và ID của B. B xác nhận bằng SMS OTP,



Hình 2-14: Quy trình ký kết văn bản bằng định danh – Bước 2

- Bước 3: B truy cập đường dẫn ký kết (link) để ký kết tài liệu. B có thể ký kết trực tiếp bằng đường dẫn (link) hoặc đăng nhập vào hệ thống X (nếu đã đăng ký).



Hình 2-15: Quy trình ký kết văn bản bằng định danh – Bước 3

2.5.2 Định danh người ký

- Bên A hoặc bên B phủ nhận việc ký kết, định danh người ký bằng nhiều phương pháp, dựa vào hệ thống ký kết sử dụng có thể sử dụng thêm phương pháp SMS OTP.
- Trường hợp cả A và B đều đăng nhập sử dụng tài khoản trên hệ thống ký kết. Thông tin đăng nhập tài khoản hộp thư gồm địa chỉ (email) và mật khẩu (password) của cá nhân ký kết. Chỉ có người chủ tài khoản mới có thông tin để đăng nhập.
- Trường hợp bên B nhận tài liệu và ký kết trực tiếp (không yêu cầu bên B có tài khoản đăng nhập vào hệ thống), chỉ duy nhất bên B mới có được đường dẫn (link) từ hộp thư (email) và mã xác thực SMS OTP từ điện thoại cá nhân.

2.5.3 Ưu điểm và khuyết điểm

Ưu điểm

Không tốn thêm chi phí trong việc mua chứng thực bên cung cấp thứ ba. Mỗi hệ thống ký kết tự tạo CA và tự xác thực. Hiệu quả cho một số lượng nhỏ các đối tác thương mại

Khuyết điểm

Yêu cầu người dùng phân phối chứng chỉ của mình, trừ đi khóa cá nhân, cho từng đối tác thương mại một cách an toàn. Khó bảo trì, bất cứ khi nào chứng chỉ được thay đổi, nó phải được phân phối cho tất cả các máy khách. Không được xác thực bởi bên thứ ba. Không hiệu quả với số lượng lớn đối tác thương mại.

Chương 3

GIẢI PHÁP ĐỀ TÀI

3.1 Giải pháp tổng quát

3.1.1 Giải pháp ký kết

Ký kết văn bản là chức năng quan trọng và cốt lõi nhất trong hệ thống. Có nhiều phương pháp ký kết văn bản đã được nhóm đề cập ở chương 2, ở đây nhóm chọn giải pháp ký kết bằng phương pháp định danh.

3.1.1.1 Quy trình chung của việc gửi tài liệu

- Người gửi đăng nhập vào hệ thống và chọn chức năng ký kết
- Người gửi tải lên tài liệu cần ký kết. Ở đây người gửi có thể chọn một hoặc nhiều tài liệu nếu cần thiết
 - Người gửi nhập thông tin người nhận (có thể nhập một hoặc nhiều người nhận) bao gồm:
 - + Họ tên của người nhận
 - + Địa chỉ email của người nhận. Ở bước này hệ thống sẽ tiến hành kiểm tra xem
 - + Người nhận có tài khoản: Không cần nhập số điện thoại
 - + Người nhận chưa có tài khoản: Cần cung cấp số điện thoại

Mục đích là để hệ thống bảo mật thêm một bước để đảm bảo rằng chỉ có duy nhất người nhận mới có thể truy cập vào tài liệu để ký kết (gồm liên kết được gửi qua email và mã code được gửi qua tin nhắn SMS).

- + Chỉ định quyền cho người nhận: Có 2 lựa chọn là chỉ xem tài liệu chứ không được phép ký lên tài liệu (chỉ đọc) và ký lên tài liệu theo các trường mà người gửi chỉ định (đọc và ký)
- + Mã khóa: Hệ thống sẽ tự phát sinh một mã khóa bảo mật để khóa tài liệu, người gửi có thể thay đổi khóa này

- + Tin nhắn riêng tư: Nội dung tin nhắn riêng tư mà người gửi muốn gửi cho người nhận được chỉ định
 - Người gửi ký lên tài liệu đồng thời chỉ định vị trí cần ký kết cho từng người nhận có quyền là “đọc và ký”
 - Người gửi nhập các thông tin: tiêu đề tài liệu, nội dung của tài liệu để hoàn tất quá trình gửi
 - Hệ thống sẽ tiến hành gửi email mời ký tài liệu cho từng người nhận đồng thời gửi mã bảo mật để mở khóa tài liệu đến số điện thoại người nhận

3.1.1.2 Quy trình chung của việc ký tài liệu

- Người nhận nhán vào liên kết trong email chứa liên kết đến tài liệu cần ký. Ở bước này nếu người nhận đã có tài khoản hệ thống thì chỉ việc đăng nhập vào hệ thống và chọn vào mục hiển thị các tài liệu chờ ký và có thể nhấn vào để ký
- Người nhận nhập mã bảo mật để mở khóa tài liệu (nhận được từ SMS)
- Hệ thống tiến hành kiểm tra và trả về tài liệu cho người dùng để ký kết. Nếu mã không chính xác hệ thống sẽ thông báo lỗi và kết thúc quá trình
- Người nhận ký lên các trường mà người gửi đã chỉ định và chọn hoàn tất
- Hệ thống gửi email thông báo cho người gửi rằng người nhận đã ký vào tài liệu
- Nếu người nhận là người cuối cùng ký vào tài liệu. Hệ thống gửi email chứa tài liệu đã hoàn thành tới người gửi và các người nhận
- Người gửi và các người nhận có thể tải về các tài liệu của mình và kết thúc quá trình

3.1.2 Giải pháp quản lý tài liệu

- Sau khi thực hiện ký kết, các tài liệu cần được quản lý và lưu trữ để người dùng có thể sử dụng và đối chiếu sau này. Vì tài liệu của dùng rất quan trọng nên việc lựa chọn nơi để lưu trữ liệu cần phải được chọn lọc để tránh việc đánh mất dữ liệu gây ra hậu quả nghiêm trọng. Việc lựa chọn nơi lưu trữ dữ liệu dựa trên các tiêu chí chính sau: tính bảo mật, tính linh hoạt, khả năng khôi phục dữ liệu nếu có lỗi, khả năng kiểm soát và mức độ thân thiện.

- Với các tiêu chí trên, nhóm quyết định chọn Cloud Storage làm nơi lưu trữ tài liệu chính của ứng dụng. Lưu trữ đám mây (Cloud storage) là hình thức sử dụng dịch vụ của nhà cung cấp dịch vụ cho phép người dùng dịch vụ của họ có thể lưu giữ, quản lý, chia sẻ và backup dữ liệu của họ từ xa. Hiện nay có rất nhiều dịch vụ nhà cung cấp dịch vụ Cloud Storage như Amazon Web Service, oracle, Google Cloud Platform,... Tuy nhiên, nhóm nhận thấy Microsoft Azure có phù hợp với hệ thống với các tính năng sau:

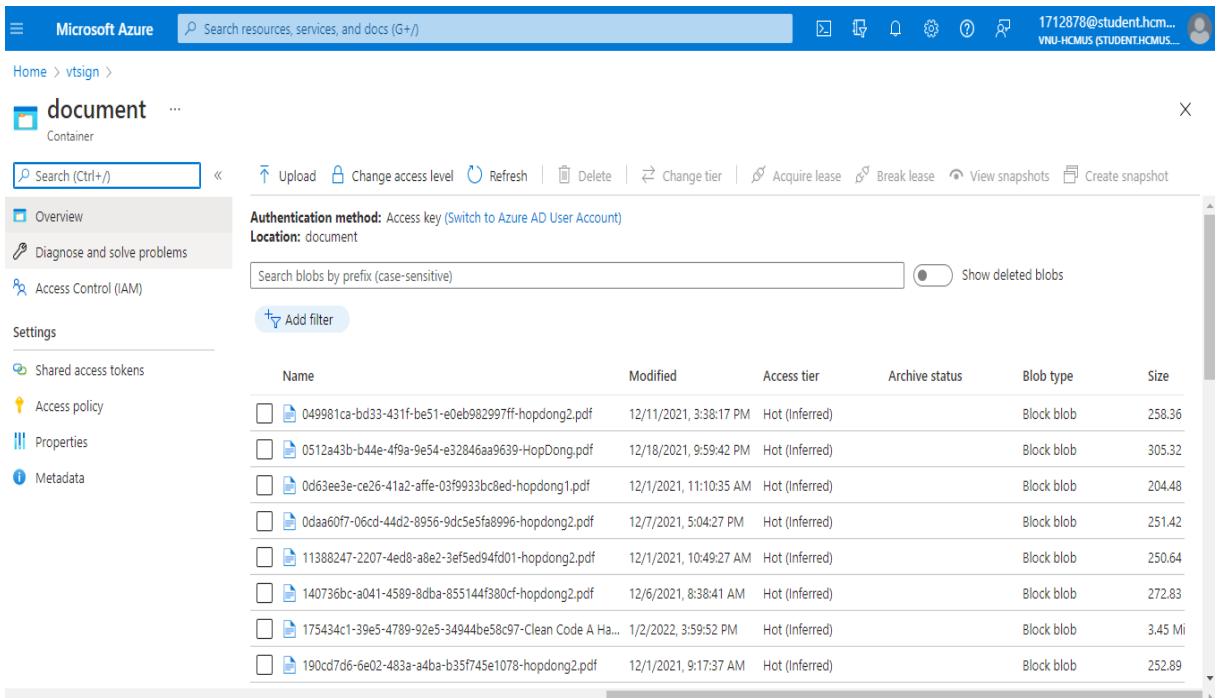
Tính linh hoạt: sử dụng azure storage có thể dễ dàng mở rộng và thu hẹp phù hợp với từng giai đoạn của hệ thống. Trong trường hợp hệ thống có lượng người dùng lớn, chúng ta có thể mở rộng không gian lưu trữ tài liệu của hệ thống, điều này mang lại tính linh hoạt cho hệ thống.

Tính an toàn: Đảm bảo dữ liệu của người dùng an toàn trong trường hợp có sự cố phần cứng tạm thời. Người dùng có thể sao lưu dữ liệu trên Azure Storage sau khi khắc phục xong sự cố và tốc độ truy xuất dữ liệu cao.

Tính bảo mật: tất cả dữ liệu được ghi vào Azure Storage đều được mã hóa và cung cấp thêm tính năng phát hiện truy cập trái phép.

Name	Last modified	Public access level	Lease state
\$logs	10/30/2021, 4:28:34 PM	Private	Available
document	11/23/2021, 4:39:27 PM	Blob	Available
document-dev	12/11/2021, 10:55:04 AM	Blob	Available
user	12/11/2021, 10:52:44 AM	Blob	Available
user-dev	12/11/2021, 10:52:57 AM	Blob	Available

Hình 3-1: Giải pháp quản lý tài liệu – Phần 1



Hình 3-2: Giải pháp quản lý tài liệu – Phần 2

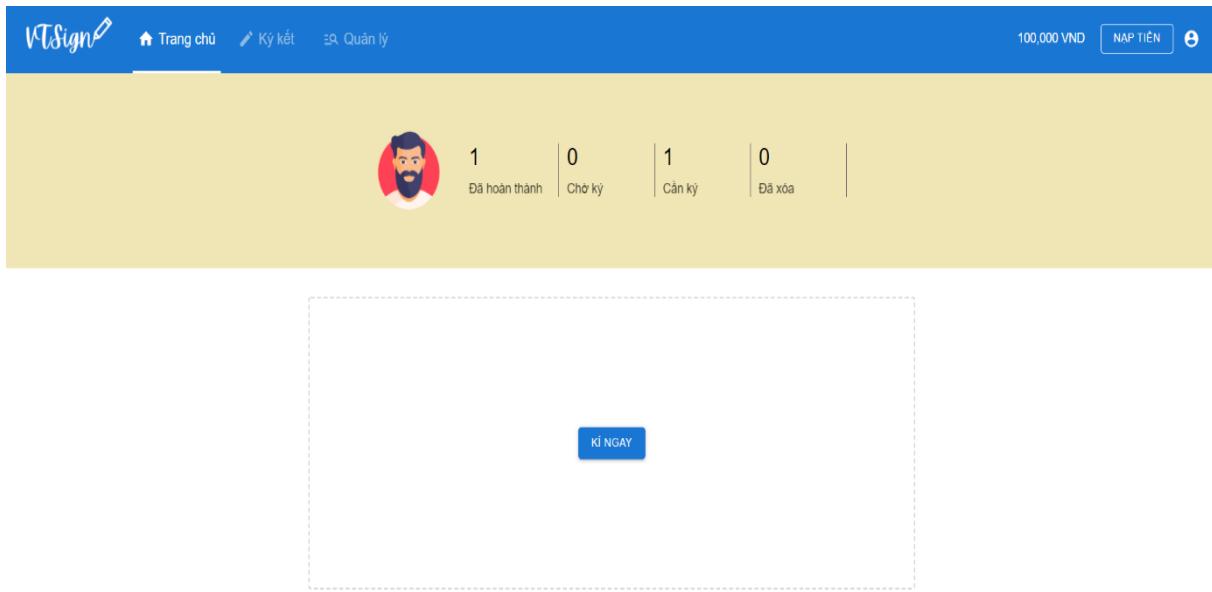
3.2 Thiết kế hệ thống

3.2.1 Thiết kế giao diện hệ thống

Giao diện hệ thống đóng góp một phần quan trọng quyết định lượng người dùng sử dụng dịch vụ. Vì vậy nhóm đã tập trung thiết kế giao diện sao cho trải nghiệm của người dùng là tốt nhất. Với độ ưu tiên là hiển thị thông tin, dễ sử dụng và thao tác đơn giản. Dưới đây sẽ liệt kê một số giao diện chính của hệ thống.

3.2.1.1 Giao diện trang chủ hệ thống

Trang chủ là nơi người dùng được điều hướng sau khi đã xem trang giới thiệu và đăng nhập thành công. Tại đây, người dùng thấy được toàn bộ chức năng của hệ thống, có thể điều hướng trực tiếp qua các trang khác. Tiêu đề có Trang chủ, Ký kết, Quản lý, Nạp tiền và quản lý tài khoản. Bên dưới tiêu đề hiện thị tóm tắt trang thái quản lý tài liệu của tài khoản như đã hoàn thành, chờ ký, cần ký và đã xóa. Cuối cùng là nút chức năng chính Ký ngay dùng để điều hướng qua trang Ký kết.

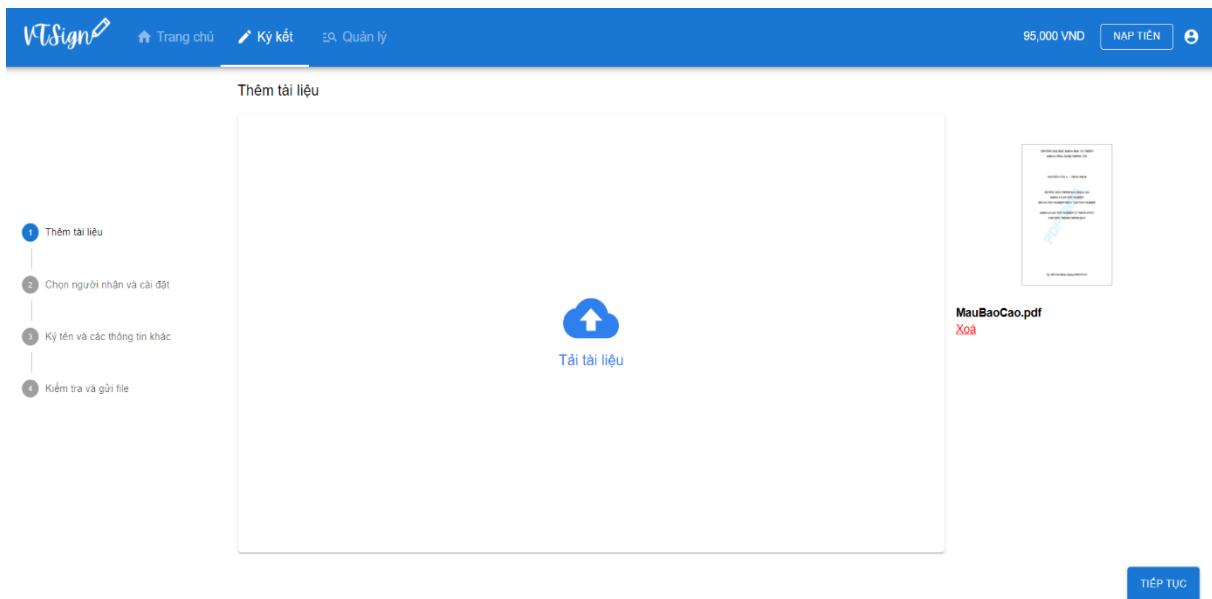


Hình 3-3: Giao diện trang chủ hệ thống

3.2.1.2 Giao diện trang ký kết

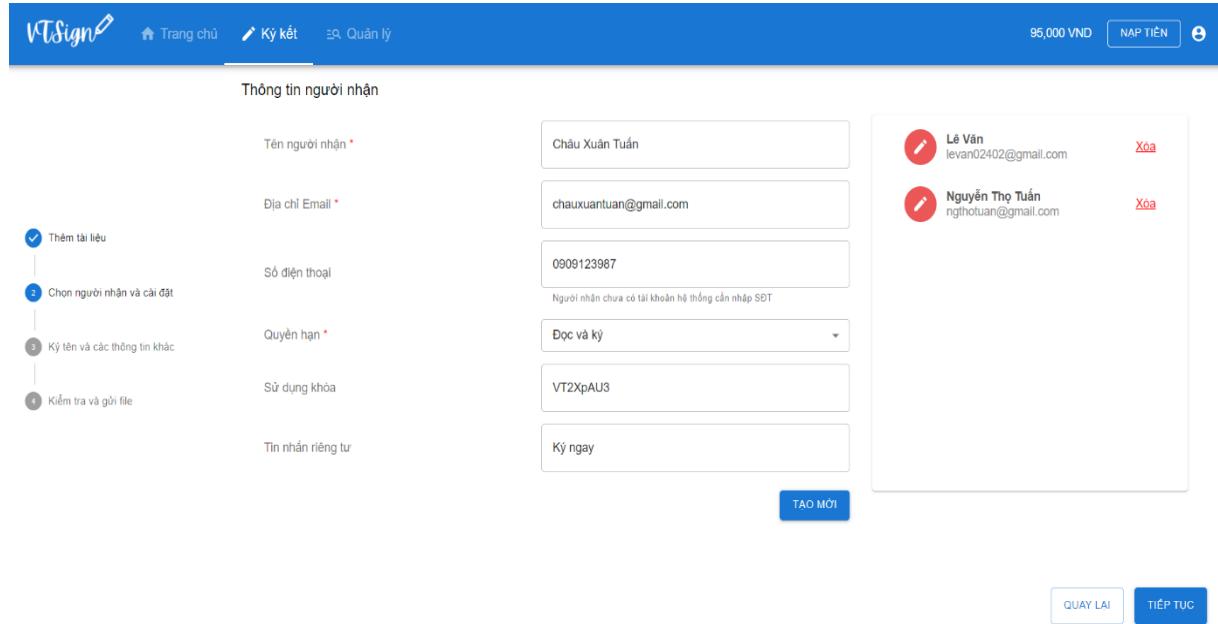
Ngay trung tâm trang Ký kết sẽ là khung thêm tài liệu có nút tải tài liệu để chọn tập tin từ máy tính hoặc có thể kéo trực tiếp tập tin vào khung để thực hiện tải. Bên trái hiển thị quá trình ký kết 4 bước

- Thêm tài liệu: người dùng thực hiện tải tài liệu theo định dạng PDF theo 2 cách chọn trực tiếp tập tin hoặc kéo tập tin.



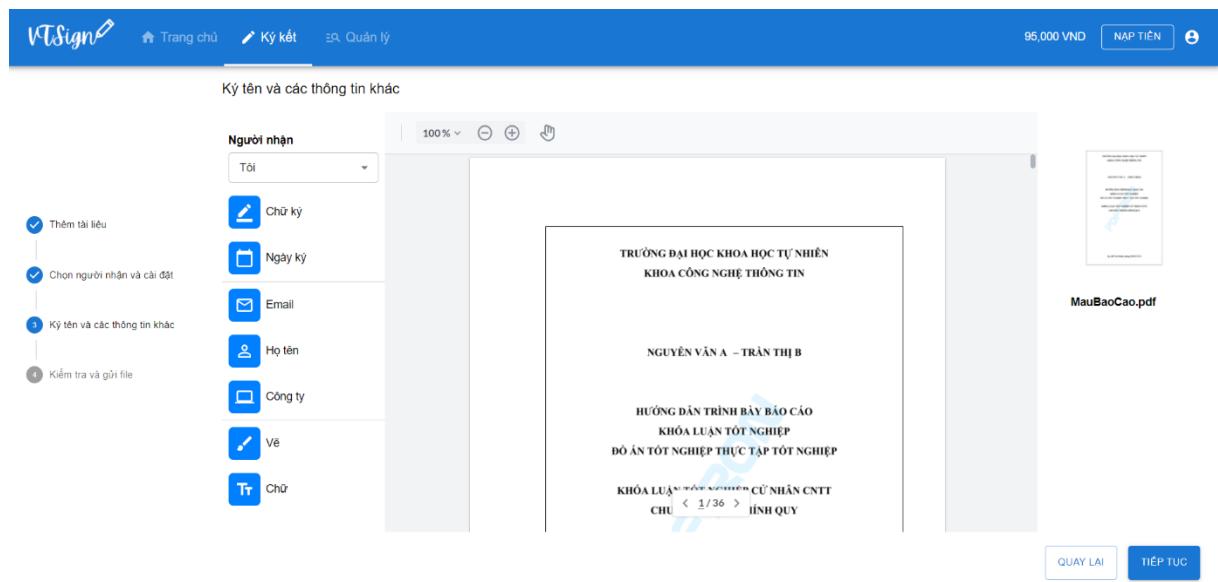
Hình 3-4: Giao diện trang ký kết – Thêm tài liệu

- Chọn người nhận và cài đặt: điền các thông tin của người nhận gồm 3 trường bắt buộc là tên người nhận, địa chỉ email, quyền hạn. Có thể gửi cho nhiều người nhận, người chưa có tài khoản hệ thống cần nhập thêm số điện thoại.



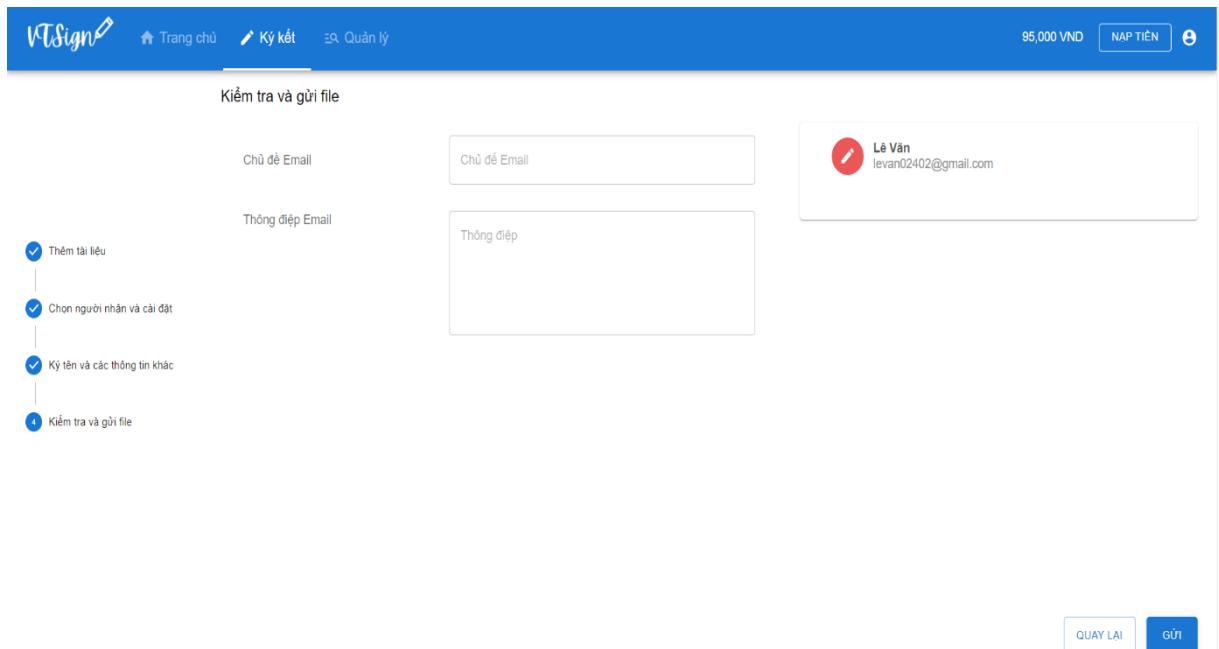
Hình 3-5: Giao diện trang ký kết – Chọn người nhận và cài đặt

- Ký tên và các thông tin khác: thiết lập gửi cho chính mình hoặc người nhận. Thực hiện kéo các trường vào tập tin PDF như chữ ký, ngày ký, thương hiệu... Đôi với chữ ký có thể vẽ trực tiếp hoặc chọn kiểu, có thể tải lên chữ ký cá nhân lưu ở máy tính.



Hình 3-6: Giao diện trang ký kết – Ký tên và thông tin khác

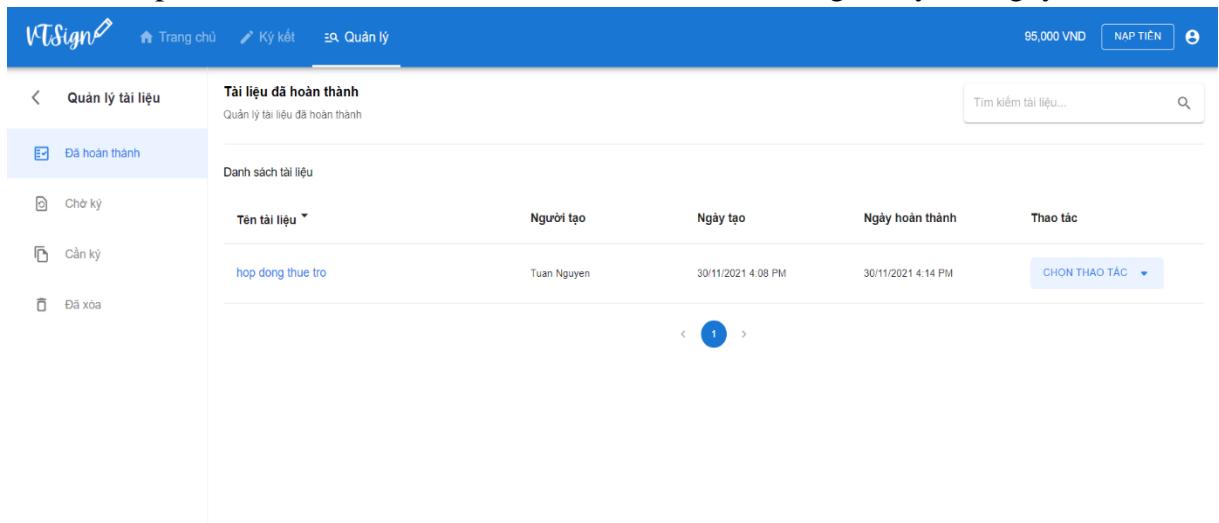
- Kiểm tra và gửi file: bước này người dùng điền 2 thông tin gồm chủ đề email và thông điệp của email để gửi cho người nhận. Quá trình hoàn tất.



Hình 3-7: Giao diện trang ký kết – Kiểm tra và gửi file

3.2.1.3 Giao diện trang quản lý tài liệu

Tại đây người dùng có thể quản lý mọi tài liệu của mình, có thể thao tác trực tiếp tất cả tài liệu, quản lý khoa học và nhanh chóng với trạng thái của tài khoản. Gồm các trạng thái như đã hoàn thành, chờ ký, cần ký và đã xóa. Mỗi mục đều hiển thị danh sách tài liệu, có thể sắp xếp theo tên hoặc thời gian ngày tạo và hoàn thành. Bên cạnh đó có thể thao tác trực tiếp cho mỗi tài liệu như xem chi tiết, xóa, tải xuống và ký kết ngay.

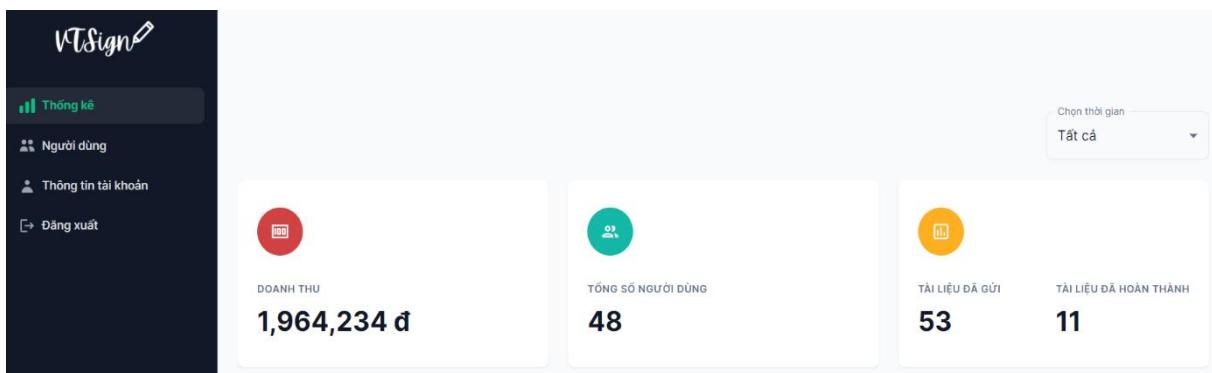


Hình 3-8: Giao diện trang quản lý tài liệu

3.2.1.4 Giao diện trang quản trị thống kê

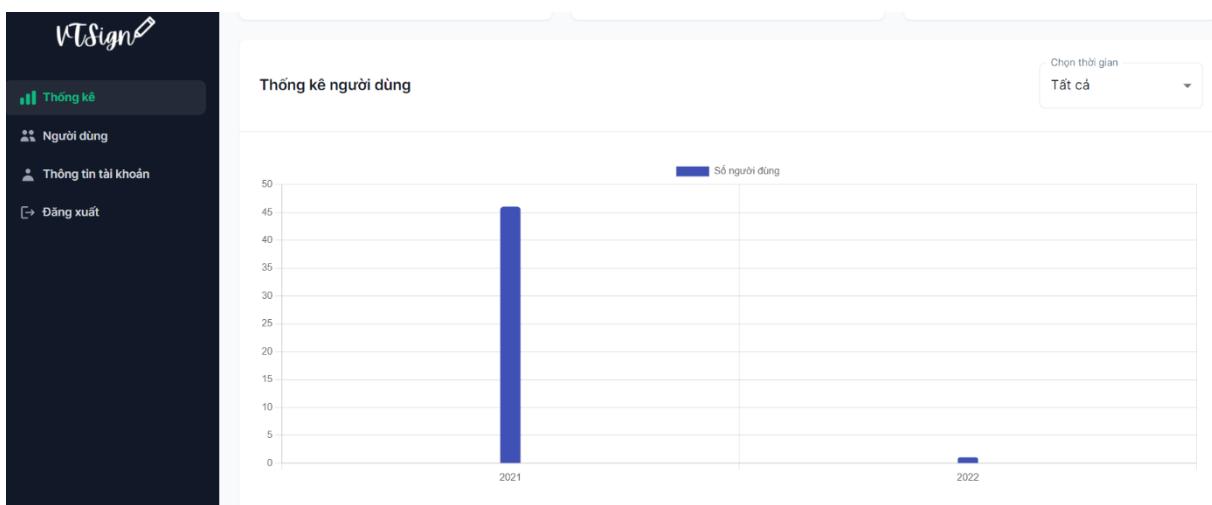
Trang quản trị thống kê báo cáo của hệ thống VTSign giúp cho người quản trị nắm rõ được doanh thu theo tuần, tháng và năm đối với thống kê người dùng, tiền và tài liệu. Giúp cho các quản trị hệ thống có thể xem xét đánh giá về phương hướng phát triển trong tương lai hoặc khắc phục. Mục thống kê gồm 4 thành phần chính:

- Hiển thị trực quan tóm tắt trạng thái con số cụ thể về doanh thu, tổng số người dùng, tài liệu đã gửi và tài liệu hoàn thành



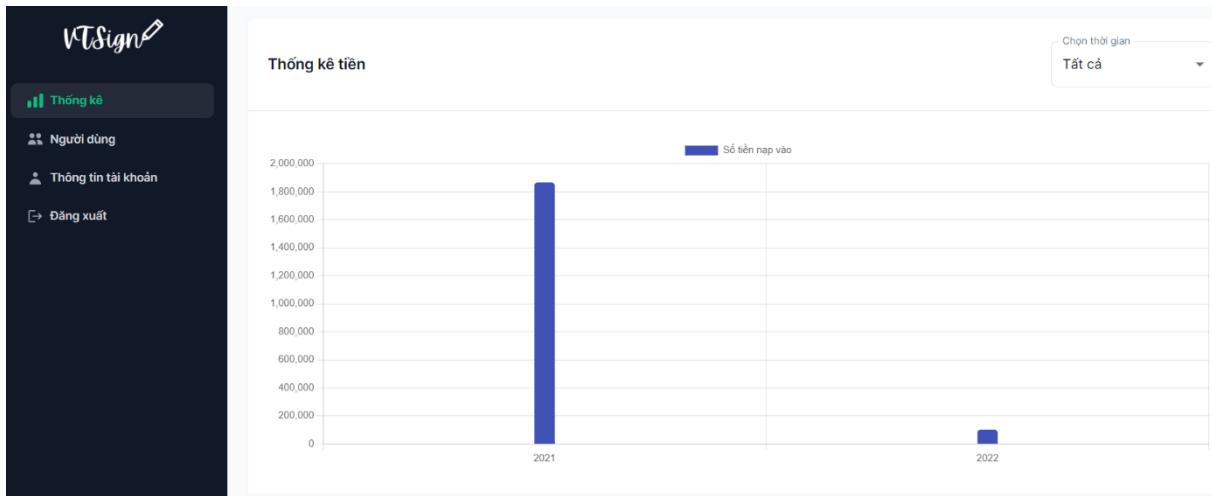
Hình 3-9: Giao diện trang quản trị - Thống kê phần 1

- Thống kê người dùng



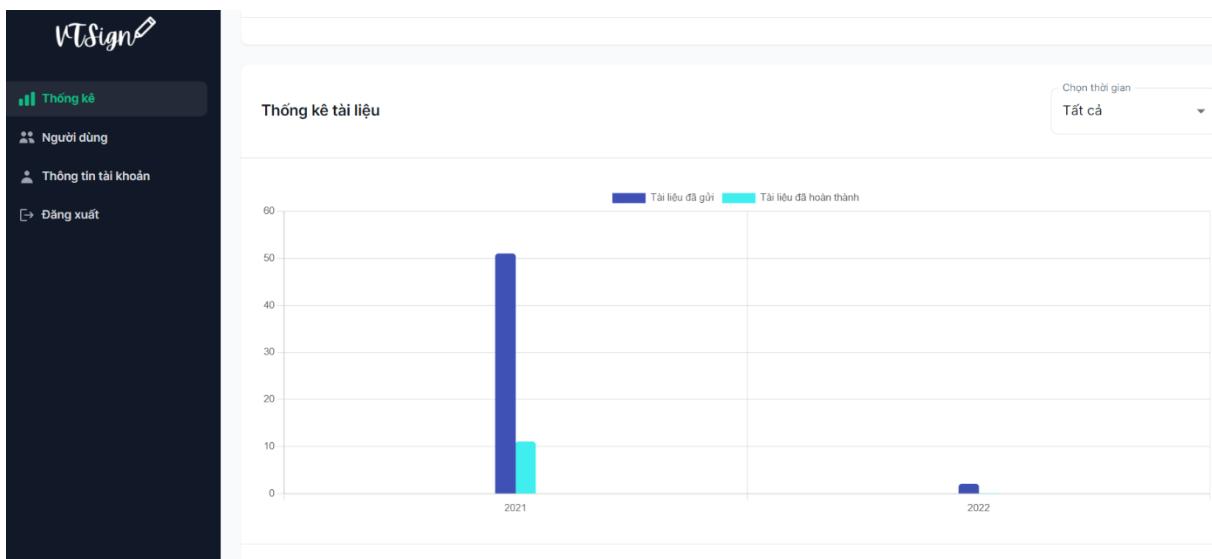
Hình 3-10: Giao diện trang quản trị - Thống kê phần 2

- Thống kê tiền



Hình 3-11: Giao diện trang quản trị - Thống kê phần 3

- Thống kê tài liệu



Hình 3-12: Giao diện trang quản trị - Thống kê phần 4

3.2.1.5 Giao diện trang quản trị người dùng

Tại mục quản trị người dùng, các quản trị viên có thể xem hoặc tìm kiếm trực tiếp toàn bộ danh sách người dùng hệ thống VTSign, với các danh sách như đang hoạt động, đã bị chặn và đã bị xóa. Bên cạnh đó quản trị viên có thể thêm ngược dòng mới hoặc thao tác trực tiếp lên người dùng đó với các hành động như xóa, chặn và chỉnh sửa. Đặc biệt với thao tác chỉnh sửa quản trị có thể chỉnh sửa thông tin cá nhân, xem được quản lý giao dịch, quản lý tài liệu của thành viên đó.

Danh sách người dùng

Thêm người dùng

ĐANG HOẠT ĐỘNG (39) ĐÃ BỊ CHẶN (6) ĐÃ BỊ XÓA (3)

TÊN	EMAIL	ĐỊA CHỈ	SỐ ĐIỆN THOẠI	CƠ QUAN	THAO TÁC
Admin Admin	admin@admin.com	227 Nguyen Van Cu	0975503208	HCMUS	
Canon Vu	Minhvu@gmail.com	654/6 Lạc Long Quân		HCMUS	
Chau Tuan	chauxuantuan@gmail.com		+84868367040		
Chau Tuan	phampham1109@gmail.com	a	0868367040	A	
doomdoog	vinaghost.hcmus@gmail.com		+84908341796		

Hiển thị mỗi trang 5 từ 1 đến 5 trong 39 người dùng

Hình 3-13: Giao diện trang quản trị người dùng – Phần 1

Quản lý giao dịch

Lưu

MÃ GIAO DỊCH SỐ TIỀN THỜI GIAN GIAO DỊCH LOẠI GIAO DỊCH PHƯƠNG THỨC THANH TOÁN MÔ TẢ

Không tìm thấy dữ liệu phù hợp

Quản lý tài liệu

Tài liệu cần ký 0	Tài liệu chờ ký 0	Tài liệu đã hoàn tất 0	Tài liệu đã xóa 0
-------------------	-------------------	------------------------	-------------------

Hình 3-14: Giao diện trang quản trị người dùng – Phần 2

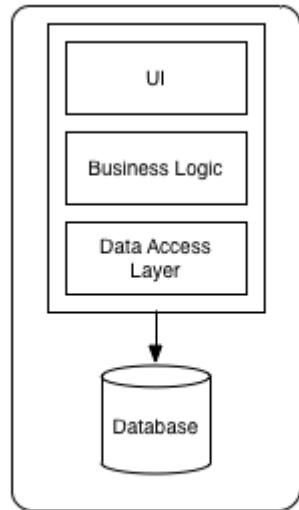
3.2.2 Thiết kế kiến trúc hệ thống

3.2.2.1 Giải pháp kiến trúc Microservice

- Giới thiệu giải pháp

- Khi nói về kiến trúc của 1 hệ thống phần mềm ta thường nghĩ đến 2 kiến trúc cho chúng đó là microservices và monolithic.

- Kiến trúc Monolithic được sử dụng từ những ngày đầu cho việc phát triển phần mềm. Tuy nhiên trong những năm gần đây thì kiến trúc monolithic lại lộ ra có nhiều nhược điểm cho việc phát triển phần mềm.

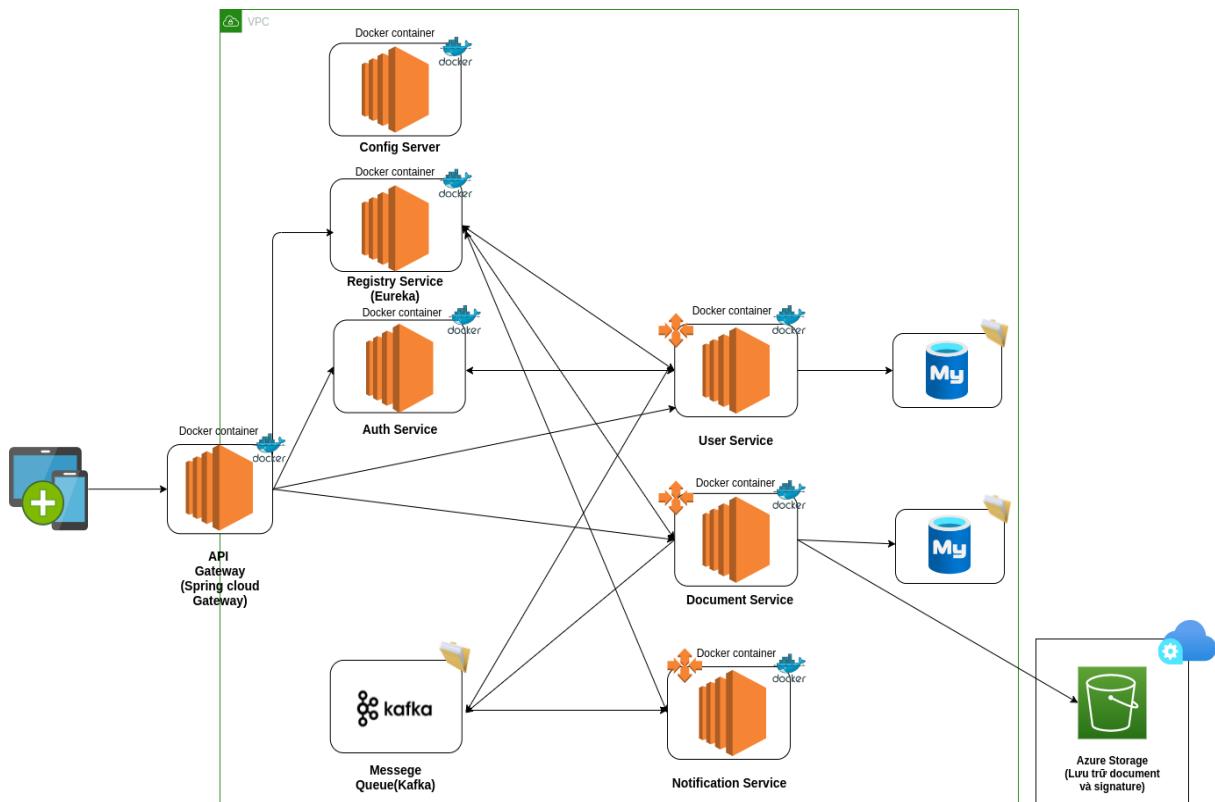


Monolithic Architecture

Hình 3-15: Kiến trúc Monolithic

- Mô hình kiến trúc Monolithic này có xu hướng phù hợp với những dự án có quy mô nhỏ. Tuy nhiên, sẽ gặp khá nhiều thách thức khi ứng dụng phát triển về quy mô và trở nên ngày càng phức tạp. Những thách thức bao gồm: Khó khăn trong việc bảo trì: vấn đề về coupling code, các khối code dính chặt lại với nhau, vấn đề cho thành viên mới sẽ khó để biết nên bắt đầu từ đâu trong 1 khối lớn. Quá trình phát triển sẽ mất đi tính linh hoạt: thời gian để xây dựng tính năng sẽ bị dài lên, bị block lẫn nhau. Bất kì một sự thay đổi dù nhỏ nào cũng cần chạy lại toàn bộ dự án => tốn khá nhiều thời gian. Tính ổn định không cao. Bất kì một lỗi nào có thể khiến toàn bộ ứng dụng bị ngừng. Tính mở rộng khó được đáp ứng trong trường hợp phải đáp ứng một lượng truy cập lớn từ phía người dùng.

- Từ những sau này nên kiến trúc Microservice được sinh ra để có thể giải quyết các vấn đề trên.



Hình 3-16: Kiến trúc Microservice

- Chi tiết giải pháp

- Microservice đề cập đến quá trình phát triển độc lập, tương đối nhỏ theo hướng chia hệ thống ra thành các services. Mỗi service này đều có một logic riêng, một trách nhiệm riêng và có thể được deploy riêng biệt. Khái niệm microservices đồng thời đề cập đến xu hướng tách biệt architecture ra thành các loose coupling service, tức là các service này sẽ có một mối liên hệ lỏng lẻo với nhau và mỗi service sẽ được nằm trong 1 ngữ cảnh nhất định.

- Các services được chạy 1 cách độc lập với nhau và sử dụng cơ sở dữ liệu riêng để lưu trữ dữ liệu, cho nên Microservices giải quyết các vấn đề của Monolithic như: Các service chạy độc lập với nhau nên khi 1 service bị dừng thì sẽ không ảnh hưởng đến các service khác. Khả năng mở rộng trở nên dễ dàng hơn khi có thể thêm 1 bản sao của service để đáp ứng lượng lớn truy cập. Thêm các services khác vào trong hệ thống mà không phải chạy lại toàn bộ hệ thống.

- Mục đích giải pháp

Hệ thống Ký kết văn bản trực tuyến là 1 hệ thống khá phức tạp và để đáp ứng yêu cầu dễ dàng mở rộng, bảo trì cho hệ thống khi hệ thống ngày càng có nhiều người sử dụng nên việc lựa chọn kiến trúc Microservice là hoàn toàn hợp lý và phù hợp với ngữ cảnh này.

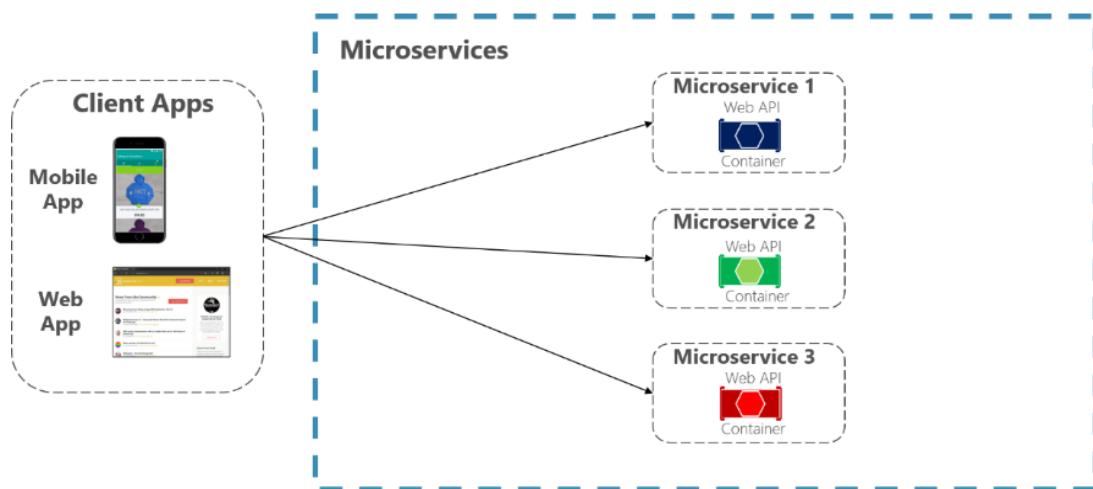
3.2.2.2 Giải pháp mô hình API Gateway

- Giới thiệu giải pháp

- Khi nói về phương thức giao tiếp giữa các microservices và ứng dụng phía client, có ba mô hình phổ biến nhất đó là Direct Pattern, API Gateway và Backend For Frontend (BFF).

- Direct Pattern là cách thiết lập cơ bản nhất cho một kiến trúc được xây dựng dựa trên microservices. Cách hoạt động của Direct Pattern được mô tả thông qua Hình 1 . Trong đó, mỗi microservice sẽ có một public endpoint (điểm truy cập mở) cho phép ứng dụng phía client hay một microservice khác có thể truy cập để giao tiếp.

Direct Client-To-Microservice communication Architecture



Hình 3-17: Cách hoạt động của Direct Pattern

- Tuy nhiên việc sử dụng Direct Pattern vẫn còn có nhiều bất cập như:
 - Kết nối: Do không có cổng API chung thì việc kết nối giữa client và Server nên client phải biết và kết nối tới nội bộ của từng service nên khi có bất kỳ sự thay đổi

nào đối với các microservices đều có khả năng khiến ứng dụng bị lỗi. Điều này khiến việc bảo trì trở nên khó khăn hơn.

- Vấn đề bảo mật: Không có cổng kết nối, các API phải mở ra bên ngoài, khả năng bị tấn công lớn hơn, khó bảo mật hơn.
- Khả năng phản hồi: Do kết nối trực tiếp nên việc đi lại mạng giữa client và server lâu gây chậm trễ cho việc phản hồi.

=> Cho nên mô hình API gateway là cách tiếp cận phù hợp hơn.

- Chi tiết giải pháp

- Khi bạn thiết kế và xây dựng các ứng dụng dựa trên kiến trúc microservice, API gateway là một phần không thể thiếu. API Gateway có thể coi là một cổng trung gian, nó là cổng vào duy nhất tới hệ thống microservices của chúng ta, API gateway sẽ nhận các requests từ phía client, chỉnh sửa, xác thực và điều hướng chúng đến những API cụ thể trên các services phía sau. Ngoài ra, API gateway còn giúp che dấu được cấu trúc của hệ thống microservices với bên ngoài

- Khi clients sẽ tương tác với hệ thống của chúng ta thông qua API gateway chứ không gọi trực tiếp tới một service cụ thể, các endpoints của những services sẽ chỉ được gọi nội bộ.

- Mục đích giải pháp

Giải quyết vấn đề giao tiếp giữa Client và Server Microservices như: Bảo mật, Hiệu năng, Khả năng kết nối.

3.2.3 Thiết kế giải pháp lưu trữ dữ liệu

- Dữ liệu lưu trữ là một phần quan trọng trong hệ thống, đối với kiến trúc Microservice, Các services khác nhau có yêu cầu lưu trữ dữ liệu khác nhau đảm bảo mối quan hệ lồng lẽo để việc deploy, development và scale được độc lập. MySQL là sự lựa chọn tốt nhất đối với hệ thống. Với hệ thống ký kết, nhóm sử dụng hai database chính tương ứng với hai service là user-service và document-service.

- Dưới đây, nhóm mô tả hai bảng chính trong 2 database để thể hiện thiết kế lưu trữ dữ liệu của nhóm

Bảng 3-1: Bảng người dùng – Table User

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	uuid	UUID	Khóa chính định danh người dùng
2	address	String	Địa chỉ người dùng
3	blocked	Boolean	Tài khoản người dùng đang bị khóa hay không?
4	email	String	Email liên hệ người dùng
5	enabled	Boolean	Người dùng đã kích hoạt hay chưa
6	firstName	String	Họ người dùng
7	lastName	String	Tên người dùng
8	organization	String	Tổ chức hoạt động
9	password	String	Mật khẩu
10	phone	String	Số điện thoại liên lạc
11	tempAccount	Boolean	Tài khoản là chính hay chưa được khởi tạo
12	balance	Long	Số dư tài khoản
13	avatar	String	Liên kết tới hình ảnh đại diện
14	deleted	Boolean	Tài khoản đã bị xóa hay chưa

15	createdBy	String	Người tạo tài khoản
16	createdAt	Date	Ngày tạo tài khoản
17	lastModifiedBy	String	Người sửa đổi cuối tài khoản
18	lastModifiedDate	Date	Ngày sửa đổi cuối tài khoản

Bảng 3-2: Bảng hợp đồng – Table Contract

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	uuid	UUID	Khóa chính định danh
2	completedDate	Date	Ngày hoàn thành hợp đồng
3	sentDate	Date	Ngày gửi hợp đồng
4	signed	Boolean	Hợp đồng đã được ký hay chưa?
5	title	String	Tên hợp đồng
6	createdBy	String	Người tạo hợp đồng
7	createdAt	Date	Ngày tạo hợp đồng
8	lastModifiedBy	String	Người sửa đổi cuối
9	lastModifiedDate	Date	Ngày sửa đổi cuối hợp đồng

Bảng 3-3: Bảng hợp đồng người dùng – Table User Contract

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	uuid	UUID	Khóa chính định danh
2	owner	Boolean	Người dùng có phải là người gửi tài liệu hay không?
3	permission	String	Quyền thao tác đối với hợp đồng
4	privateMessage	String	Tin nhắn riêng tư
5	publicMessage	String	Tin nhắn chung cho tất cả người nhận
6	secretKey	String	Khóa để mở khóa hợp đồng
7	status	String	Trạng thái hiện tại của hợp đồng
8	viewedDate	Date	Ngày xem hợp đồng
9	signedDate	Date	Ngày ký hợp đồng
10	preStatus	String	Trạng thái trước khi xóa. Dùng để khôi phục lại đúng trạng thái
11	contractUUID	UUID	Khóa ngoại. Thuộc về hợp đồng nào
12	userUUID	UUID	Khóa ngoại. Thuộc về người dùng nào
13	createdBy	String	Người tạo

14	createdDate	Date	Ngày tạo
15	lastModifiedBy	String	Người sửa lần cuối
16	lastModifiedDate	Date	Ngày sửa đổi lần cuối

3.3 Tổng kết

Kết thúc chương 3, nhóm đã làm rõ một số giải pháp tổng quan và cách thiết kế hệ thống VTSign. Bên cạnh đó nhóm trình bày sơ lược lý do vì sao chọn lựa các giải pháp này. Ở chương 4 cài đặt và triển khai, nhóm sẽ trình bày cách áp dụng những giải pháp đã nêu trong chương 3 để cài đặt và xây dựng hệ thống. Nội dung gồm các thư viện, môi trường và công cụ trong quá trình phát triển và quá trình xử lý những khó khăn gặp phải.

Chương 4

CÀI ĐẶT VÀ TRIỂN KHAI

4.1 Spring Framework

4.1.1 Giới thiệu tổng quan

- Spring là một nền tảng được xây dựng để phát triển các ứng dụng web trên ngôn ngữ Java. Nó được giới thiệu lần đầu tiên năm 2004. Tới năm 2006 các project con (sub-project) đã xuất hiện, mỗi project con tập trung vào một lĩnh vực khác nhau
- Spring là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code...
- Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lớn.
- Spring Framework được xây dựng dựa trên 2 nguyên tắc thiết kế chính là: DI (Dependency Injection) và AOP (Aspect Oriented Programming).
- Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object).

4.1.2 Một số dự án của Spring được sử dụng trong đồ án

- **Spring Boot**

Spring Boot là một framework giúp chúng ta phát triển cũng như chạy ứng dụng một cách nhanh chóng. Trong dự án nhóm dùng Spring Boot để viết các API Restful webservice

- **Spring Security**

Nhóm sử dụng Spring Security để làm cơ chế xác thực (authentication) và phân quyền (authorization) cho các services

- Spring Cloud

Spring Cloud là một công nghệ phần mềm sử dụng để phát triển các ứng dụng phân tán. Mục tiêu của Spring Cloud là làm sao để các thành phần của ứng dụng có thể giao tiếp với nhau. Spring cloud gồm nhiều thư viện hỗ trợ như Service Discovery (sử dụng Netflix Eureka), API gateway (Spring Cloud Gateway), Cloud Configuration (Spring Cloud Config Server) các thư viện này phù hợp để triển khai kiến trúc hệ thống microservice nhóm.

4.2 ReactJs

Đối với ứng dụng phía khách, nhóm chọn sử dụng Reactjs để cài đặt. Ngày nay, ReactJS là Thư viện JavaScript mã nguồn mở được rất phổ biến. Nó giúp tạo ra các ứng dụng web ấn tượng. Mục tiêu chính của ReactJS là phát triển Giao diện người dùng (UI) để cải thiện tốc độ của ứng dụng. Nhóm nhận thấy Reactjs phù hợp với hệ thống mà nhóm đang thực hiện với nhiều tính năng sau.

4.2.1 Tạo ứng dụng web động trên nền dễ dàng hơn

Để tạo một ứng dụng web động cụ thể với các chuỗi HTML rất khó vì nó yêu cầu mã hóa phức tạp, nhưng React JS đã giải quyết vấn đề đó và làm cho nó dễ dàng hơn. Nó cung cấp ít mã hóa hơn và cung cấp nhiều chức năng hơn. Nó sử dụng JSX (JavaScript Extension), là một cú pháp cụ thể cho phép các trích dẫn HTML và cú pháp thẻ HTML để hiển thị các thành phần con cụ thể. Nó cũng hỗ trợ việc xây dựng các mã máy có thể đọc được.

4.2.2 Các thành phần có thể tái sử dụng

Một ứng dụng web ReactJS được tạo thành từ nhiều thành phần và mỗi thành phần có logic và điều khiển riêng. Các thành phần này chịu trách nhiệm xuất ra một đoạn mã HTML nhỏ, có thể tái sử dụng, có thể được sử dụng lại ở bất cứ đâu bạn cần. Mã có thể sử dụng lại giúp phát triển và bảo trì ứng dụng dễ dàng hơn. Các Thành phần này có thể được lồng ghép với các thành phần khác để cho phép các ứng dụng phức tạp được xây dựng từ các khối xây dựng đơn giản.

4.2.3 Nâng cao hiệu suất

ReactJS cải thiện hiệu suất do DOM ảo.. Hầu hết các nhà phát triển đều gặp phải vấn đề khi DOM được cập nhật, điều này làm chậm hiệu suất của ứng dụng. ReactJS đã giải quyết vấn đề này bằng sử dụng DOM ảo. React Virtual DOM tồn tại hoàn toàn trong bộ nhớ và là đại diện cho DOM của trình duyệt web. Do đó, khi chúng ta viết một thành phần React, không ghi trực tiếp vào DOM. Thay vào đó, ta đang viết các thành phần ảo phản ứng sẽ chuyển thành DOM, dẫn đến hiệu suất mượt mà và nhanh hơn.

4.2.4 Các phần mềm sử dụng ReactJs hiện nay

Hiện nay, với độ phổ biến và phát triển mạnh mẽ của Reactjs. Nhiều trang web lớn đã và đang sử dụng Reactjs để phát triển website của họ. Những tên tuổi lớn phải kể đến như: Facebook, Instagram, Netflix, Dropbox....

4.3 Cài đặt kiến trúc

Nhóm bắt đầu cài đặt kiến trúc dự án dựa trên giải pháp mà nhóm đã tóm tắt ở chương 3 của báo cáo. Kiến trúc dự án là Microservice cần chia ra làm các service nhỏ để cài đặt. Kiến trúc của nhóm gồm 7 service chính như sau

4.3.1 Config Server

- Mỗi một dịch vụ (service) của nhóm là một dự án (project) được phát triển độc lập. Trong một project, ngoài mã (code) nó còn chứa các cấu hình, chẳng hạn như các thông tin kết nối vào cơ sở dữ liệu, thông tin về biến môi trường để gửi email, các thông tin về api key của bên thứ ba cung cấp. Các thông tin này được đặt trong các tập tin riêng biệt.
- Để quản trị các tập tin cấu hình này nhóm sử dụng một repository trên github để quản lý một cách nhanh chóng cho toàn bộ service.
- Nhóm sử dụng Spring Cloud Config Server làm một server riêng để đọc các cấu hình biến môi trường trên repository git và cung cấp các biến môi trường cấu hình này cho các service khác.

- Để triển khai server này, chúng ta cần tạo một dự án spring boot web, sau đó thêm vào một dependency là spring – cloud – config - server nằm trong groupId là org.springframework.cloud
- Tiếp đến cần cấu hình các biến môi trường để cho config-server này đọc dữ liệu từ một git repository trong file application.yml. Các biến cấu hình cần thiết bao gồm:
 - + server.port để chỉ định cổng chạy của server.
 - + spring.cloud.config.server.git.uri: địa chỉ uri của repository git
 - + spring.cloud.config.server.git.username: username đăng nhập có quyền truy cập vào repo git
 - + spring.cloud.config.server.git.password : password đăng nhập có quyền truy cập vào repo git
- Trong quá trình triển khai nhóm có gặp 2 vấn đề:
 - + Nếu để repo là public (lúc này sẽ không cần 2 biến môi trường là username và password) thì các biến môi trường sẽ bị lộ với cộng đồng điều này là không an toàn, các key config của cloud bên thứ ba sẽ bị lộ dẫn đến tài khoản sẽ tạm thời bị vô hiệu hóa. Cách giải quyết vấn đề là để repository của git ở lại chế độ private.
 - + Nhóm sử dụng github làm repository lưu trữ biến môi trường. Khi cung cấp username, với password để truy cập vào repo sẽ bị từ chối do từ ngày 13/08/2021 Github loại bỏ tính năng xác thực người dùng bằng username và password thông thường. Cách giải quyết vấn đề này đó là sử dụng tạo ra một token có quyền chỉ đọc repo trong cài đặt dành cho nhà phát triển, và dùng token này thay cho mật khẩu.
- Bước cuối trong để dựng config server là sử dụng annotation @EnableConfigServer trong file ConfigServerApplication để kích hoạt Config Server.

4.3.2 Registry Service

- Đây là một máy chủ dùng để quản lý, đặt tên cho các service. Trong trường hợp không muốn gán cứng các địa chỉ ip cho mỗi service. Khi mà các service của chúng ta sử dụng IP động, nó sẽ tự động cập nhật, chúng ta không cần thay đổi code.

- Nhóm lựa chọn Eureka để xây dựng Registry Service. Mỗi service sẽ tự động đăng ký và Eureka sẽ có nhiệm vụ quản lý các Service này. Nếu Eureka server không nhận được bất kỳ thông báo nào từ service thì service đó sẽ bị gỡ khỏi Eureka một cách tự động.
- Bắt đầu xây dựng Registry Service, nhóm sẽ dùng Maven để quản lý các dependencies. Nhóm khai báo file pom.xml có các dependencies Web: spring-boot-starter-web thuộc org.springframework.boot và Eureka Server tương ứng với: spring-cloud-starter-netflix-eureka-server thuộc org.springframework.cloud
- Tiếp theo, sẽ cấu hình trong file application.properties với các thông số như sau:
 - + spring.application.name=eureka-server: để cấu hình tên của service
 - + server.port=8761: port của service sẽ chạy
 - + eureka.client.register-with-eureka=false: cấu hình để eureka không tự đăng ký vào chính mình
 - + eureka.client.fetch-registry=false: cấu hình để eureka không gọi để lấy thông tin
- Cuối cùng, ta sẽ khai báo annotation @EnableEurekaServer trong project main.
- Trong quá trình xây dựng nhóm gặp phải vấn đề như sau:
 - + Eureka Service sẽ được public để các service con khác gọi đến đăng ký, điều này sẽ gây ra vấn đề về bảo mật khi bất kì ai cũng có thể gọi đến và đăng ký service của mình.
 - + Để giải quyết vấn đề này, nhóm đã cấu hình Eureka username và mật khẩu để có thể đăng ký, với mỗi service muốn đăng vào Eureka, service cần truyền thêm tham số user và password kèm đường dẫn đến Registry service để có thể xác thực đăng ký.

4.3.3 API Gateway

- Nó là một lớp trung gian giữa user và các service của bạn. Eureka server đã giải quyết vấn đề đặt tên cho từng service thay vì dùng địa chỉ IP của chúng. Vì vậy khi client gọi đến service, api gateway có nhiệm vụ định tuyến yêu cầu đó đến các service tương ứng thông qua việc lấy dữ liệu đó từ Eureka.
- Ngoài ra, Api Gateway còn có nhiệm vụ cân bằng tải, trong trường hợp số lượng người dùng tăng sau thời gian sử dụng, nhu cầu về đáp ứng lượng truy cập vào từng service cũng tăng, điều này phát sinh việc tăng instance cho từng service. Lúc này, Api gate đóng vai trò là một cân bằng tải giúp phân chia người dùng đến instance của từng service.
- Bắt đầu xây dựng Api Gateway, nhóm dùng Maven để quản lý các dependencies. Nhóm khai báo file pom.xml có các dependencies sau:
 - + spring-cloud-starter-gateway thuộc org.springframework.cloud: để cấu hình api gateway.
 - + spring-cloud-starter-netflix-eureka-client thuộc org.springframework.cloud: hỗ trợ đăng ký đến Registry Service .
 - + spring-cloud-starter-config thuộc org.springframework.cloud: để kết nối đến Config Server lấy thông tin cấu hình.
 - + spring-cloud-starter-openfeign thuộc org.springframework.cloud: để gọi tới auth-service lấy thông tin token user và đính kèm vào các request đi vào các service tiếp theo.
- Ở việc cấu láy thông tin service để kết nối tới các service chúng ta cần cấu hình tùy theo path mà người dùng gửi tới để gọi tới service tương ứng. Được thể hiện trong file cấu hình APIGatewayConfiguration. Chi tiết như sau:

```

public RouteLocator gatewayRoute(RouteLocatorBuilder builder) {
    return builder.routes()
        .route(r -> r.path("/user/"))
        .filters(f ->
1           .filter(gatewayFilterFactory.apply(new GatewayFilterFactory.Config())))
2               .uri("lb://user-service")
3           )
4       .route(r -> r.path("/document/"))
5           .filters(f ->
6               .filter(gatewayFilterFactory.apply(new GatewayFilterFactory.Config())))
7                   .uri("lb://document-service")
8               )
9           .route(r -> r.path("/auth/"))
10          .uri("lb://auth-service")
11      )
12      .route(r -> r.path("/notification/"))
13          .filters(f ->
14              .filter(gatewayFilterFactory.apply(new GatewayFilterFactory.Config())))
15                  .uri("lb://notification-service")
16              )
17          .route(r -> r.path("/v3/api-docs/*"))
18              .filters(f -> f.rewritePath("/v3/api-
19 docs/(?<path>. )", "/${path}/v3/api-docs/"))
20                  .uri(String.format("http://localhost/%s",
21 env.getProperty("server.port")))
22              )
23          .build();
}

```

- Cuối cùng, chúng ta sẽ khai báo annotation @EnableEurekaClient trong file ApiGatewayApplication để kích hoạt đăng ký đến Register Service.

4.3.4 Notification service

- Để thực hiện các tác vụ thông báo (bằng email, SMS) nhóm sử dụng một service riêng để làm điều này. Lý do cần một service thông báo riêng là bởi vì:
 - + Thông báo dùng email, và sms các biến môi trường này là chung và nên tách biệt riêng vào một service. Các service khác có nhu cầu thông báo chỉ cần gửi yêu cầu và nội dung sang.
 - + Gửi mail là một tác vụ nặng và tốn thời gian. Sẽ tốt hơn nếu chúng ta có một service riêng lẻ để làm điều đó, tăng tốc độ phản hồi cho khách hàng.
- Nhóm sử dụng Spring Boot để tạo một service đơn giản. Tiếp theo để triển khai các tính năng cần thiết service này nhóm sử dụng các thư viện phụ thuộc sau:

- + spring-boot-starter-mail thuộc org.springframework.boot: Thư viện hỗ trợ để gửi email.
- + spring-cloud-starter-config thuộc org.springframework.cloud: Thư viện hỗ trợ đọc các biến môi trường từ Config Server đã trình bày ở mục 4.3.1
- + spring-kafka thuộc org.springframework.kafka: để xử lý các message queue nhận yêu cầu từ các service khác có nhu cầu gửi thông báo gửi đến.
- + thymeleaf-layout-dialect thuộc nz.net.ultraq.thymeleaf: đẩy các biến dữ liệu động vào trong template email.
- + twilio thuộc com.twilio.sdk: sdk của twilio để gửi SMS tới số điện thoại.

- Bước cuối cùng để dựng notification service là sử dụng annotation @EnableKafka trong file NotificationServiceApplication để kích hoạt sử dụng kafka, đồng thời trong mỗi hàm lắng nghe sự kiện từ các service khác gửi đến cần thêm annotation @KafkaListener (tên topic).

4.3.5 Auth Service

- Trong auth service, chúng ta sẽ làm hai việc: một là xác thực định danh mà người dùng cung cấp và hai là generate ra một token trong trường hợp xác thực hợp lệ hoặc trả về một exception nếu nó không hợp lệ.
- Bắt đầu xây dựng Auth Service, nhóm dùng Maven để quản lý các dependencies. Nhóm khai báo file pom.xml có các dependencies sau:
 - + spring-cloud-starter-config thuộc org.springframework.cloud: Thư viện hỗ trợ đọc các biến môi trường từ Config Server đã trình bày ở mục 4.3.1
 - + spring-cloud-starter-openfeign thuộc org.springframework.cloud: Rest client để giao tiếp với user-service .
 - + jjwt-api, jjwt-impl và jjwt-jackson thuộc io.jsonwebtoken: thư viện giúp cung cấp mã jwt cũng như xác thực jwt trong java.
- Tiếp theo, chúng ta sẽ khai báo annotation @EnableEurekaClient trong file AuthServiceApplication để kích hoạt đăng ký đến Register Service.

4.3.6 Document Service

- Document Service là service để xử lý các request liên quan đến hợp đồng ký kết.
- Bắt đầu xây dựng Document Service, nhóm dùng Maven để quản lý các dependencies. Nhóm khai báo file pom.xml có các dependencies sau:
 - + spring-boot-starter-data-jpa thuộc spring-boot-starter-data-jpa: khai báo ORM hỗ trợ truy vấn đến database
 - + spring-cloud-starter-config thuộc org.springframework.cloud: Thư viện hỗ trợ đọc các biến môi trường từ Config Server đã trình bày ở mục 4.3.1
 - + spring-cloud-starter-netflix-eureka-client thuộc org.springframework.cloud: hỗ trợ đăng ký đến Registry Service .
 - + spring-boot-starter-security khai thuộc org.springframework.boot: khai báo hỗ trợ authentication và authorization.
 - + jjwt-api, jjwt-impl và jjwt-jackson thuộc io.jsonwebtoken: thư viện giúp cung cấp mã jwt cũng như xác thực jwt trong java.
 - + spring-kafka thuộc org.springframework.kafka: dùng để đẩy các message tới notification service để thông báo.
 - + mysql-connector-java thuộc mysql: driver để hỗ trợ kết nối đến database mysql.
 - + azure-spring-boot-starter-storage thuộc com.azure.spring: hỗ trợ kết nối đến Azure Storage để thao tác với tài liệu.
- Tiếp theo, chúng ta sẽ khai báo annotation @EnableEurekaClient trong file AuthServiceApplication để kích hoạt đăng ký đến Register Service.

4.3.7 User Service

- User Service là service để xử lý các yêu cầu liên quan đến tài khoản, thông tin người dùng, thanh toán.
- Bắt đầu xây dựng User Service, nhóm dùng Maven để quản lý các dependencies. Nhóm khai báo file pom.xml có các dependencies sau:

- + spring-boot-starter-data-jpa thuộc spring-boot-starter-data-jpa: khai báo ORM hỗ trợ truy vấn đến database.
 - + spring-cloud-starter-config thuộc org.springframework.cloud: Thư viện hỗ trợ đọc các biến môi trường từ Config Server đã trình bày ở mục 4.3.1
 - + spring-cloud-starter-netflix-eureka-client thuộc org.springframework.cloud: hỗ trợ đăng ký đến Registry Service.
 - + spring-boot-starter-security khai báo thuộc org.springframework.boot: khai báo hỗ trợ authentication và authorization.
 - + jjwt-api, jjwt-impl và jjwt-jackson thuộc io.jsonwebtoken: thư viện giúp cung cấp mã jwt cũng như xác thực jwt trong java.
 - + spring-kafka thuộc org.springframework.kafka: dùng để đẩy các message tới notification service để thông báo.
 - + mysql-connector-java thuộc mysql: driver để hỗ trợ kết nối đến database mysql.
 - + azure-spring-boot-starter-storage thuộc com.azure.spring: hỗ trợ kết nối đến Azure Storage để thao tác với hình ảnh avatar của người dùng.
- Tiếp theo, chúng ta sẽ khai báo annotation @EnableEurekaClient trong file AuthServiceApplication để kích hoạt đăng ký đến Register Service.

4.4 Cài đặt chức năng

4.4.1 Đăng ký, đăng nhập

4.4.1.1 Đăng ký

- Hệ thống cung cấp API với endpoint https://api.vtsign.tech/auth/register sử dụng phương thức HTTP POST cho phép người dùng thực hiện đăng ký tài khoản. Người dùng cần cung cấp các thông tin cần thiết như: email, password, phone, organization, address, first_name, last_name. Kết quả trả về của endpoint được mô tả ở bảng dưới.

Bảng 4-1: Bảng trạng thái HTTP cho việc đăng ký tài khoản

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Đăng ký thành công
2	409	Email đã được sử dụng
3	419	Thiếu trường dữ liệu
4	500	Lỗi hệ thống

- Sau khi người dùng nhập đầy đủ thông tin đăng ký, front-end sẽ gửi request đến Auth Service với endpoint trên. Yêu cầu sẽ được tiếp nhận tại package controller tại method register (ở đây chúng ta cấu hình annotation `@PostMapping("/register")` để tiếp nhận yêu cầu).
- Method register nhận vào tham số `RegisterClientRequestDto` bao gồm các trường yêu cầu. Controller sẽ gọi hàm `register` từ service với tham số truyền vào là `RegisterClientRequestDto`. Tại hàm `register`, chúng ta tiến hành gọi proxy đến hàm đăng ký tại User-Service thông qua Feign, Feign – một ứng dụng client HTTP khai báo do Netflix phát triển. Nhằm mục đích đơn giản hóa các ứng dụng HTTP API.
- Một cách đơn giản, lập trình viên chỉ cần khai báo và chú thích một interface trong khi việc triển khai thực tế được cung cấp trong thời gian chạy. Hàm `register` tại Auth-Service thông qua end-point <https://api.vtsign.tech/user/api/register> để tiến hành đăng ký người dùng.
- Tương tự tại User Service, `UserController` sẽ tiếp nhận yêu cầu đăng ký từ Auth Service tại method `register` (ở đây ta cấu hình annotation `@PostMapping("/register")` để điều hướng).

4.4.1.2 Đăng nhập

- Hệ thống cung cấp API với endpoint <https://api.vtsign.tech/auth/login> sử dụng phươn thức HTTP POST cho phép người dùng thực hiện đăng nhập tài khoản.

- Người dùng cần cung cấp thông tin email và password để đăng nhập vào hệ thống. Kết quả trả về của endpoint này được mô tả bằng bảng dưới đây.

Bảng 4-2: Bảng trạng thái HTTP cho việc đăng nhập

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Đăng nhập thành công, trả về thông tin người dùng
2	401	Sai email hoặc mật khẩu
3	419	Thiếu email hoặc mật khẩu
4	423	Tài khoản của người dùng chưa được kích hoạt
5	500	Lỗi hệ thống

- Dữ liệu người dùng nhập sẽ được phía front-end gọi lên thông qua api phía bên trên. Tại Auth-service nhóm em có viết một hàm để tiếp nhận yêu cầu này ở trong package controller với annotation @PostMapping("/login").
- Tại đây sẽ gọi tới jwtService.login() để tiến hành xác thực tài khoản người dùng. Trong JwtService sẽ tiến hành gọi sang user-service để lấy về thông tin của người dùng, và tiến hành kiểm tra.
 - + thông tin là không hợp lệ: chuyển tiếp các lỗi tương ứng về cho người dùng.
 - + thông tin là hợp lệ: tiến hành tạo mã jwt (access_token, refresh_token) đính kèm vào thông tin của người dùng và trả về sử dụng data transfer object (LoginClientResponseDto) để đính kèm thông tin, các thông tin nhạy cảm như password của user dù đã được mã hóa nhưng vẫn không trả về cho người dùng.
- Ở bước trên trình bày sẽ gọi sang user-service để tiến hành lấy thông tin người dùng.

4.4.2 Ký kết hợp đồng

4.4.2.1 Gửi hợp đồng

- Hệ thống cung cấp API với endpoint <https://api.vtsign.tech/document/signing> sử dụng phương thức HTTP POST cho phép người dùng gửi hợp đồng.
- Tại front-end, nhóm sử dụng PDFTron để xử lý hợp đồng dạng pdf. Sau khi điền đủ đầy đủ thông tin cùng với hợp đồng đã được yêu cầu các trường để ký. Front-end gửi yêu cầu đến endpoint trên để gửi tài liệu ký kết cho đối tác.
- Tại back-end, yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-3: Bảng trạng thái HTTP cho việc gửi hợp đồng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Gửi hợp đồng thành công
4	400	Thiếu trường hoặc accessToken
5	500	Lỗi hệ thống

- Yêu cầu sẽ được xử lý chính tại Document-service và Notification-service. Bên trong document service, yêu cầu được điều hướng đến method signing tại class DocumentController (ta cấu hình annotation `@PostMapping(value = "/signing")` để cấu hình điều hướng). Tham số nhận được là DocumentClientRequest và danh sách các file hợp đồng cần ký kết.
- Tiếp theo, chúng ta tiếp tục gọi đến method createUserDocument để tiến hành xử lý. Đầu tiên, chúng ta kiểm tra xem số tiền trong tài khoản của người dùng có đủ để gửi tài liệu hay không? (mặc định 5000 vnd/tài liệu), nếu thỏa mãn điều kiện, ta tiến hành tạo tài liệu và gửi thông tin người nhận đến kafka để gửi thông báo ký kết.

- Tại notification-service, method consumeMessage sẽ nhận được thông tin người nhận và tiến hành gửi gmail chứa đường dẫn đến tài liệu ký kết và sms chứa mã để mở khóa tài liệu.

4.4.2.2 Ký hợp đồng

- Chức năng cung cấp API cung cấp dịch vụ ký hợp đồng nhận được từ đối tác được xem là chức năng cốt lõi của hệ thống. Nó cho phép người dùng sử dụng dịch vụ ký hợp đồng mà hệ thống cung cấp.
- Hệ thống được xây dựng nhằm mục đích phục vụ đem lại sự tiện lợi cho mọi người, thay thế cách ký hợp đồng truyền thống. Do vậy hệ thống cung cấp 2 endpoint là 1 là cho người dùng đã có tài khoản và người dùng chưa có tài khoản.
- Đối với người dùng chưa có tài khoản thì hệ thống sẽ tạo 1 đường dẫn ký kết để gửi vào mail của người nhận, và sẽ có 1 mã khóa bí mật được người gửi tạo và gửi qua người nhận thông qua số điện thoại. Khi người dùng nhận được mã khóa bí mật thì người dùng nhấn vào đường dẫn trong email đó là endpoint do hệ thống cung cấp(<https://api.vtsign.tech/document/apt/signing>)sử dụng phương thức GET để xác nhận mã khóa bí mật và mở khóa tài liệu.
- Khi người là người cuối cùng mở khóa để ký kết thì hệ thống sẽ trả về cho front-end là là người ký cuối cùng ký kết để front end nhận biết và merge các XFDF khác do những người dùng khác đã ký kết. Đây là bảng trạng thái của endpoint mở khóa tài liệu.

Bảng 4-4: Bảng trạng thái HTTP cho việc mở khóa tài liệu

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Mở khóa tài liệu thành công
2	400	Thiếu trường hoặc accessToken hoặc trường bất kỳ
3	403	Không có quyền xem và mở khóa hợp đồng này

4	404	Không tìm thấy hợp đồng hoặc hợp đồng đã bị xóa
5	423	Hợp đồng đã được ký kết
6	500	Lỗi hệ thống

- Yêu cầu sẽ được xử lý chính tại Document-service khi người dùng nhập đúng mã khóa bí mật thì tài liệu sẽ được trả về và hiển thị lên giao diện cho phép người dùng ký kết.
- Hệ thống cung cấp endpoint để việc ký kết văn bản(<https://api.vtsign.tech/document/apt/signing>) sử dụng phương thức POST.
- Khi người dùng ký kết và nhấn nút hoàn tất thì front-end sử dụng pdf component(PDFTron) sẽ tạo ra 1 chuỗi XFDF theo dạng XML và đưa lên hệ thống. Khi hệ thống nhận được sẽ kiểm tra tất cả mọi người đã ký kết hết chưa và cho tài liệu được ký kết hoàn tất. Đây là bảng trạng thái endpoint của ký kết văn bản.

Bảng 4-5: Bảng trạng thái HTTP cho việc ký kết văn bản

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
4	404	Không tìm thấy người dùng này
5	419	Yêu cầu không đúng định dạng
6	500	Lỗi hệ thống

- Sau khi ký kết thành công nếu tài liệu đã được hoàn thành, endpoint này sẽ gửi 1 sự kiện vào Topic hoàn tất hợp đồng và Notification đang lắng nghe trên topic đó sẽ kích hoạt và gửi tài liệu đã hoàn thành cho tất cả mọi người đến email và sẽ hiển thị lên trang quản lý của từng thành viên(nếu có tài khoản).

- Với những người đã có tài khoản, vẫn có thể sử dụng phương pháp trên để ký kết hợp đồng hoặc có cách khác để ký kết cho những người đã có tài khoản là đăng nhập vào hệ thống vào mục quản lý, vào xem chi tiết và ký kết.
- Với những người có tài khoản và đăng nhập hệ thống, hệ thống cung cấp endpoint(<https://api.vtsign.tech/document/sign-document>) để ký kết hợp đồng, việc ký kết này dựa vào jwt gửi lên cùng với yêu cầu. Luồng làm việc giống với người chưa có tài khoản.

4.4.2.3 Thanh toán chi phí sử dụng dịch vụ

- Hệ thống cung cấp API với endpoint <https://api.vtsign.tech/user/deposit> sử dụng phương thức HTTP POST cho phép người dùng nạp tiền vào tài khoản.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-6: Bảng trạng thái HTTP cho việc thanh toán chi phí

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thanh toán thành công
2	400	Thiếu trường hoặc accessToken
3	404	Không tìm thấy user
4	419	Thiếu trường yêu cầu
6	500	Lỗi hệ thống

- Yêu cầu sẽ được xử lý chính tại User-Service. Bên trong User-Service, yêu cầu được điều hướng đến method depositMoney ở UserController (ta cấu hình annotation `@PostMapping(value = "/signing")`). Tham số nhận được sẽ là `userDetails` và `userDepositDto` bao gồm phương thức thanh toán và số tiền. Sau đó, chúng ta tiếp tục gọi đến method deposit trong userService để xử lý.
- Method deposit nhận được lời gọi từ controller và tiến hành tạo data để gửi yêu cầu đến ZalopayServer. Data bao gồm các thông `appId`, `appUser`, `appTransId`, `appTime`,

amount, embedData, bankCod, mac, callbackUrl, redirectUrl,... Sau đó chúng ta tiến hành gửi request đến <https://sb-openapi.zalopay.vn/v2> với body là data mà chúng ta đã tạo thông qua Feign. Lúc này, Zalopay sẽ xử lý yêu cầu. Nếu thành công, chúng ta sẽ nhận được kết quả từ zalopay bao gồm các thông tin như: code, message, subCode, subMessage, transToken, orderUrl, orderToken.

- Front-end sẽ nhận được thông tin từ request và tiến hành điều hướng đến orderUrl có trong dữ liệu trả về. Lúc này, trang thanh toán sẽ xuất hiện tùy thuộc vào phương thức thanh toán mà người dùng đã lựa chọn. Người dùng tiến hành thanh toán.
- Sau khi người dùng thanh toán thành công, zalopay sẽ gửi một callback đến urlCallback mà ta đã gửi trong yêu cầu trước. Chúng ta sẽ nhận callback tại endpoint <https://api.vtsign.tech/user/apt/deposit/callback>. Yêu cầu được điều hướng đến method orderCallback tại AcceptController với tham số nhận được là zaloPayCallbackRequest bao gồm data, mac, type. AcceptController gọi đến method updateUserBalance để cập nhật tài khoản của người dùng. Trong method updateUserBalance sẽ kiểm tra callback có hợp lệ hay không dựa trên mac nhận được, nếu hợp lệ, ta tiến hành cập nhật tài khoản người dùng.

4.4.3 Quản lý hợp đồng

- Để quản lý hợp đồng thì nhóm đã đề ra cho hợp đồng có 5 trạng thái chính là : hoàn thành, chờ ký, cần ký, đã xóa, ẩn.
- Để quản lý hợp đồng hệ thống cung cấp 4 endpoint chính để thực hiện việc quản lý các hợp đồng.
 - Lọc hợp đồng theo trạng thái của hợp đồng
- Hệ thống cung cấp API với endpoint(<https://api.vtsign.tech/document/filter>) sử dụng phương thức GET cho phép người dùng lấy ra danh sách các tài liệu hợp đồng theo từng trạng thái như hoàn thành, chờ ký, cần ký và xóa.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-7: Bảng trạng thái HTTP chov việc lọc hợp đồng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	404	Không tìm thấy hợp đồng
4	500	Lỗi hệ thống

- Yêu cầu sẽ được xử lý chính tại Document-Service. Bên trong Document-Service yêu cầu được điều hướng đến /filter . Tham số truyền vào sẽ là các biến như status, ngày hoàn thành, ngày tạo, title để lọc hoặc tìm kiếm danh sách hợp đồng tương ứng. Khi các tham số được truyền vào, Khi đó document-service sẽ sử dụng JPA Specification để thực hiện việc query trên database và trả về kết quả danh sách các tài liệu của UserUUID đó và có trạng thái đó, theo những trường đó.
 - Xóa hợp đồng
- Hệ thống cung cấp API với endpoint(<https://api.vtsign.tech/document/delete>) sử dụng phương thức DELETE cho phép người dùng xóa hợp đồng không cần thiết(Thay đổi trạng thái của hợp đồng)
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-8: Bảng trạng thái HTTP cho việc xóa hợp đồng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	404	Không tìm thấy hợp đồng

4	422	Người dùng này không thể xóa hợp đồng
5	500	Lỗi hệ thống

- Yêu cầu sẽ được xử lý chính tại Document-Service. Bên trong Document-Service yêu cầu được điều hướng đến “/delete”. Tham số truyền vào là userId,contractId và userContractId để kiểm tra người dùng có tồn tại, hợp đồng có tồn tại và người dùng này có hợp đồng này không, khi thực hiện việc xóa hợp đồng thì trạng thái hiện tại của hợp đồng vẫn được giữ nguyên để thực hiện việc rollback mà vẫn đúng trạng thái trước khi xóa.
 - Khôi phục hợp đồng
- Hệ thống cung cấp API với endpoint(<https://api.vtsign.tech/document/restore>) sử dụng phương thức POST cho phép người dùng khôi phục các hợp đồng đã xóa.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-9: Bảng trạng thái HTTP cho việc khôi phục hợp đồng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	404	Không tìm thấy hợp đồng
4	422	Người dùng này không thể khôi phục hợp đồng
5	500	Lỗi hệ thống

- Yêu cầu sẽ được xử lý chính tại Document-Service. Bên trong Document-Service yêu cầu được điều hướng đến “/delete”. Tham số truyền vào là userId,contractId và userContractId để kiểm tra người dùng có tồn tại, hợp đồng có tồn tại và người dùng này có hợp đồng này không, chỉ thực hiện được việc khôi phục hợp đồng khi hợp đồng đang ở trạng thái đã xóa, khi khôi phục hợp đồng sẽ quay trở lại trạng thái trước khi bị xóa.
 - Ẩn hợp đồng
- Hệ thống cung cấp API với endpoint(<https://api.vtsign.tech/document/hidden>) sử dụng phương thức DELETE cho phép người dùng ẩn hợp đồng không cần thiết(Thay đổi trạng thái của hợp đồng)
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-10: Bảng trạng thái HTTP cho việc ẩn hợp đồng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	404	Không tìm thấy hợp đồng
4	422	Người dùng này không thể xóa hợp đồng
5	500	Lỗi hệ thống

- Yêu cầu sẽ được xử lý chính tại Document-Service. Bên trong Document-Service yêu cầu được điều hướng đến “/hidden”. Tham số truyền vào là userId,contractId và userContractId để kiểm tra người dùng có tồn tại, hợp đồng có tồn tại và người dùng này có hợp đồng này không, khi thực hiện việc ẩn hợp đồng, hợp đồng sẽ được thay đổi trạng thái thành ẩn và người dùng không thể

nhìn thấy hợp đồng, chỉ có người quản lý mới có thể thấy, khi hợp đồng bị ẩn thì không thể khôi phục được nữa.

4.4.4 Quản lý người sử dụng dịch vụ

4.4.4.1 Thêm người dùng

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/create-user>) sử dụng phương thức POST cho phép admin lấy danh sách user.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-11: Bảng trạng thái HTTP cho việc thêm người dùng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	201	Thành công
3	403	Thiếu access token
4	409	Email đã tồn tại
5	419	Thiếu trường dữ liệu
6	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User-Service, yêu cầu sẽ được điều hướng và tiếp nhận tại method createUser trong ManagementController (tại cấu hình annotation `@PostMapping(value = "/management/create-user")`).
- Method createUser nhận vào tham số `userRequestDto` bao gồm các trường yêu cầu. Controller sẽ gọi hàm `register` từ service với tham số truyền vào là `userRequestDto`. Tại hàm `save`, ta tiến hành kiểm tra các trường dữ liệu và lưu thông tin người dùng.
- Sau đó, ta gọi `userProducer` để gửi message đến kafka tại topic `user-service-register`. Lúc này, method `consumeMessage` tại service của Notification Service sẽ nhận được message từ topic trên. Ta tiếp tục sử dụng `emailSenderService` để gửi mail đính kèm đường dẫn kích hoạt tài khoản đến người dùng.

4.4.4.2 Cập nhật người dùng

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/update-user>) sử dụng phương thức POST cho phép admin lấy danh sách user.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-12: Bảng trạng thái HTTP cho việc cập nhật người dùng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	403	Đường dẫn bị cấm
4	404	Không tìm thấy user
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User-Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức updateUser trong ManagementController (ta cấu hình annotation `@PostMapping(value = "/management/signing")`).
- Method update nhận vào tham số userId và userUpdateDto bao gồm các trường cập nhật. Controller sẽ gọi hàm updateUser từ service với tham số truyền vào là userId và userUpdateDto. Tại hàm updateUser, ta tiến hành kiểm tra các trường dữ liệu và cập nhật thông tin người dùng.
- Sau đó, ta gọi userProducer để gửi message đến kafka tại topic user-service-notify-common. Lúc này, method consumeMessageCommon tại service của Notification Service sẽ nhận được message từ topic trên. Ta tiếp tục sử dụng emailSenderService để gửi mail để thông báo đã cập nhật thông tin người dùng.

4.4.4.3 Xem danh sách người dùng đang hoạt động

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/list>) sử dụng phương thức GET cho phép admin lấy danh sách user.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-13: Bảng trạng thái HTTP cho việc xem danh sách người dùng hoạt động

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	403	Đường dẫn bị cấm
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức getManagement trong ManagementController (ta cấu hình annotation `@PostMapping(value = "/management/list")`).
- Phương thức getManagement nhận vào tham số page, pageSize, sortField, sortType, keyword. Sau đó, ta sẽ gọi phương thức getUserManagement từ service với tham số truyền vào là những tham số nhận được ở trên. Tại đây, ta tạo pageable để truy vấn dữ liệu. Ta sử dụng hỗ trợ của jpa để truy vấn dữ liệu theo pageable và sau đó trả dữ liệu về controller để trả về cho người dùng.

4.4.4.4 Xem danh sách người dùng đã bị khóa

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/list-block>) sử dụng phương thức GET cho phép admin lấy danh sách user đã bị khóa.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-14: Bảng trạng thái HTTP cho việc xem danh sách người dùng đã bị khóa

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	403	Đường dẫn bị cấm
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User-Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức getBlockedUsers trong ManagementController (ta cấu hình annotation @GetMapping(value = "/management/list-block").
- Phương thức nhận vào tham số là (page, pageSize, sortField, sortType, keyword) trong đó tất cả các trường đè là không bắt buộc. Controller sẽ gọi hàm getBlockedUsers từ Service. Sau đó từ Service sẽ gọi hàm bên trong @Repository để sử dụng JPA lấy dữ liệu từ Cơ Sở Dữ Liệu . Tùy vào các trường mà người dùng nhập lên thì danh sách user trả về được sắp xếp và trả về đúng với yêu cầu người dùng mong muốn.

4.4.4.5 Xem danh sách người dùng đã bị xóa

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/list-deleted>) sử dụng phương thức GET cho phép admin lấy danh sách user.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-15: Bảng trạng thái HTTP cho việc xem danh sách người dùng đã bị xóa

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
-----	--------------------	---------------------------

1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	403	Đường dẫn bị cấm
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức getDeletedUsers trong ManagementController (ta cấu hình annotation `@GetMapping(value = "/management/list-deleted")`).
- Phương thức getDeletedUsers nhận vào tham số page, pageSize, sortField, sortType, keyword. Sau đó, ta sẽ gọi phương thức getDeletedUsers từ service với tham số truyền vào là những tham số nhận được ở trên. Tại đây, ta tạo pageable để truy vấn dữ liệu. Ta sử dụng hỗ trợ của jpa để truy vấn dữ liệu người dùng đã bị xóa bằng điều kiện deleted=true theo pageable và sau đó trả dữ liệu về controller để trả về cho người dùng.

4.4.4.6 Khóa và mở người dùng

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/block-user>) sử dụng phương thức PUT cho phép admin có thể khóa hoặc mở khóa 1 tài khoản nào đó.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-16: Bảng trạng thái HTTP cho việc khóa và mở người dùng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken

3	403	Đường dẫn bị cấm
4	404	Không tìm thấy người dùng này
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User-Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức blockUser trong ManagementController (ta cấu hình annotation @PutMapping(value = "/management/block-user")).
- Phương thức nhận vào tham số là UserStatusDto bao gồm userId và status trong đó tất cả các trường đều là bắt buộc. Controller sẽ gọi hàm blockUser từ Service. Service sẽ thực hiện việc gọi đến Repository để JPA cập nhật lại trạng thái của người sử dụng tùy vào status được truyền lên.

4.4.4.7 Xóa người dùng

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/delete-user>) sử dụng phương thức DELETE cho phép admin có thể khóa xóa một tài khoản bất kỳ.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-17: Bảng trạng thái HTTP cho việc xóa người dùng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	403	Đường dẫn bị cấm
4	404	Không tìm thấy người dùng

		này
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User-Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức deleteUser trong ManagementController (ta cấu hình annotation @GetMapping(value = "/management/delete-user")).
- Phương thức nhận vào tham số là UserStatusDto bao gồm userId và status. Controller sẽ gọi phương thức deleteUser từ Service. Tại phương thức này, ta gọi findById với tham số truyền vào là userId, với kết quả user tìm kiếm được, ta setDeleted(isDelete) để xóa hoặc hoàn tác người dùng tùy thuộc vào status người dùng yêu cầu.
- Sau đó, gọi gửi message đến Notification Service thông qua kafka tại topic user-service-notify-common. Lúc này, method consumeMessageCommon tại service của Notification Service sẽ nhận được message từ topic trên. Ta tiếp tục sử dụng emailSenderService để gửi mail để gửi thông báo tài khoản người dùng đã bị khóa hoặc hoàn tác đến người dùng.

4.4.5 Báo cáo cho quản trị viên về tổng quan hệ thống

4.4.5.1 Thống kê tổng quan

- Hệ thống sẽ tính toán các thông số tổng quan (số lượng người dùng, số lượng người đăng ký, số lượng tài liệu đã gửi, hoàn thành) và hiển thị cho người quản trị số lượng theo thời gian (ngày, tuần, tháng, năm, tất cả)
- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/count-user>) để tính toán số lượng người dùng của hệ thống theo thời gian.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-18: Bảng trạng thái HTTP cho việc thống kê tổng quan

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	403	Đường dẫn bị cấm
4	404	Không tìm thấy dữ liệu thời gian (type) phù hợp
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức countByDate trong ManagementController (annotation @GetMapping("/management/count-user")).
- Phương thức nhận vào tham số là type (date, week, month, year, all). Nếu type là all thì sẽ tiến hành đếm tất cả người dùng bằng hàm count() trong service. Nếu type là các type còn lại sẽ tiến hành lấy ra ngày bắt đầu và ngày kết thúc theo type bằng DateUtil.getDateBetween() và tiến hành gọi hàm countUserBetweenDate() trong service. Cuối cùng sẽ trả về số lượng tính toán được cho người dùng. Tương tự hệ thống cung cấp thêm 2 API
- Endpoint (<https://api.vtsign.tech/user/management/total-deposit>) để tính toán số tiền người dùng nạp vào hệ thống theo thời gian.
- Endpoint (<https://api.vtsign.tech/document/management/count-contract>) để tính toán số lượng tài liệu (được gửi, đã hoàn thành) của hệ thống theo thời gian.

4.4.5.2 Vẽ biểu đồ số lượng người đăng ký sử dụng

- Hệ thống sẽ dựa vào số liệu người dùng đăng ký theo từng khung thời gian(tuần, tháng, năm) để vẽ ra biểu đồ. Từ đó, người quản trị có thể có cái nhìn tổng quan về quá trình hoạt động của hệ thống.
- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/statistic-user>) sử dụng phương thức GET cho phép front-end có thể lấy về số liệu để vẽ biểu đồ.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-19: Bảng trạng thái HTTP cho việc vẽ biểu đồ số lượng người đăng ký sử dụng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
3	403	Đường dẫn bị cấm
4	404	Không tìm thấy khoảng thời gian
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức statisticUser trong ManagementController (ta cấu hình annotation `@GetMapping("/management/statistic-user")`).
- Phương thức nhận vào tham số là type (WEEK, MONTH, YEAR). Controller sẽ gọi hàm getStatisticUser từ Service. Ta tiếp tục gọi phương thức getListLocalDateTime truyền vào type ta nhận được một danh sách gồm thời gian bắt đầu và ngày kết thúc của từng khung thời gian. Với từng ngày bắt đầu và kết

thúc trong danh sách, ta gọi `countAllByCreatedDateBetween` thuộc `userRepository` để lấy số liệu.

- Cuối cùng, ta gửi số liệu này cùng với khoảng thời gian tương ứng về cho front-end. Tại front-end, ta sử dụng thư viện chart.js cùng với dữ liệu nhật được từ back-end để vẽ biểu đồ.

4.4.5.3 Vẽ biểu đồ tổng quan về tiền nạp vào hệ thống

- Hệ thống sẽ dựa vào số liệu số tiền người dùng nạp vào hệ thống theo từng khung thời gian(tuần, tháng, năm) để vẽ ra biểu đồ. Từ đó, người quản trị có thể có cái nhìn tổng quan về quá trình sử dụng dụng hệ thống
- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/management/statistic-money>) sử dụng phương thức GET cho phép front-end có thể lấy về số liệu để vẽ biểu đồ.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-20: Bảng trạng thái HTTP cho việc vẽ biểu đồ tổng quan về tiền nạp vào hệ thống

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
3	403	Đường dẫn bị cấm
4	404	Không tìm thấy khoảng thời gian
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Được điều hướng và tiếp nhận tại phương thức `statisticMoney` trong `ManagementController` (annotation `@GetMapping("/management/statistic-money")`).

- Phương thức nhận vào tham số là type (WEEK, MONTH, YEAR). Controller sẽ gọi hàm getStatisticMoney từ Service. Ta tiếp tục gọi phương thức getListLocalDateTime truyền vào type ta nhận được một danh sách gồm thời gian bắt đầu và ngày kết thúc của từng khung thời gian. Với từng ngày bắt đầu và kết thúc trong danh sách, ta gọi getSumAmountByStatus thuộc TransactionMoneyRepository để lấy số liệu.
- Cuối cùng, ta gửi số liệu này cùng với khoảng thời gian tương ứng về cho front-end. Tại front-end, ta sử dụng thư viện chart.js cùng với dữ liệu nhận được từ back-end để vẽ biểu đồ.

4.4.5.4 Vẽ biểu đồ tổng quan về số lượng tài liệu đã gửi hoặc hoàn thành

- Hệ thống sẽ dựa vào số liệu lượng tài liệu được gửi đi trong hệ thống theo từng khung thời gian(tuần, tháng, năm) để vẽ ra biểu đồ. Từ đó, người quản trị có thể có cái nhìn tổng quan về quá trình sử dụng dụng hệ thống.
- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/document/management/statistic-contract>) sử dụng phương thức GET cho phép front-end có thể lấy về số liệu để vẽ biểu đồ.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-21: Bảng trạng thái HTTP cho việc vẽ biểu đồ tổng quan về số lượng tài liệu đã gửi

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
3	403	Đường dẫn bị cấm
4	404	Không tìm thấy khoảng thời gian

5	422	Không tìm thấy thẻ loại
6	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại Document Service. Tại Document Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức statisticContract trong ManagementController (ta cần hình annotation @GetMapping("/management/statistic-contract")).
- Phương thức nhận vào tham số là type (WEEK, MONTH, YEAR, ALL). Controller sẽ gọi hàm getStatisticContract từ Service. Ta tiếp tục gọi phương thức getListLocalDateTime truyền vào type ta nhận được một danh sách gồm này bắt đầu và ngày kết thúc của từng khung thời gian. Với từng ngày bắt đầu và kết thúc trong danh sách, ta gọi countAllByCreatedDateBetween thuộc contractRepository để lấy số liệu.
- Cuối cùng, ta gửi số liệu này cùng với khoảng thời gian tương ứng về cho front-end. Tại front-end, ta sử dụng thư viện chart.js cùng với dữ liệu nhận được từ back-end để vẽ biểu đồ.

4.4.6 Báo cáo cho quản trị viên về quá trình truy cập và dung lượng của từng người dùng

4.4.6.1 Báo cáo về lịch sử giao dịch người dùng

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/manage/transactions>) sử dụng phương thức GET cho phép admin xem lịch sử giao dịch của từng người sử dụng.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-22: Bảng trạng thái HTTP cho việc báo cáo lịch sử giao dịch người dùng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
-----	--------------------	---------------------------

1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	403	Đường dẫn bị cấm
4	419	Thiếu trường dữ liệu truy vấn
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User-Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức findAllTransactions trong ManagementController (ta cấu hình annotation @GetMapping(value = "/management/transactions").
- Phương thức này cần truyền vào các tham số dạng truy vấn gồm mã định danh người dùng(id), trang hiện tại (page, mặc định 1), kích thước mỗi trang(size, mặc định 4), trường dữ liệu sắp xếp (sort_field, mặc định là ngày giao dịch), kiểu sắp xếp tăng dần hay giảm dần(sort_type, mặc định là tăng dần). Controller sẽ gọi hàm getTransactionManagementList từ service với tham số truyền vào là các biến truy vấn ở trên. Phương thức này sẽ sử dụng hàm findAll, và trả về kết quả là mảng lịch sử giao dịch, kèm theo các thông số về phân trang được định nghĩa trong lớp DTOList.

4.4.6.2 Báo cáo về số lượng tài liệu của người dùng

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/document/manage/count-all-contract>) sử dụng phương thức GET cho phép admin xem số lượng tài liệu (đã hoàn thành, chờ ký, cần ký, đã xóa) của từng người sử dụng.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-23: Bảng trạng thái HTTP cho việc báo cáo số lượng tài liệu người dùng

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
3	403	Đường dẫn bị cấm
4	419	Thiếu trường dữ liệu truy vấn
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại Document Service. Tại đây yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức countContracts trong ManagementController (ta có hình annotation `@GetMapping("/count-all-contract")`). Phương thức này cần truyền vào các tham số dạng truy vấn là mã định danh người dùng(id). Controller sẽ gọi hàm countAllContractWithAnyStatus từ service với tham số truyền vào là mã định danh người dùng đã nhận được ở trên.
- Phương thức này sẽ gọi hàm countAllByUserAndStatus theo bốn trạng thái (đã hoàn thành, chờ ký, cần ký, đã xóa) và trả về cho người dùng thông qua lớp SummaryContractDTO.

4.4.7 Báo cáo cho người dùng về quá trình truy cập và dung lượng sử dụng của tài khoản

4.4.7.1 Báo cáo về lịch sử giao dịch

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/user/transactions>) sử dụng phương thức GET cho phép người dùng xem lịch sử giao dịch của mình.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-24: Bảng trạng thái HTTP cho việc báo cáo lịch sử giao dịch

STT	Mã trạng thái HTTP	Thông điệp đèn người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
4	419	Thiếu trường dữ liệu truy vấn
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại User Service. Tại User-Service, yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức findAllTransactions trong UserController (ta cấu hình annotation `@GetMapping(value = "/transactions")`).
- Phương thức này cần truyền vào các tham số dạng truy vấn gồm trang hiện tại (page, mặc định 1), kích thước mỗi trang(size, mặc định 4), trường dữ liệu sắp xếp (sort_field, mặc định là ngày giao dịch), kiểu sắp xếp tăng dần hay giảm dần(sort_type, mặc định là giảm dần).
- Về thông tin người dùng hiện tại sẽ được lấy ra trong userDetails. Controller sẽ gọi hàm getTransactionManagementList từ service với tham số truyền vào là các biến truy vấn ở trên và thông tin người dùng hiện tại. Phương thức này sẽ sử dụng hàm findAll, và trả về kết quả là mảng lịch sử giao dịch, kèm theo các thông số về phân trang được định nghĩa trong lớp DTOList.

4.4.7.2 Báo cáo về số lượng tài liệu

- Hệ thống cung cấp API với endpoint (<https://api.vtsign.tech/document/count>) sử dụng phương thức GET cho phép người dùng xem số lượng tài liệu (đã hoàn thành, chờ ký, cần ký, đã xóa) mình.
- Yêu cầu sẽ được xử lý và trả về kết quả theo bảng mô tả trạng thái dưới đây.

Bảng 4-25: Bảng trạng thái HTTP cho việc báo cáo số lượng tài liệu

STT	Mã trạng thái HTTP	Thông điệp đến người dùng
1	200	Thành công
2	400	Thiếu trường hoặc accessToken
4	419	Thiếu trường dữ liệu truy vấn
5	500	Lỗi hệ thống

- Yêu cầu được xử lý chính tại Document Service. Tại đây yêu cầu sẽ được điều hướng và tiếp nhận tại phương thức countContracts trong DocumentController (ta cấu hình annotation `@GetMapping("/count")`). Phương thức này không cần nhận bất kỳ tham số nào.
- Về thông tin người dùng hiện tại sẽ được lấy ra trong userDetails. Controller sẽ gọi hàm countAllContractWithAnyStatus từ service với tham số truyền vào là mã định danh người dùng được lấy ra từ trong userDetails. Phương thức này sẽ gọi hàm countAllByUserAndStatus theo bốn trạng thái (đã hoàn thành, chờ ký, cần ký, đã xóa) và trả về cho người dùng thông qua lớp SummaryContractDTO.

Chương 5

TỔNG KẾT VÀ ĐÁNH GIÁ

5.1 Kiến thức thu được

Trong thời gian thực hiện khóa luận, nhóm đã học tập cải thiện bổ sung được thêm nhiều kiến thức. Các kiến thức chuyên môn, kinh nghiệm làm việc nhóm, xử lý các vấn đề gặp phải, giải quyết mâu thuẫn, trao đổi và phát triển dự án, bao gồm:

- Kỹ năng trình bày báo cáo và viết tài liệu một cách bài bản khoa học. Thực hiện đúng các quy chuẩn đề ra. Trích dẫn tài liệu tham khảo cụ thể theo chuẩn, khả năng tổng hợp các kiến thức trên các sách, bài báo về đề tài.
- Khả năng làm việc nhóm, giao tiếp giữa các thành viên. Kỹ năng lên kế hoạch cụ thể theo từng quá trình, phân chia nhiệm vụ hợp lý phù hợp với khả năng của từng thành viên trong nhóm.
- Áp dụng phương pháp quản lý công việc bằng mô hình Kanban, sử dụng công cụ Trello xuyên suốt quá trình thực hiện. Tổng quát rõ ràng công việc, bao gồm các cột tương ứng với trạng thái quá trình đang thực hiện.
- Tinh thần trách nhiệm công việc được giao, thử sức ở nhiều vai trò khác nhau khi xây dựng hệ thống gồm quản lý dự án, kiểm thử, thiết kế giao diện. Tác phong chuyên nghiệp, trình bày báo cáo cho trưởng nhóm và giảng viên hướng dẫn.
- Học hỏi và áp dụng kiến trúc microservice, mô hình CI/CD, thiết kế giao diện sử dụng thư viện mã nguồn mở JavaScript. Khái niệm và kiến thức liên quan đến chữ ký điện tử, chữ ký số, chuẩn hash và thuật toán mã hóa. Quá trình và chức năng đầy đủ của một hệ thống ký kết văn bản.
- Khả năng học hỏi công nghệ và kiến thức mới, kỹ năng viết mã nguồn, tìm kiếm và sửa lỗi, bảo trì và nâng cấp. Kinh nghiệm khắc phục lỗi khi gặp sự cố và triển khai sản phẩm cho người dùng cuối.

5.2 Sản phẩm thu được

5.2.1 Môi trường phát triển

- Hệ điều hành: Windows, Linux
- Công cụ xây dựng và phát triển phần mềm: Visual Studio Code, IntelliJ
- Các thư viện và nền tảng sử dụng:

Bảng 5-1: Các thư viện và nền tảng sử dụng

Tên thư viện / nền tảng	Tóm tắt chức năng
ReactJs	Thư viện JavaScript mã nguồn mở giúp xây dựng giao diện người dùng
NodeJs	Một hệ thống phần mềm được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là máy chủ web
Material UI	Một thư viện React Component
Npm	Chương trình tạo và quản lý thư viện JavaScript trong môi trường NodeJs
Maven	Công cụ quản lý và thiết lập tự động 1 dự án phần mềm, chủ yếu dùng cho lập trình viên Java

5.2.2 Môi trường triển khai

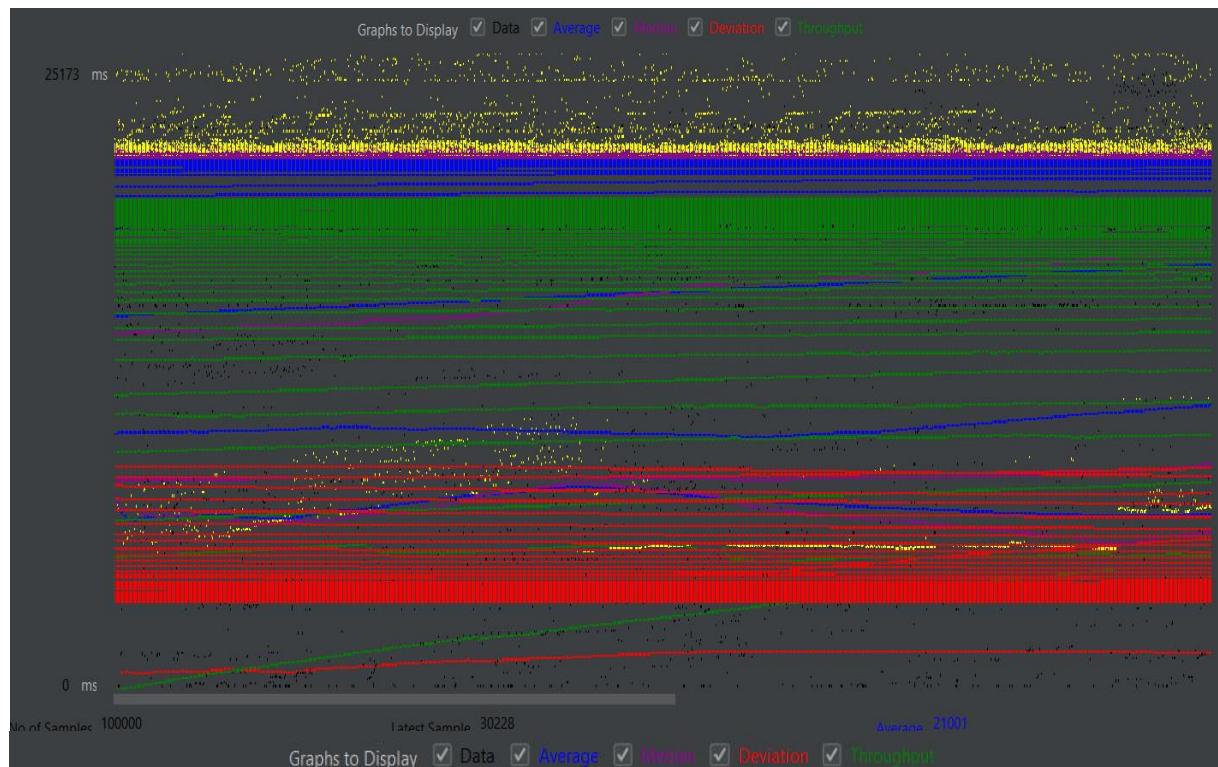
Digital Ocean - nhà cung cấp cơ sở hạ tầng đám mây của Mỹ có trụ sở chính tại Thành phố New York với các trung tâm dữ liệu trên toàn thế giới. DigitalOcean cung cấp cho các nhà phát triển dịch vụ đám mây giúp triển khai và mở rộng quy mô các ứng dụng chạy đồng thời trên nhiều máy tính.

5.2.3 Hiệu năng hệ thống

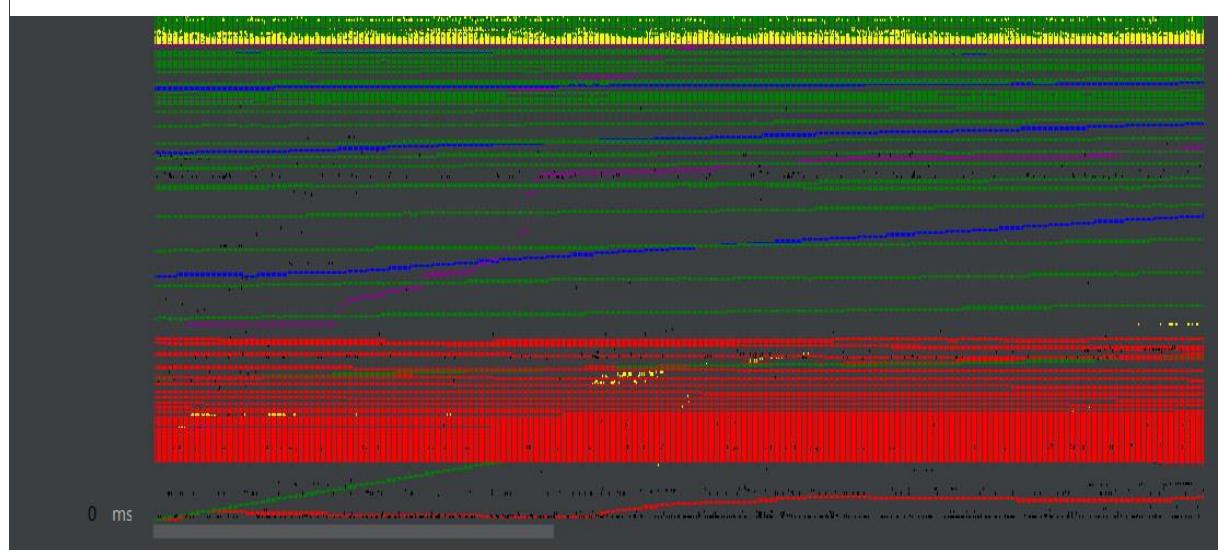
Kịch bản kiểm thử:

Trường hợp hệ thống có từ 1 đến 1 triệu người dùng. Mỗi người khi sử dụng hệ thống sẽ dùng 3 lần chức năng ký kết chính. Mỗi người sẽ ký 3 tài liệu, mỗi người dùng chạy 3 lần. Tập tin kèm theo dung lượng kèm theo là 1,4mb.

- **Hệ thống có từ 100.000 - 500.000 người dùng vào sử dụng**



Hình 5-2: Kết quả kiểm thử với 100.000 người dùng truy cập

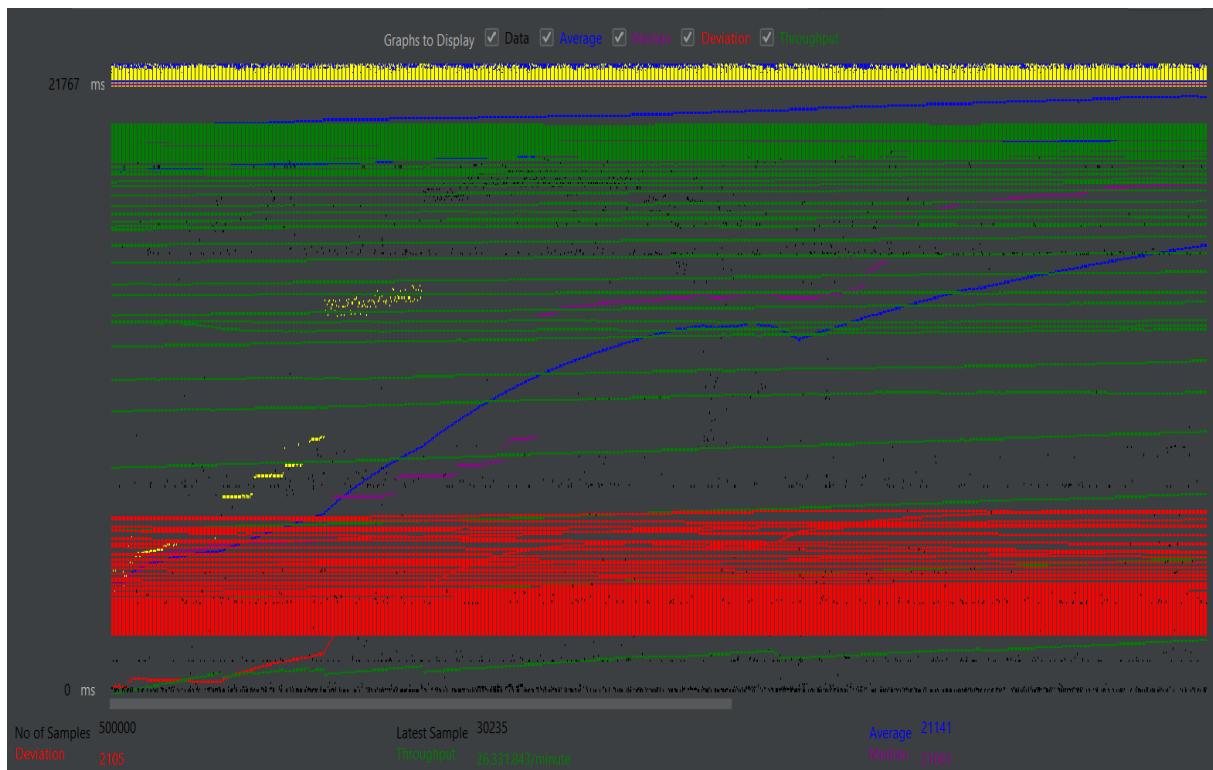


Hình 5-1: Kết quả kiểm thử với 250.000 người dùng truy cập

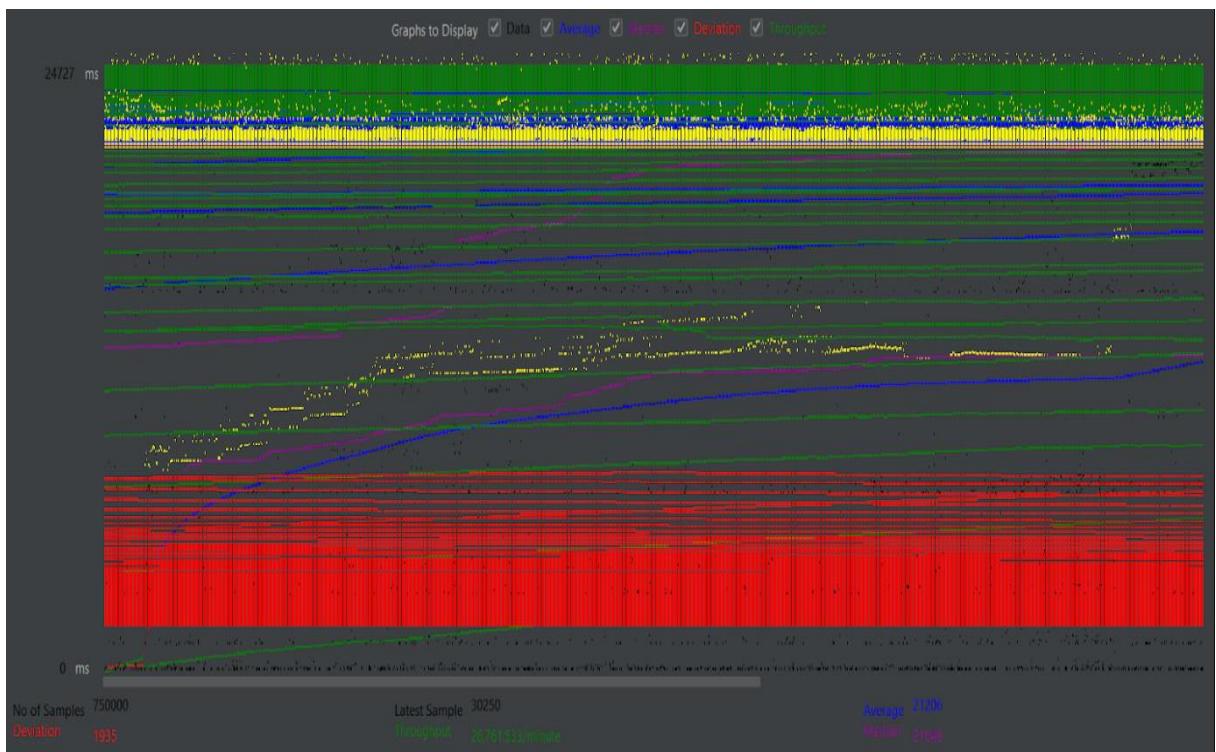
- Qua dữ liệu các biểu đồ trên ta nhận thấy rằng**

- Với số lượng 100.000 user thì số số lượng request mà server có thể xử lý trong một phút chỉ khoảng 18000, với số lượng user là 250.000 và 500.000 thì con số này là khoảng 26000.
- Tương tự, kết quả về độ sai lệch(Dev) thu được với số lượng user 100.000 có kết quả không tốt hơn so với số lượng user 250.000 và 500.000.
- Về thời gian phản hồi trung bình của tất cả các request ở 3 kịch bản kiểm thử trên đều cho kết quả có độ chênh lệch không lớn.

- Nhìn chung, hệ thống có chất lượng tốt khi số lượng user tăng lên**
- Hệ thống có từ trên 500.000 - 1.000.000 người dùng vào sử dụng**



Hình 5-3: Kết quả kiểm thử với 500.000 người dùng truy cập



Hình 5-5: Kết quả kiểm thử với 750.000 người dùng truy cập



Hình 5-4: Kết quả kiểm thử với 1.000.000 người dùng truy cập

- **Qua dữ liệu thu được từ đô thị, ta thấy hệ thống hoạt động tốt khi số lượng user tăng lên**
 - Độ sai lệch khá nhỏ (1935 với kịch bản 750.000 user và 1942 với kịch bản 750.000 user).
 - Số lượng request mà server có thể xử lý cho kết quả tốt, khoảng 26000 request/phút (kết quả có sự tương đồng so với các kịch bản từ 100.000 đến 500.000 user)
 - Về thời gian phản hồi trung bình của tất cả các request ở 2 kịch bản kiểm thử trên đều cho kết quả có độ chênh lệch không lớn.

5.2.4 Các chức năng đã cài đặt

Bảng 5-2: Bảng chức năng đã hoàn thành

Đăng ký
Đăng nhập
Tạo tài liệu, tải lên một tài liệu để ký kết
Nhập thông tin người nhận
Nhập tin nhắn riêng tư cho từng người nhận
Cài đặt quyền cho người nhận
Ký tài liệu đã tải lên
Cài đặt tài liệu
Nhập chủ đề, thông điệp của tài liệu
Gửi lời mời ký kết cho các đối tác qua email
Gửi tin nhắn chứa mã khóa (code) cho đối tác qua SMS OTP

Nhận được email có chứa đường dẫn đến tài liệu cần ký
Thông báo qua mail cho tôi khi đối tác đã xem tài liệu
Nhập mã Code trước khi ký tài liệu.
Ký tài liệu được nhận mà không cần phải tạo tài khoản hệ thống
Ký tài liệu được nhận trong mục quản lý
Nhận được email thông báo đối tác đã ký tài liệu
Quản lý các tài liệu đã hoàn thành
Xem chi tiết tài liệu hoàn thành
Quản lý các tài liệu chờ ký
Xem chi tiết tài liệu đang chờ ký
Quản lý các tài liệu cần ký
Xem chi tiết tài liệu cần ký
Quản lý các tài liệu đã xóa
Xem chi tiết tài liệu đã xóa
Xem tổng quát các tài liệu yêu cầu của mình về số lượng
Đăng xuất
Xem thông tin tài khoản của tôi
Thay đổi thông tin tài khoản
Nạp tiền vào tài khoản theo nhiều hình thức khác nhau

Cho phép người sử dụng dùng thử dịch vụ
Hiển thị số tiền còn lại của tôi
Hiển thị lịch sử giao dịch của tôi
Tạo mới một tài khoản cho hệ thống
Xem danh sách tất cả người sử dụng
Khóa/Mở khóa người sử dụng
Xóa/Khôi phục người sử dụng
Cập nhật thông tin cho từng người dùng (thay đổi quyền, thay đổi thông tin cơ bản)
Xem chi tiết người dùng (thông tin, số lượng tài liệu, quá trình thanh toán)
Thống kê lượng tiền nạp vào hệ thống theo (ngày, tuần, tháng, năm, tất cả)
Vẽ biểu đồ tổng quan về tiền nạp vào hệ thống theo (tuần, tháng, năm, tất cả)
Thống kê số lượng người đăng ký sử dụng hệ thống theo (ngày, tuần, tháng, năm, tất cả)
Vẽ biểu đồ tổng quan về số lượng người đăng ký sử dụng hệ thống (tuần, tháng, năm, tất cả)
Thống kê số lượng tài liệu đã gửi/hoàn thành theo (ngày, tuần, tháng, năm, tất cả)
Vẽ biểu đồ tổng quan về số lượng tài liệu đã gửi/hoàn thành theo (tuần, tháng, năm, tất cả)

Thông tin giới thiệu website
Cải thiện UI/UX trang ký kết
Hiển thị trang thông báo người dùng vào mail xác nhận tài khoản đã đăng ký của họ
Lấy lại mật khẩu khi quên
Kiểm tra lại sự hợp lệ của các trường dữ liệu ở trang đăng ký
Hiện thị lỗi cho người dùng
Chỉnh sửa footer trang chính
Chỉnh sửa giao diện (tất cả các trang) cho phù hợp với kích thước màn hình điện thoại, máy tính bảng
Xem lịch sử ký kết
Admin xem chi tiết user nút riêng
Ký nhiều hợp đồng liên tiếp
Gửi lại liên kết kích hoạt tài khoản

5.2.5 So sánh tính năng với các hệ thống khác trên thị trường

Bảng 5-3: Bảng so sánh tính năng của VTSign với hệ thống khác

Tính năng	SignNow	AdobeSign	DocuSign	VTSign
Gửi tài liệu cần ký	x	x	x	x
Mời ký số lượng lớn	x	x	x	x

Chỉnh sửa tài liệu trước khi gửi	X	X		
Tạo các mẫu có thể sử dụng lại	X	X	X	
Dùng thử hệ thống toàn bộ tính năng				X
Gửi lời mời qua email	X	X	X	X
Ký kết hợp đồng không cần tài khoản	X			X
Ký kết trực tiếp	X			X
Thanh toán chi phí người dùng	X	X	X	X
Bảo mật hợp đồng bằng SMS OTP			X	X
Quản lý tài liệu hợp đồng	X		X	X

5.3 So sánh kết quả thu được với mục tiêu ban đầu

Bảng 5-4: Bảng so sánh mục tiêu ban đầu với kết quả thu được

Mục tiêu ban đầu	Nhận xét mức độ hoàn thành
Trình bày lý do xây dựng hệ thống ký kết văn bản trực tuyến VTSign.	Đã trình bày đầy đủ ở chương 1 mục 1.3 của báo cáo đồ án.
Trình bày các tính năng cơ bản của 3 hệ thống ký kết văn bản trực tuyến, phát hiện một số tính năng chưa hoàn hảo và thiếu.	Đã trình bày đầy đủ ở chương 1 mục 1.2 của báo cáo đồ án. Bao gồm bảng so sánh

	chức năng và ưu khuyết điểm của 3 hệ thống so với hệ thống VTSign.
Trình bày tóm tắt lý thuyết nền tảng và giải pháp để xây dựng hệ thống ký kết văn bản trực tuyến. Bao gồm các chủ đề và khái niệm liên quan đến Chữ ký số, Chữ ký điện tử, quy trình ký kết văn bản trong đời sống và bằng máy tính, phương pháp ký kết bằng Digital Signature, phương pháp định danh.	Đã trình bày đầy đủ ở chương 2 của báo cáo đồ án.
Phân hệ web services (Back-end): Cung cấp các API bao gồm: cho phép người dùng đăng ký, và thanh toán chi phí sử dụng dịch vụ, cho phép người quản trị quản lý người sử dụng dịch vụ, xử lý việc quản lý và ký kết văn bản, cung cấp các báo cáo cho admin về quá trình truy cập và báo cáo cho người dùng về quá trình truy cập, dung lượng sử dụng của từng người dùng sử dụng, dung lượng sử dụng của tài khoản. Áp dụng Microservices, mô phỏng, thiết kế, thực hiện, báo cáo kết quả các kịch bản kiểm thử khả năng tải mong muốn (load tests), khả năng chịu tải tối đa (stress tests), khả năng xâm nhập hệ thống (penetration testing).	Đã xây dựng các web services API và đáp ứng các yêu cầu.
Phân hệ web (Front-end): Các giao diện kết nối với các tính năng tương ứng với back-end, đơn giản dễ dàng thao tác và	Đã xây dựng trang web và đáp ứng các yêu cầu.

phù hợp với nhiều đối tượng người dùng sử dụng.	
Đề xuất giải pháp và chỉnh sửa các tính năng chưa hoàn hảo và/hoặc xây dựng tính năng còn thiếu.	
Viết 80 trang báo cáo đồ án theo đúng nội dung trình bày trong tài liệu “Hướng dẫn viết khóa luận hướng ứng dụng hoặc đồ án tốt nghiệp” mà giáo viên cung cấp, theo đúng chuẩn nhà trường yêu cầu và trích dẫn tài liệu tham khảo một cách chi tiết, đầy đủ.	Báo cáo đồ án được viết tương đối đầy đủ và chính xác.

5.4 Định hướng phát triển và nghiên cứu

- Cải thiện và cấu trúc lại toàn bộ mã nguồn của hệ thống trong tương lai, bao gồm các mục đích như tối ưu và bảo trì hay nâng cấp hoặc thêm chức năng mới.
- Cập nhật lại các tài liệu kèm theo gồm báo cáo đồ án, tài liệu nghiên cứu phát triển để phù hợp với công nghệ hiện đại.
- Phát triển ứng dụng trên đa nền tảng hệ điều hành Android và iOS, bên cạnh đó nâng cấp giao diện của hệ thống trên trình duyệt.
- Hoàn thiện hơn về chức năng ký kết chính của hệ thống, sửa một số lỗi còn tồn đọng hoặc đổi phương pháp ký kết văn bản để an toàn và bảo mật hơn.
- Cải thiện tốc độ xử lý các chức năng của ứng dụng, giúp hệ thống chạy trơn tru không gặp vấn đề trong quá trình sử dụng, trải nghiệm tốt hơn cho người dùng.
- Đưa ra các bản cập nhật theo quý hoặc trong trường hợp phát sinh lỗi. Chạy kiểm tra trên các nền tảng triển khai phiên bản mới.

5.5 Lời kết

Báo cáo đồ án hệ thống ký kết văn bản trực tuyến được xây dựng là sản phẩm của một quá trình dài nghiên cứu, học tập của nhóm trong khoảng thời gian 6 tháng làm việc. Hệ thống còn nhiều hạn chế về tốc độ phản hồi, trải nghiệm người dùng, chất lượng dịch vụ, giao diện và các chức năng của hệ thống, nhưng đồ án này đã đem lại cho nhóm được nhiều kiến thức cũng như phương hướng để triển khai một dự án trong thực tế. Kiến thức thu được từ quá trình thực hiện đề tài này sẽ là nền tảng cho nhóm đủ kiến thức để tiếp tục nghiên cứu và phát triển sự nghiệp của bản thân, tự tin phỏng vấn và làm việc cho các công ty về công nghệ.

PHỤ LỤC 1: GIAO DIỆN CHỨC NĂNG VTSIGN

- Dưới đây là hình ảnh giao diện các màn hình chức năng cụ thể của ứng dụng hệ thống VTSign. Khi người dùng truy cập vào trang web vtsign.tech trang giới thiệu hệ thống hiển thị giao diện như sau:



Hình phụ lục 1.1:Trang giới thiệu hệ thống VTSign

- Bao gồm tên đầy đủ hệ thống, 3 giá trị cốt lõi Nhanh Chóng – Đơn giản – Bảo mật

- Nút ĐĂNG KÝ: khi người dùng ấn nút này sẽ được điều hướng tới trang đăng ký tài khoản hệ thống VTSign vtsign.tech/register, dùng để lưu trữ thông tin cá nhân và đăng nhập vào hệ thống.

Đăng ký

Họ và tên đệm * Nhập họ và tên đệm

Tên * Nhập tên

Địa chỉ Email * Nhập địa chỉ email

Mật khẩu * Nhập mật khẩu

Xác thực mật khẩu * Nhập xác thực mật khẩu

Số điện thoại * Nhập số điện thoại

Cơ quan * Nhập cơ quan

Địa chỉ * Nhập địa chỉ

ĐĂNG KÝ

Hình phụ lục 1.2: Trang đăng ký tài khoản VTSign

- Nút ĐĂNG NHẬP: khi người dùng ấn nút này sẽ được điều hướng tới trang đăng nhập vào hệ thống VTSign vtsign.tech/login, để người dùng bắt đầu sử dụng tài khoản của mình.

Đăng nhập

Địa chỉ Email * Vui lòng nhập địa chỉ email

Mật khẩu * Vui lòng nhập mật khẩu

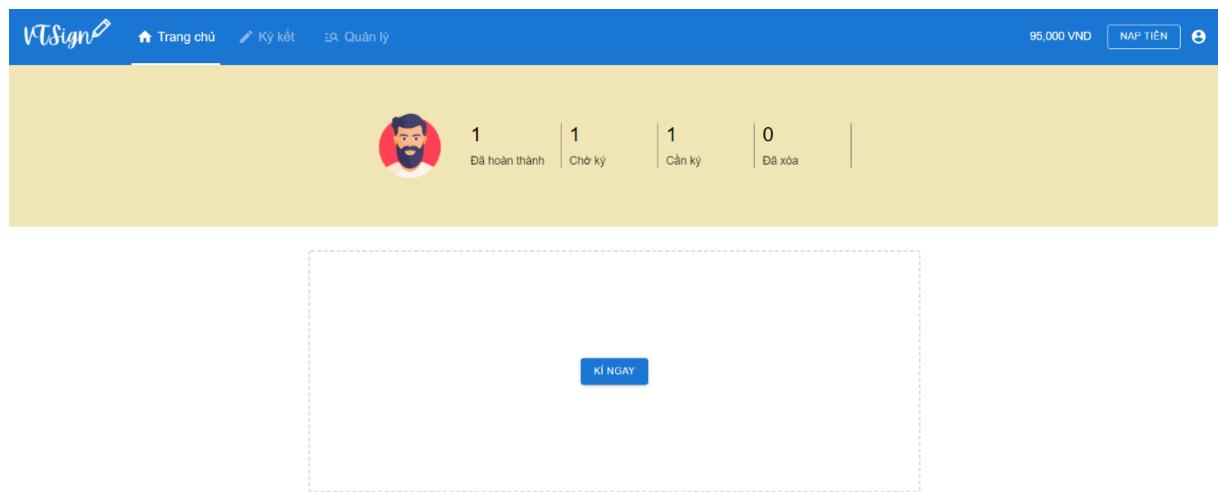
ĐĂNG NHẬP

[Bạn đã quên mật khẩu?](#)

Bạn chưa có tài khoản? [Đăng ký](#).

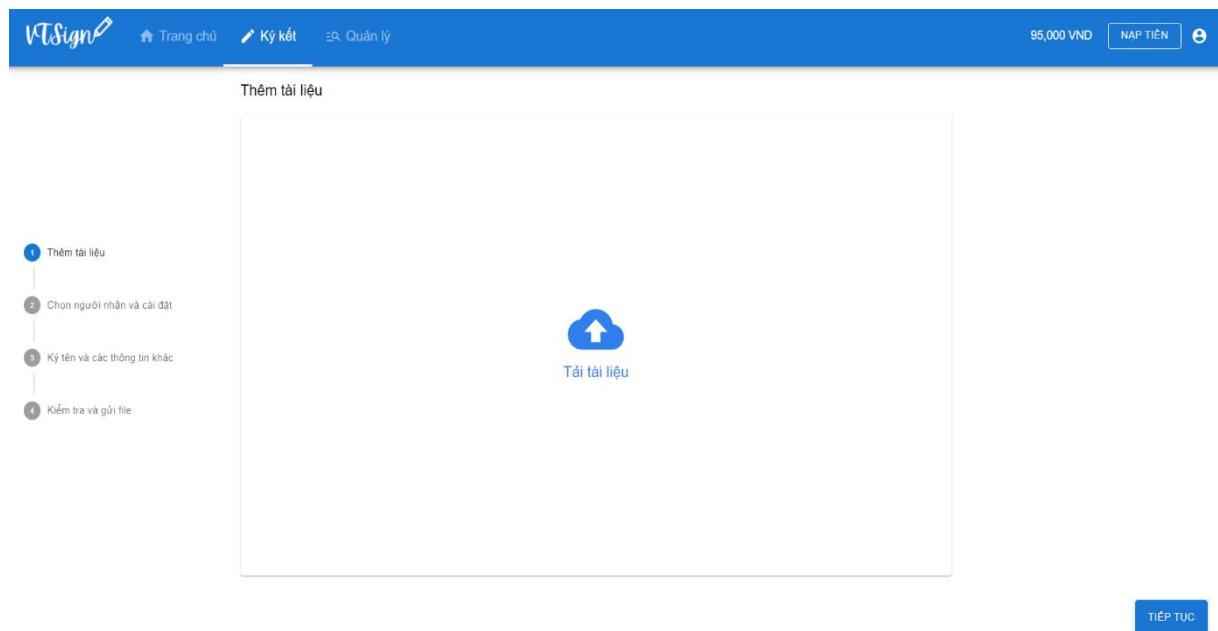
Hình phụ lục 1.3: Trang đăng nhập vào hệ thống VTSign

- Sau khi người dùng đã có tài khoản VTSign, sau khi đăng nhập sẽ được điều hướng tới trang vtsign.tech/home hiển thị trang chủ hệ thống giao diện hiển thị như sau.



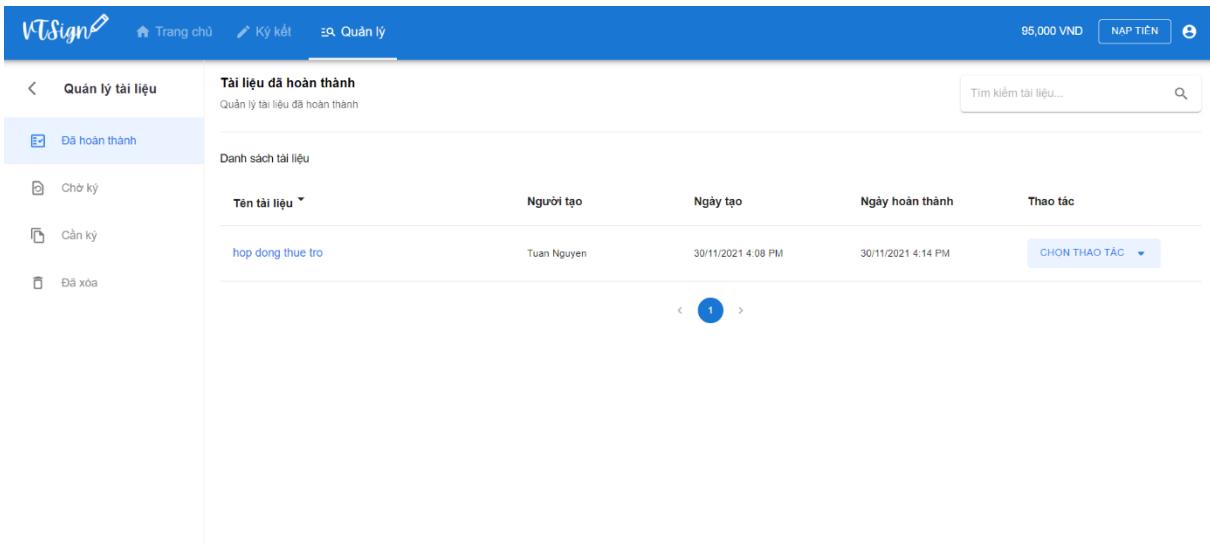
Hình phụ lục 1.4: Trang chủ hệ thống VTSign

- Nút Ký kết: người dùng sẽ được điều hướng tới trang vtsign.tech/signing, là trang chức năng ký kết chính của hệ thống, giao diện như sau.



Hình phụ lục 1.5: Trang ký kết chính của VTSign

- Nút Quản lý: người dùng sẽ được điều hướng tới trang vtsign.tech/manage, là trang quản lý tài liệu của người dùng hệ thống, giao diện như sau.



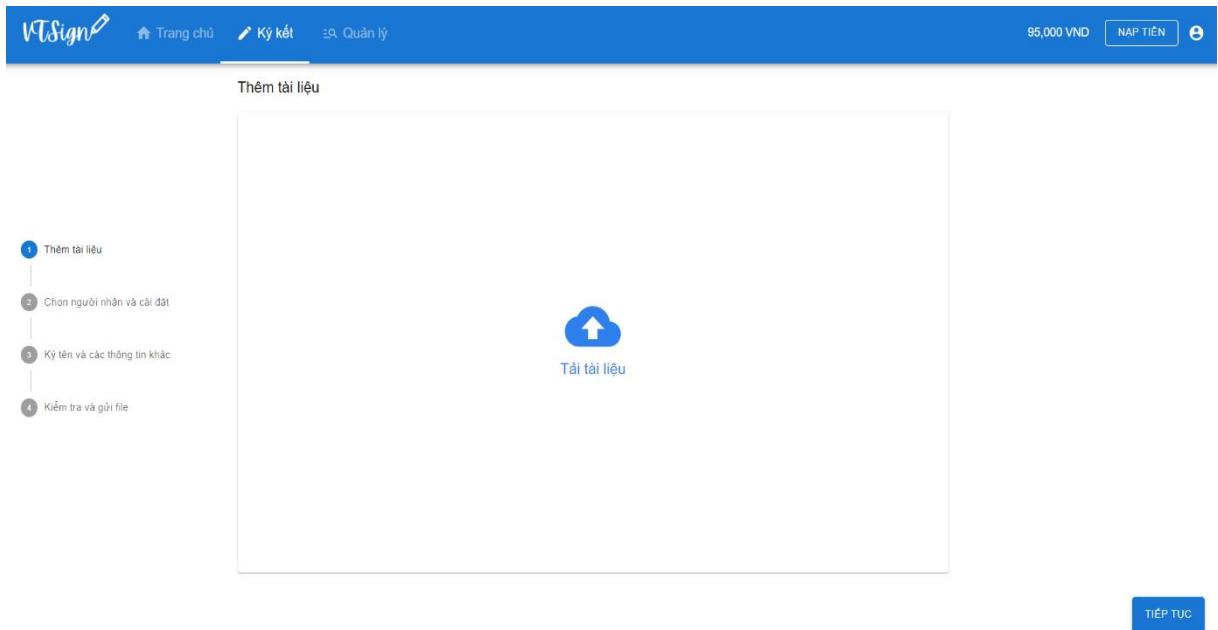
Hình phụ lục 1.6: Trang quản lí tài liệu người dùng VTSign

- Nút NẤP TIỀN: người dùng muốn nạp tiền vào hệ thống để sử dụng lâu dài, với mỗi tài khoản được tạo mới sẽ được 10,000VNĐ dùng thử, với mỗi hợp đồng gửi đi sẽ trừ 5,000VNĐ. Bao gồm nhiều phương thức thanh toán khác nhau, giao diện như sau.

Hình phụ lục 1.7: Trang nạp tiền vào tài khoản VTSign

- Người dùng muốn bắt đầu thực hiện chức năng chính ở trang Ký kết, có thể truy cập từ Trang chủ -> KÍ NGAY hoặc ấn vào nút Ký kết ở tiêu đề ứng dụng. Quá trình ký kết gồm các giai đoạn:

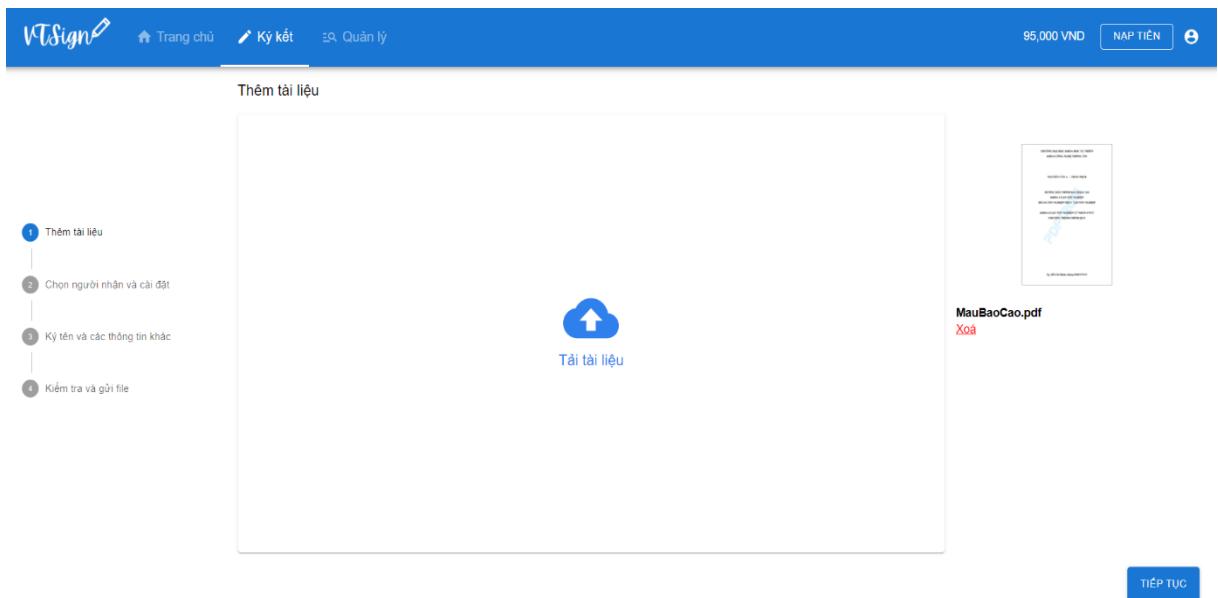
- Bước 1: Truy cập vào trang ký kết



Hình phụ lục 1.8: Trang ký kết chính – Bước 1

- **Bước 2: Tải lên tài liệu**

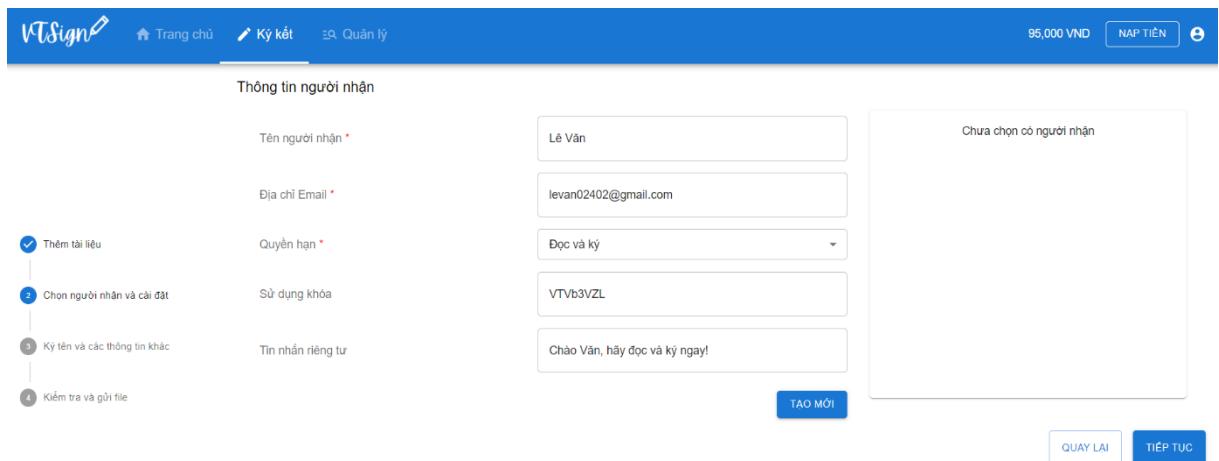
Người dùng thực hiện tải tài liệu theo định dạng PDF theo 2 cách chọn trực tiếp tập tin trên máy tính hoặc kéo tập tin thẳng vào khung.



Hình phụ lục 1.9: Trang ký kết chính – Bước 2

- **Bước 3: Chọn người nhận và cài đặt**

Điền các thông tin của người nhận gồm 3 trường bắt buộc là tên người nhận, địa chỉ email, quyền hạn. Có thể gửi cho nhiều người nhận, người chưa có tài khoản hệ thống cần nhập thêm số điện thoại.



Thông tin người nhận

Tên người nhận *

Lê Văn

Địa chỉ Email *

levan02402@gmail.com

Quyền hạn *

Đọc và ký

Sử dụng khóa

VTVb3VZL

Tin nhắn riêng tư

Chào Văn, hãy đọc và ký ngay!

TAO MỚI

Chưa chọn có người nhận

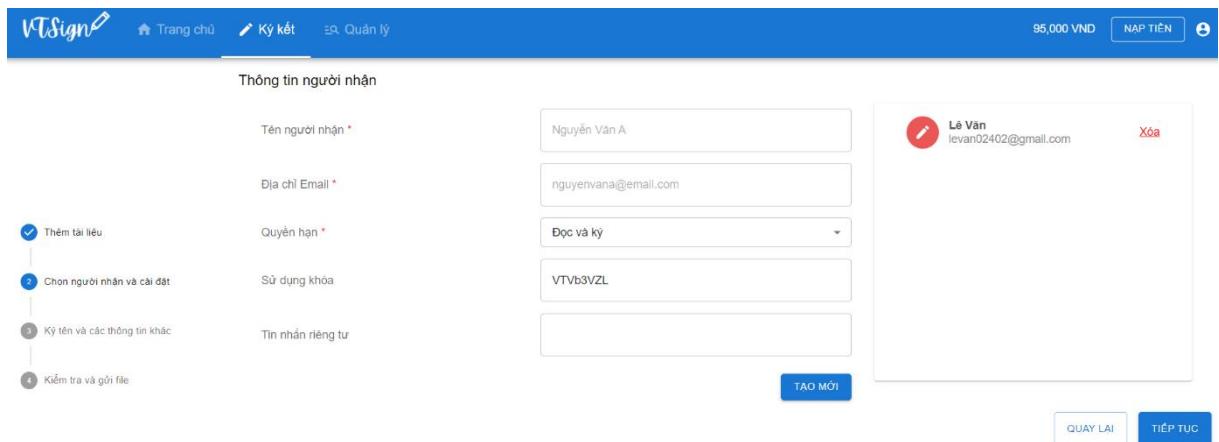
1 Thêm tài liệu
2 Chọn người nhận và cài đặt
3 Ký tên và các thông tin khác
4 Kiểm tra và gửi file

QUAY LẠI

TIẾP TỤC

Hình phụ lục 1.10: Trang ký kết chính – Bước 3

- Bước 4: Án nút TẠO MỚI, tiếp đến án nút TIẾP TỤC



Thông tin người nhận

Tên người nhận *

Nguyễn Văn A

Địa chỉ Email *

nguyenvana@email.com

Quyền hạn *

Đọc và ký

Sử dụng khóa

VTVb3VZL

Tin nhắn riêng tư

TAO MỚI

Lê Văn
levan02402@gmail.com

Xóa

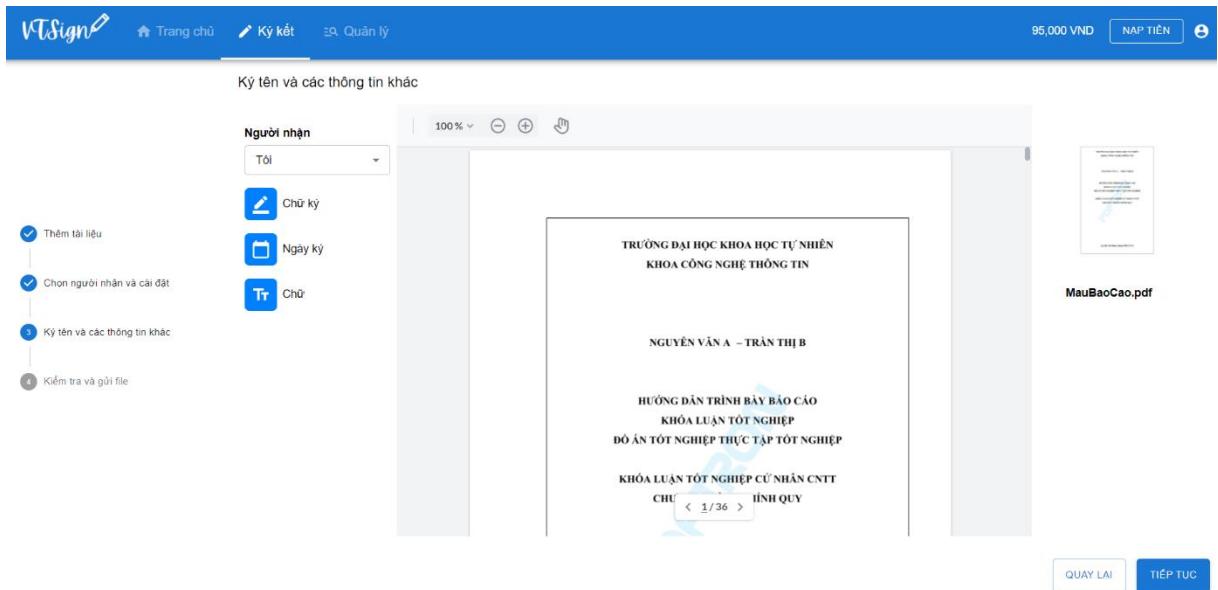
1 Thêm tài liệu
2 Chọn người nhận và cài đặt
3 Ký tên và các thông tin khác
4 Kiểm tra và gửi file

QUAY LẠI

TIẾP TỤC

Hình phụ lục 1.11: Trang ký kết chính – Bước 4

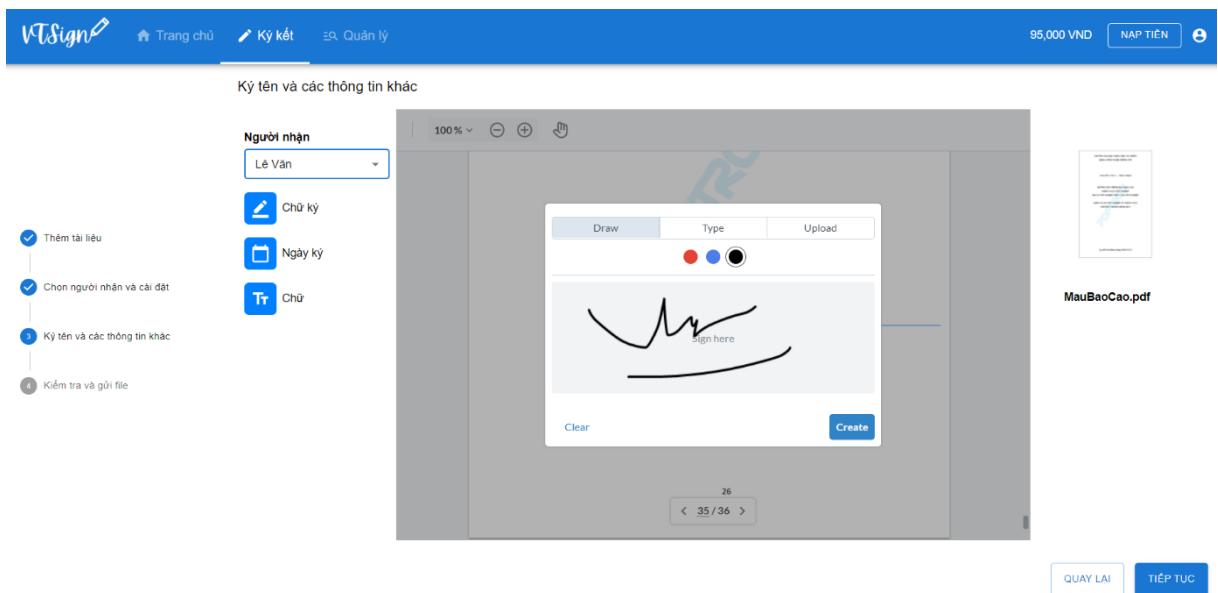
- Bước 5: Tới quá trình Ký tên và các thông tin khác



Hình phụ lục 1.12: Trang ký kết chính – Bước 5

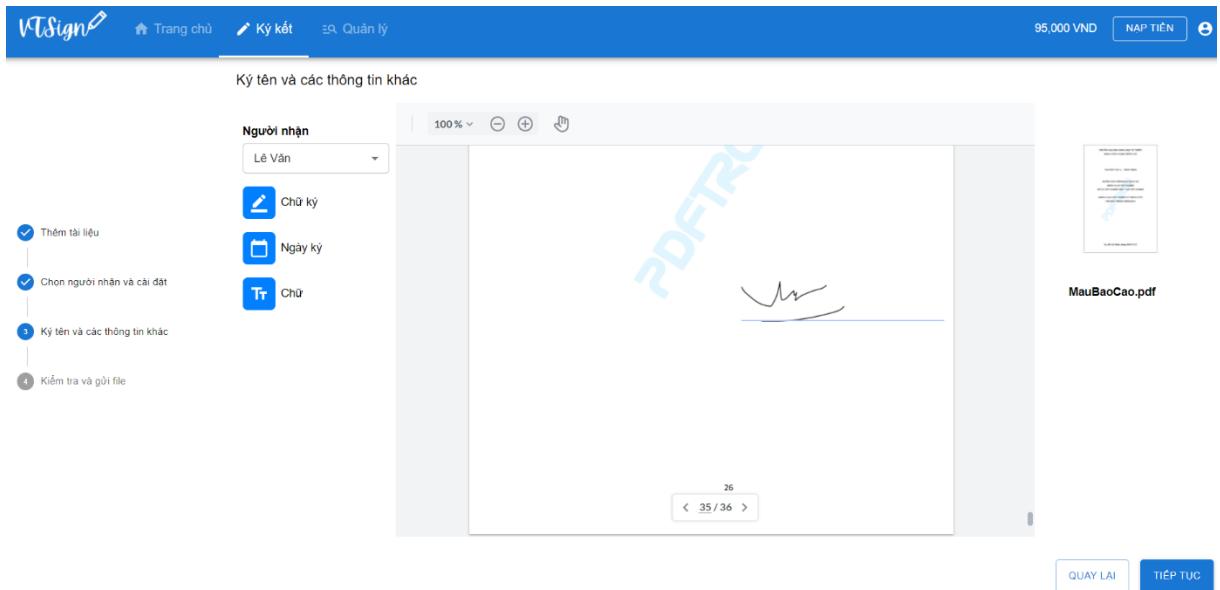
- **Bước 6: Cài đặt tài liệu ký kết**

Thiết lập gửi cho chính mình hoặc người nhận. Thực hiện kéo các trường vào tập tin PDF như chữ ký, ngày ký, thương hiệu... Đối với chữ ký có thể vẽ trực tiếp hoặc chọn kiểu, có thể tải lên chữ ký cá nhân lưu ở máy tính.



Hình phụ lục 1.13: Trang ký kết chính – Bước 6

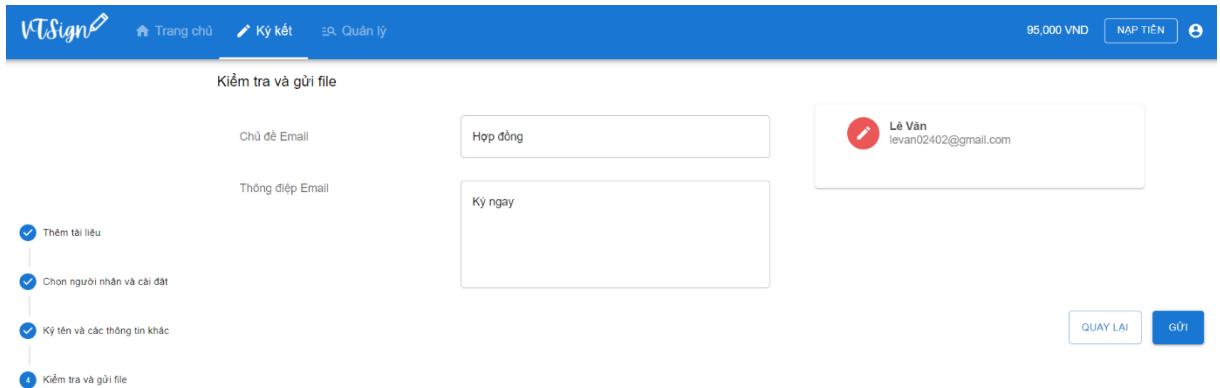
- **Bước 7: Án Create, sau đó án TIẾP TỤC để qua quá trình tiếp theo**



Hình phụ lục 1.14: Trang ký kết chính – Bước 7

- **Bước 8: Kiểm tra và gửi file**

Bước này người dùng điền 2 thông tin gồm chủ đề email và thông điệp của email để gửi cho người nhận. Sau đó ấn nút GỬI, quá trình hoàn tất.



Hình phụ lục 1.15: Trang ký kết chính – Bước 8

Sau đó hệ thống sẽ thông báo tiến độ hoàn thành việc ký kết và điều hướng trở lại trang chủ để theo dõi trạng thái.



Hình phụ lục 1.15 Trang ký kết chính – Kết thúc

TÀI LIỆU THAM KHẢO

- [1] "signNow: eSign PDF with Electronic Signature Online," airSlate Inc, 2021. [Online]. Available: <https://www.signnow.com/>. [Accessed 04 October 2021].
- [2] "DocuSign | #1 in Electronic Signature and Agreement Cloud," DocuSign Inc, 2021. [Online]. Available: <https://www.docusign.com/>. [Accessed 04 October 2021].
- [3] "E-signatures & digital signing software - Adobe," Adobe Inc, 2021. [Online]. Available: <https://www.adobe.com/sign.html>. [Accessed 04 October 2021].
- [4] "Nafosted - Quỹ phát triển khoa học và công nghệ Quốc gia," NAFOSTED, 2021. [Online]. Available: <https://nafosted.gov.vn/thong-bao-trien-khai-ky-dien-tu-doi-voicac-hop-dong-thuc-hien-de-tai-nhiem-vu-hoat-dong-do-quy-phat-trien-khoa-hoc-va-cong-nghe-quoc-gia-tai-tro-ho-tro/>. [Accessed 24 11 2021].
- [5] J. Katz, Digital signatures, Springer Science & Business Media, 2010.
- [6] Adams, Carlisle, Lloyd and Steve, Understanding PKI: concepts, standards, and deployment considerations, Addison-Wesley Professional, 2003.
- [7] D. Pinkas, Integris, J. Ross and N. Pope, Electronic Signature Formats for long term electronic signatures, The Internet Society, 2001.

