



**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. 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 thể 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.**

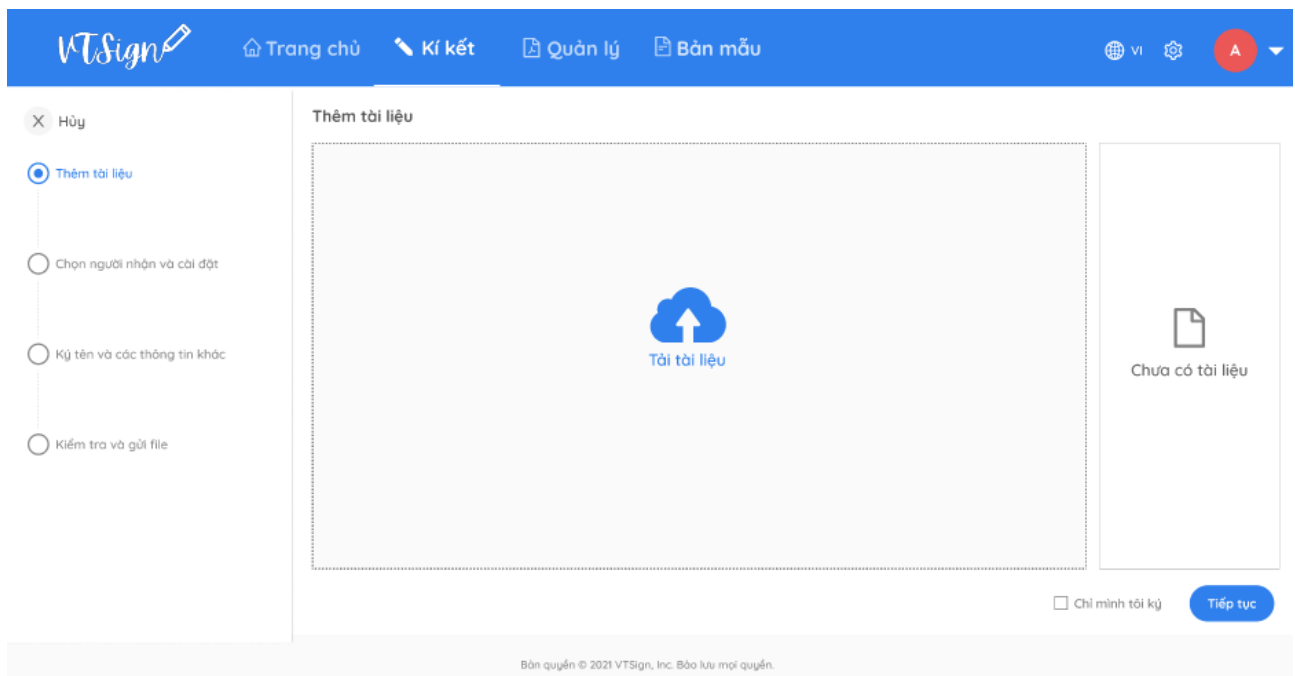
## 2.3 Phạm vi của đề tài

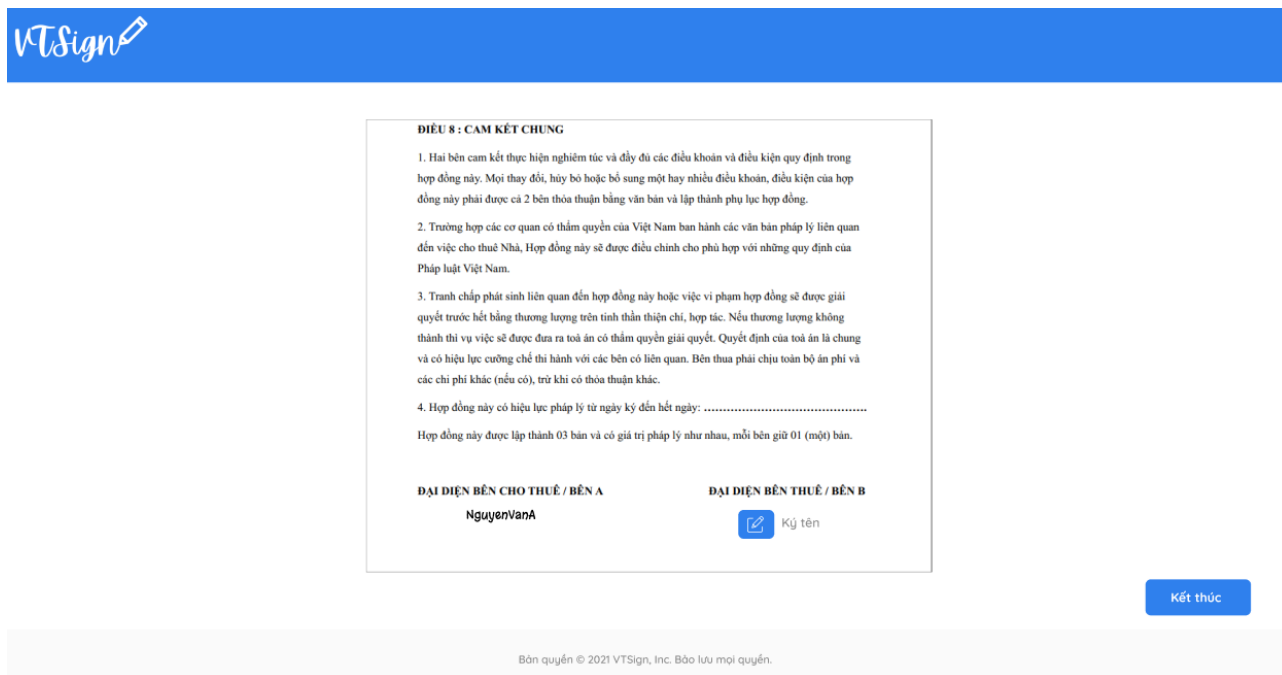
### Các tính năng không thực hiện

- 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

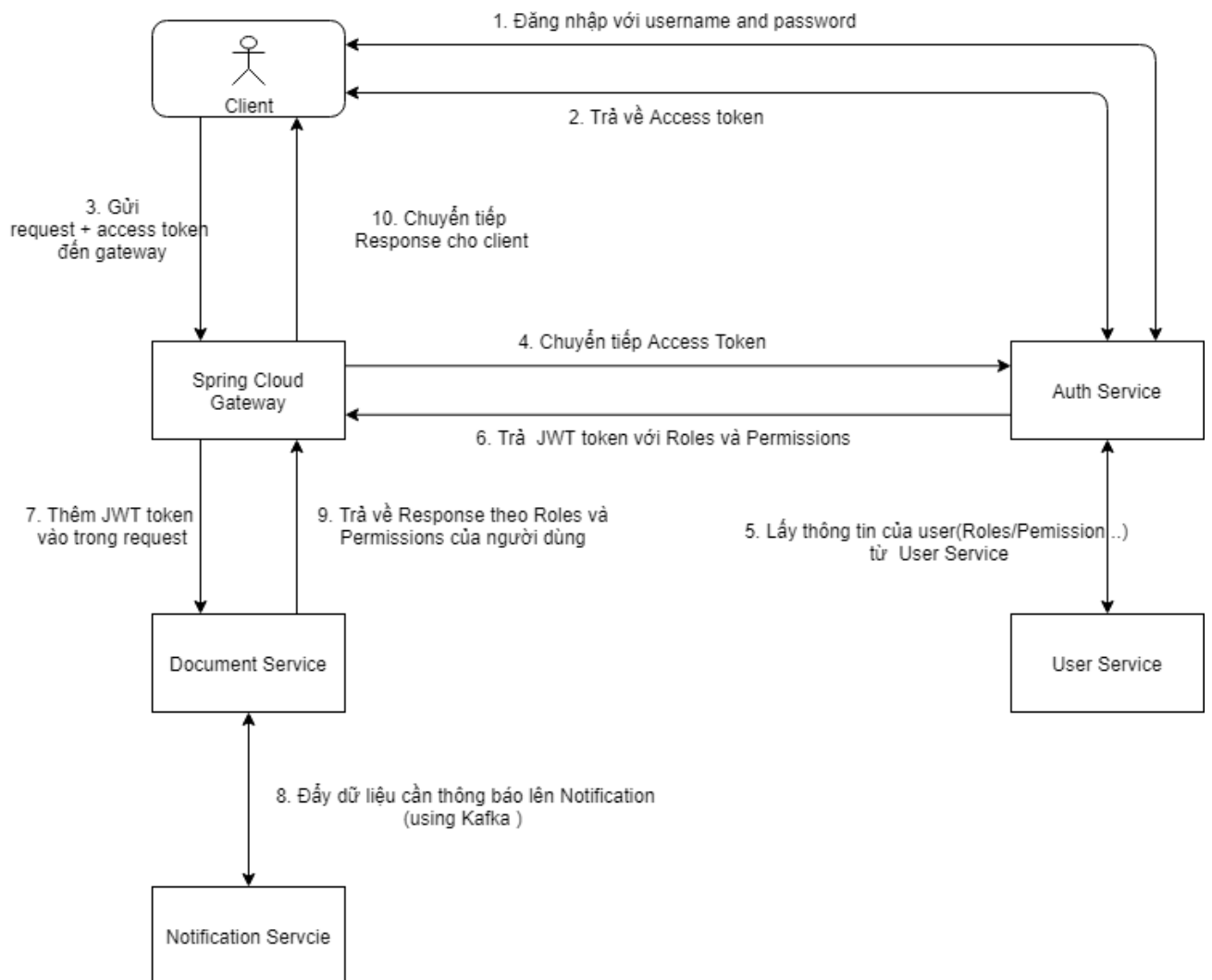
\* **Bản mẫu:** demo các trang chức năng chính của website.





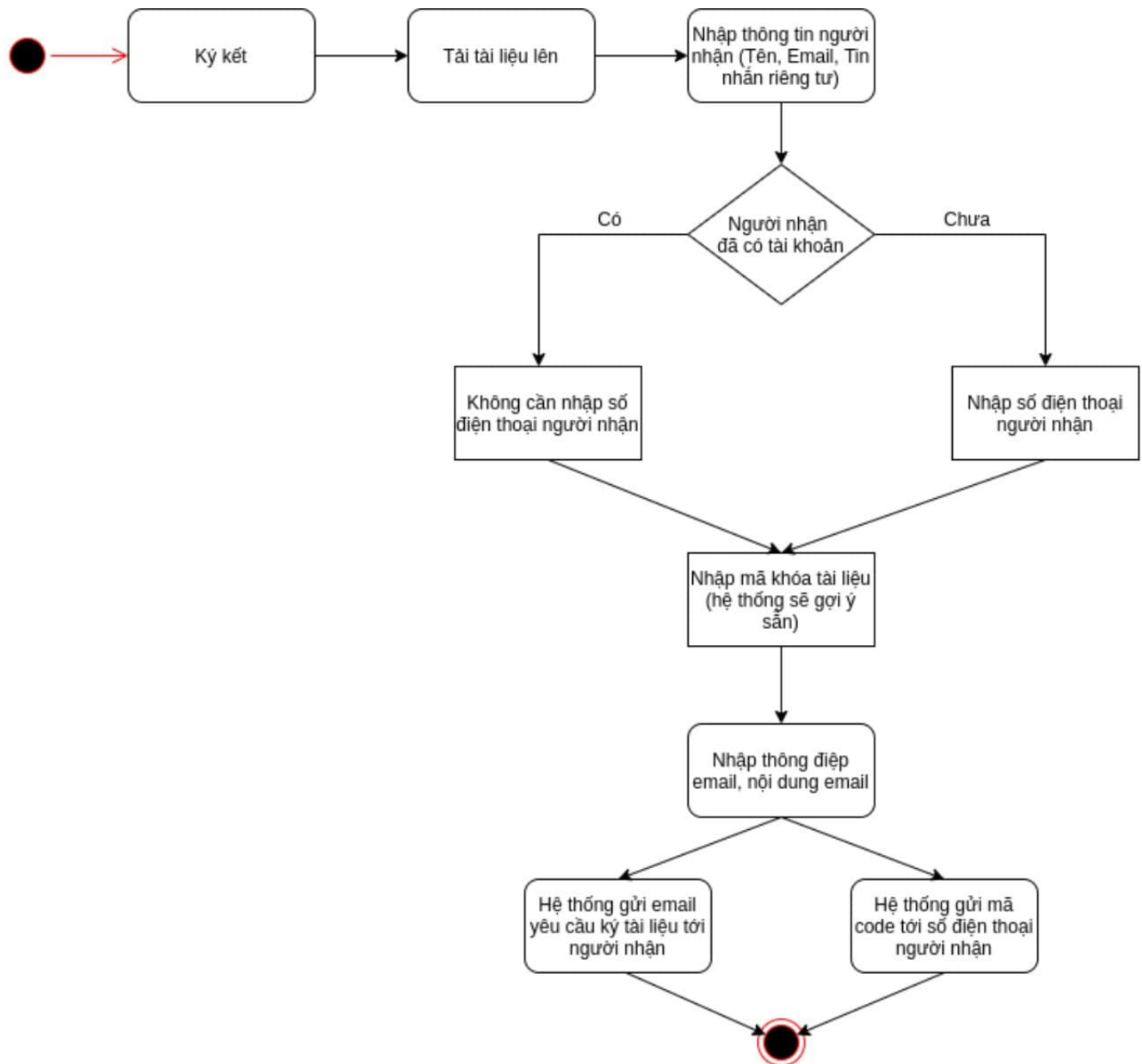
\* **Kiến trúc:**

Kiến trúc Process:

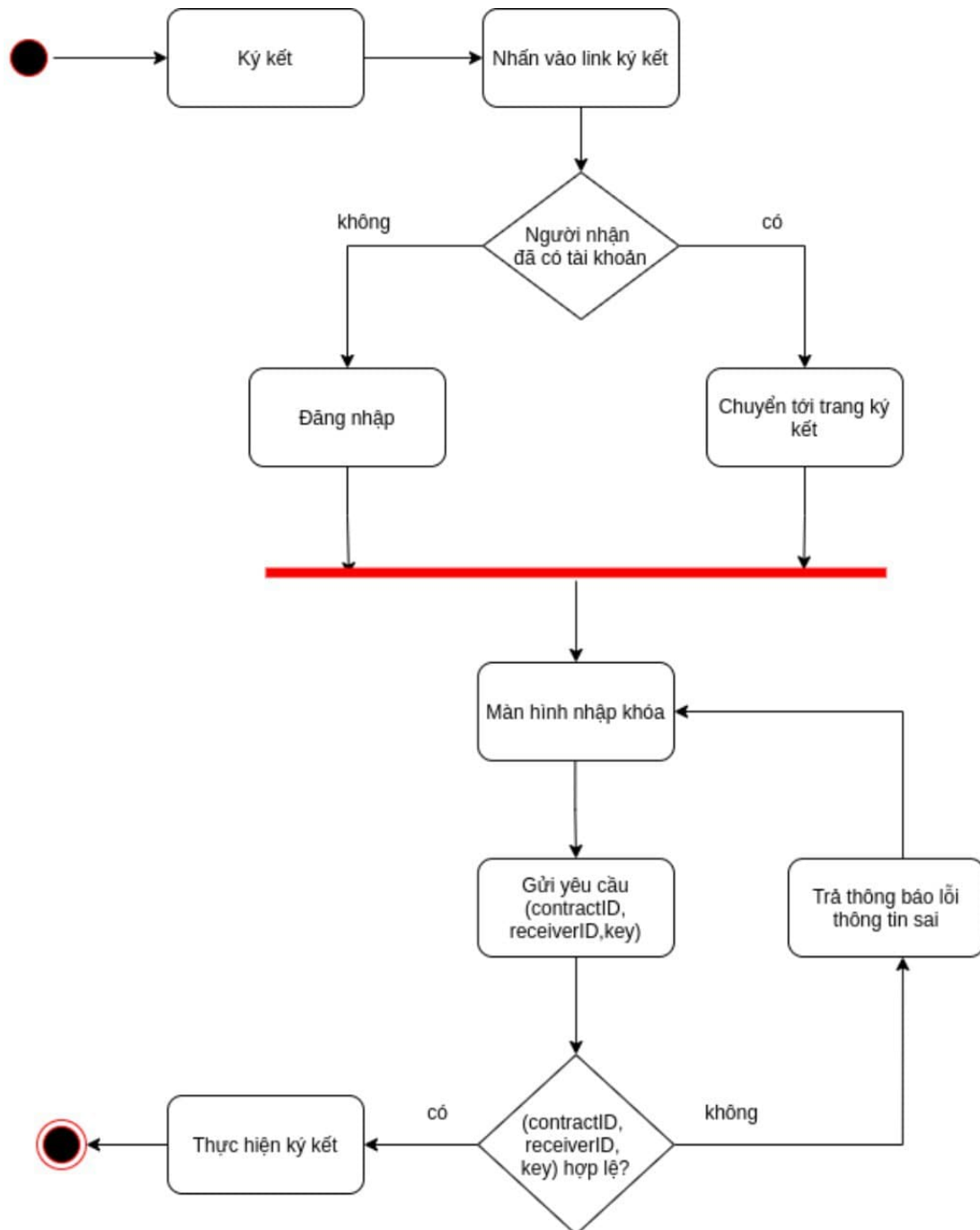


- Hình trên thể hiện quy trình khi thực hiện yêu cầu thì yêu cầu sẽ đi qua các service. Yêu cầu được gửi lên từ user (đã có tài khoản) thông qua phải có AccessToken. Sau đó Spring Cloud Gateway sẽ gửi AccessToken lên AuthService và lấy roles và permissions. Tạo ra Access Token mới và trả về cho gateway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

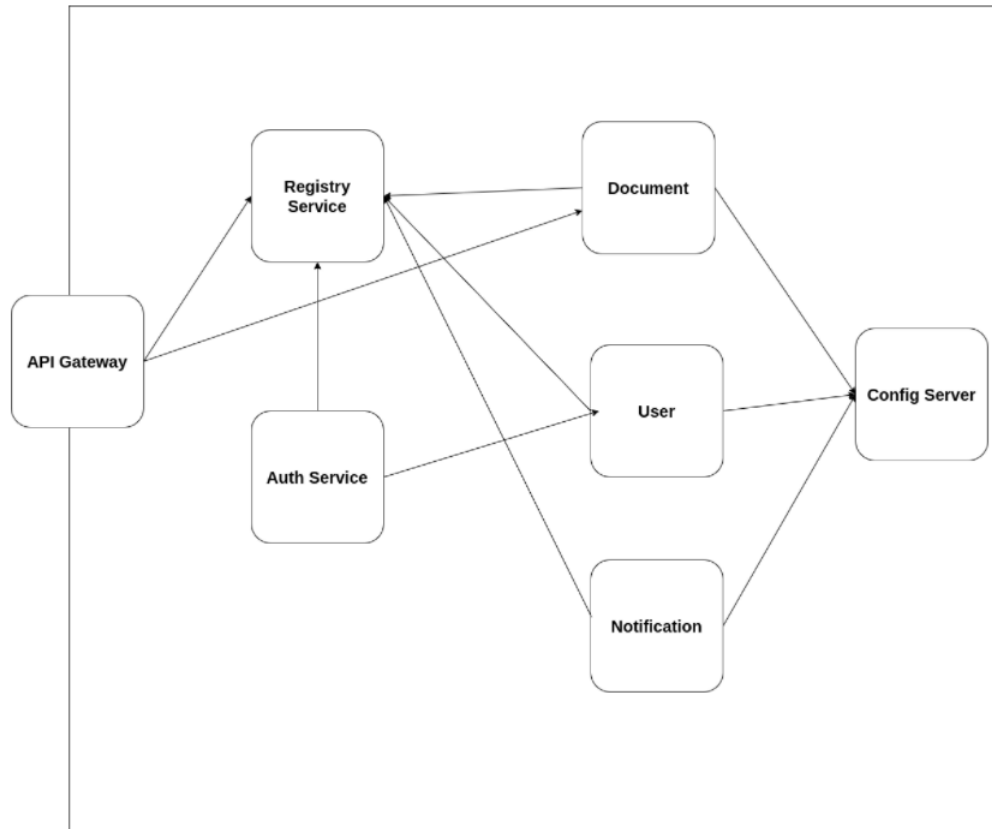
- Mô tả luồng gửi tài liệu:



- Mô tả luồng ký tài liệu:



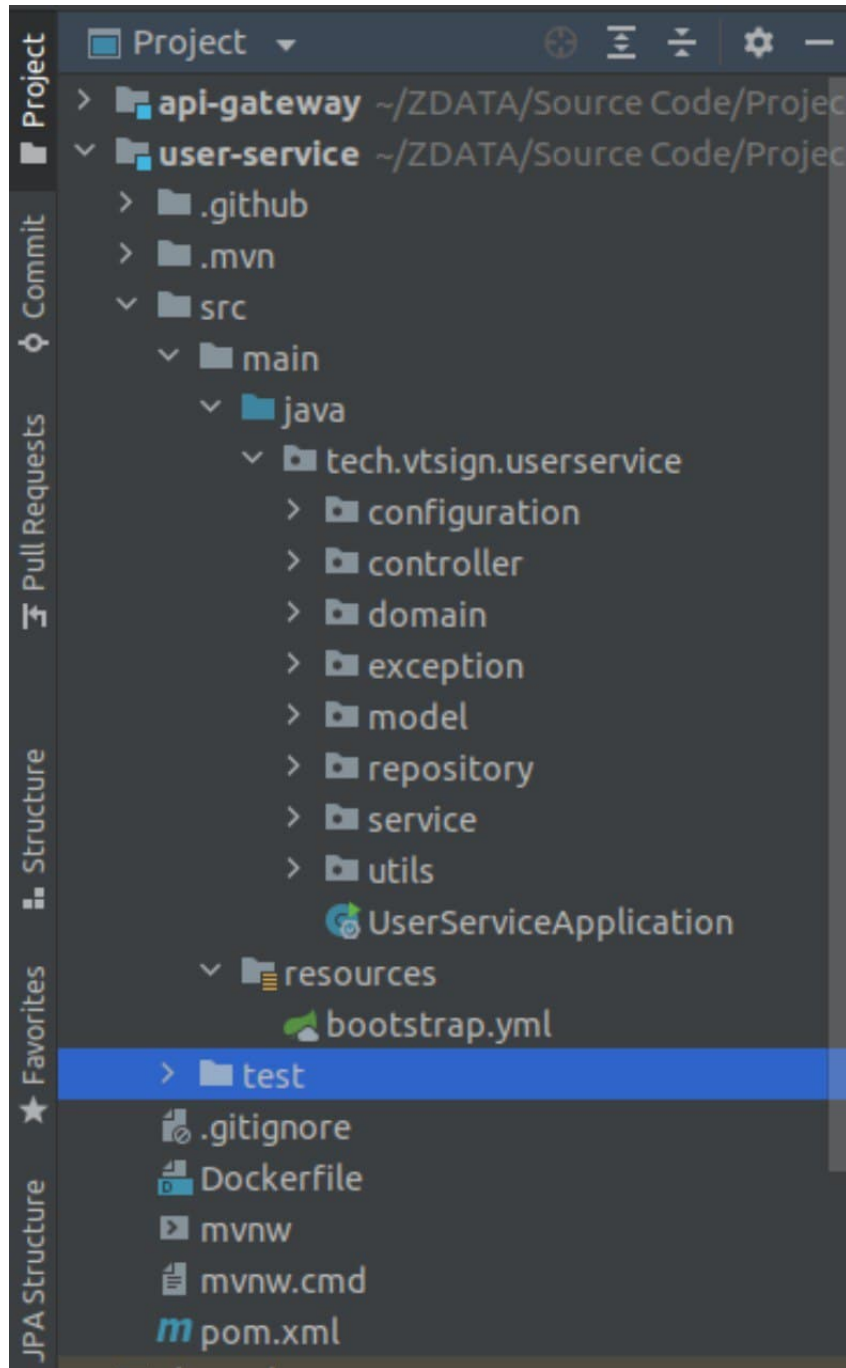
Kiến trúc Logical:



- 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.

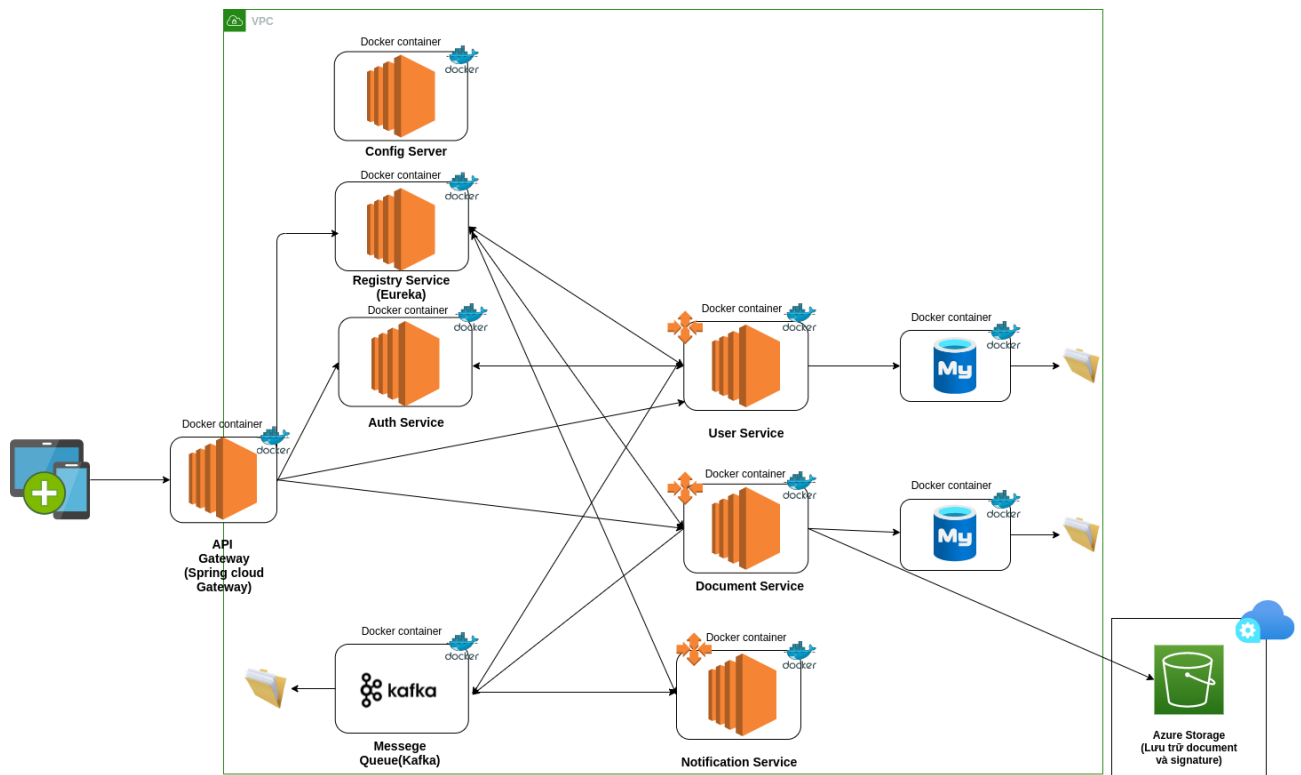


Kiến trúc Development:



- 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

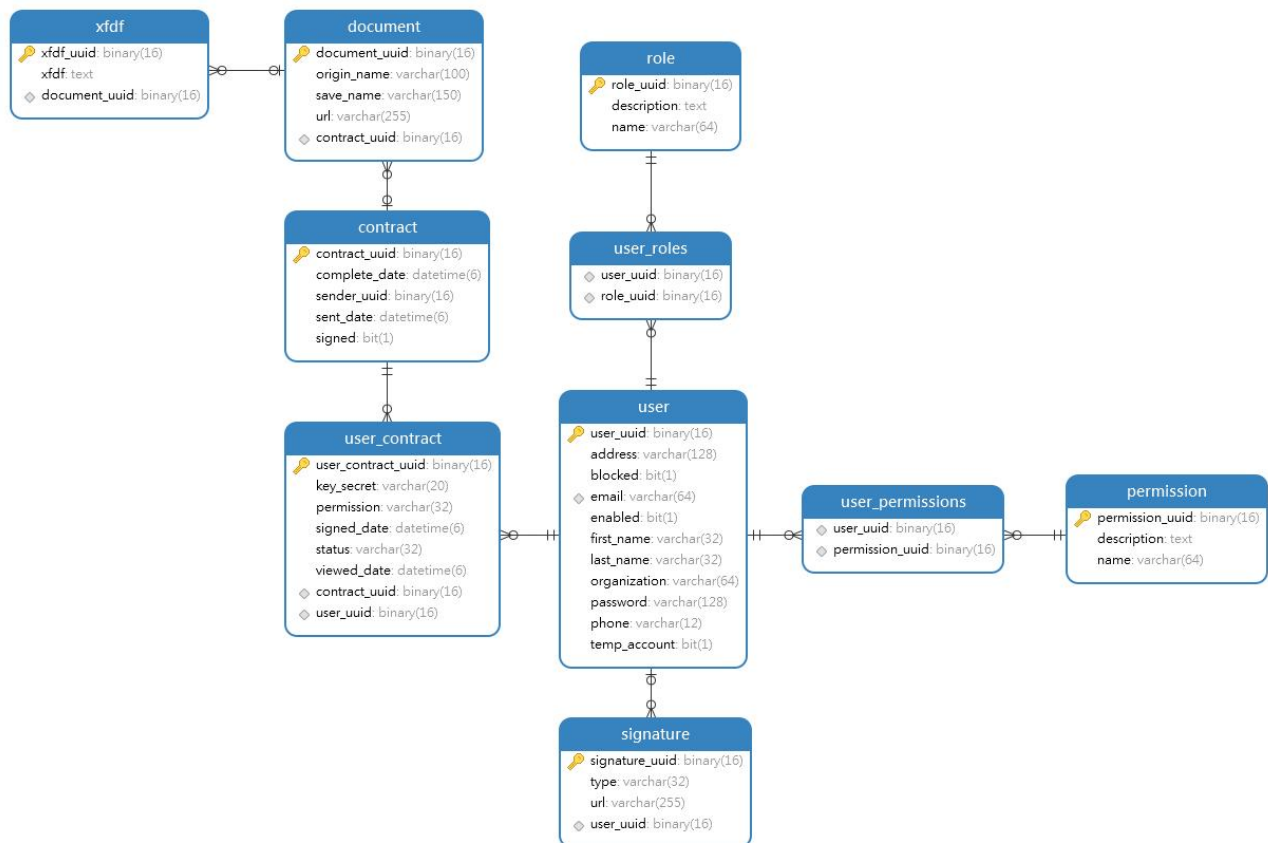
## - Kiến trúc Deployment



- 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 một cách 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<sup>1</sup> 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<sup>2</sup> - 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 khỏi đe dọa nâng cao	✓	✗	✓	✓
Quản lý tài liệu có thời hạn	✓	✗	✓	✓

---

<sup>1</sup><https://jmeter.apache.org/>

<sup>2</sup><https://nmap.org/>

**Bảng ưu điểm khuyết điểm của 4 hệ thống**

<b>Hệ thống</b>	<b>Ưu điểm</b>	<b>Khuyết điểm</b>
SignNow	<ul style="list-style-type: none"> <li>- Đăng nhập/dăng ký nhanh bằng tài khoản Facebook, Google.</li> <li>- Hệ thống sử dụng eSignature đơn giản, giao diện hiện đại dễ sử dụng, có LiveChat hỗ trợ khách hàng.</li> <li>- Bản dùng thử không giới hạn nhiều chức năng, bản quyền giá hợp lý, phù hợp nhiều đối tượng.</li> </ul>	<ul style="list-style-type: none"> <li>- Hỗ trợ ít ngôn ngữ, tập trung thị trường nước ngoài.</li> <li>- Giao diện trình duyệt chưa có chế độ nền tối.</li> <li>- Không dùng phương pháp Digital Signature, bảo mật còn hạn chế.</li> </ul>
DocuSign	<ul style="list-style-type: none"> <li>- Sử dụng phương pháp định danh Self-Signed Digital Signature, không đòi hỏi người dùng trả tiền mua Chứng thực (Certificate).</li> <li>- Sử dụng các công nghệ mã hóa và bảo mật dữ liệu, thông báo qua tin nhắn (SMS Delivery).</li> <li>- Phù hợp với các công ty quy mô lớn, đặt nặng về bảo mật chuyên môn.</li> <li>- Nhiều công cụ hỗ trợ kèm theo, đáp ứng nhiều tiêu chuẩn bảo mật</li> </ul>	<ul style="list-style-type: none"> <li>- Không có đăng nhập/dăng ký nhanh bằng tài khoản Facebook, Google.</li> <li>- Hệ thống nặng, truy cập chậm ở khu vực Việt Nam, không có LiveChat.</li> <li>- Không được chỉnh sửa tài liệu trước khi gửi và ký kết trực tiếp.</li> <li>- Hỗ trợ ít ngôn ngữ, tập trung thị trường Mỹ, bản quyền giá cao</li> </ul>

Hệ thống	Ưu điểm	Khuyết điểm
AdobeSign	<ul style="list-style-type: none"> <li>- Sử dụng phương pháp Digital Signature, đòi hỏi người dùng trả tiền mua 1 Chứng thực (Certificate).</li> <li>- Đồng bộ và cá nhân hóa với phần mềm liên quan của Adobe.</li> </ul>	<ul style="list-style-type: none"> <li>- Đăng nhập/đăng ký phức tạp, hệ thống khá chậm và tải nhiều thông tin và thông báo.</li> <li>- Giao diện cũ khó nhìn, nhiều tính năng thừa và khó sử dụng, Không ký trực tiếp.</li> <li>- Bản quyền và phí sử dụng cao, bản dùng thử nhiều ràng buộc chức năng.</li> </ul>
VTSign	<ul style="list-style-type: none"> <li>- Hệ thống xử lý và hiển thị nhanh, không hiện các thông báo gây nhiễu.</li> <li>- Bảo mật tài liệu, xác thực người dùng bằng email và số điện thoại cá nhân.</li> <li>- Giao diện thao tác đơn giản dễ sử dụng, phù hợp nhiều lứa tuổi.</li> <li>- Hỗ trợ ngôn ngữ tiếng việt, bản quyền sử dụng phù hợp người Việt Nam.</li> </ul>	<ul style="list-style-type: none"> <li>- Không có đăng nhập/đăng ký nhanh bằng tài khoản Facebook, Google.</li> <li>- Không chỉnh sửa tài liệu trước khi gửi, không LiveChat, không hỗ trợ nhiều ngôn ngữ.</li> <li>- Phù hợp với người dùng cá nhân, các tổ chức nhỏ, không chuyên môn cao.</li> </ul>

### 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 IDEA: một IDE Java để phát triển các phần mềm máy tính.
- Spring Framework: Hệ sinh thái Spring để triển khai web service với Java một cách dễ dàng.
- 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
- Digital Ocean: 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 đề án. Kinh nghiệm tích lũy đạt được khi thực hiện một đề 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"> <li>- Liên hệ giảng viên hướng dẫn xem xét, bàn luận để thống nhất nhận thực hiện đề tài.</li> <li>- Tìm hiểu thêm về đề tài. Nghiên cứu quy trình thực hiện đề án của giảng viên hướng dẫn.</li> <li>- Chuẩn bị và nộp đơn đăng ký thực tập dự án tốt nghiệp.</li> </ul>	Tất cả thành viên
01/09/2021 - 30/09/2021	<ul style="list-style-type: none"> <li>- Giai đoạn khởi tạo dự án, khảo sát thị trường với các hệ thống tương tự.</li> <li>- Chuẩn bị bản mẫu Prototype và Proof of Concept.</li> <li>- Khởi tạo và hoàn thành chương 1 báo cáo. Khởi tạo đề cương chi tiết, kế hoạch sơ bộ vai trò.</li> </ul>	Tất cả thành viên
01/10/2021 - 31/10/2021	<ul style="list-style-type: none"> <li>- Thiết kế luồng hoạt động dự kiến của hệ thống.</li> <li>- Tìm hiểu và lựa chọn các công cụ, công nghệ, thư viện hỗ trợ xây dựng hệ thống.</li> <li>- Cập nhật chương 2 báo cáo và đề cương.</li> </ul>	Tất cả thành viên

Thời gian	Công việc	Người thực hiện
01/11/2021 - 30/11/2021	<ul style="list-style-type: none"> <li>- Tổ chức mã nguồn, thiết kế giao diện trang chủ và đăng nhập.</li> <li>- Hoàn tất khái niệm chương 2, cập nhật chương 5 và đề cương. Gửi giảng viên góp ý để chỉnh sửa tài liệu.</li> <li>- Hoàn tất và nộp đề cương chi tiết cho khoa 10/11.</li> </ul>	Tất cả thành viên
01/12/2021 - 31/12/2021	<ul style="list-style-type: none"> <li>- Phát triển thiết kế kiến trúc hệ thống. Triển khai CI/CD. Xây dựng các chức năng đã đặt ra.</li> <li>- Thực hiện triển khai chức năng xử lý dữ liệu người dùng trên máy khách - chủ.</li> <li>- Hoàn tất cơ bản giao diện hệ thống. Triển khai phiên bản thử nghiệm.</li> <li>- Hoàn tất chương 5, cập nhật chương 4 và thảo luận với giảng viên.</li> </ul>	Tất cả thành viên
01/01/2022 - 31/01/2021	<ul style="list-style-type: none"> <li>- Tiếp tục xử lý các vấn đề còn lại của hệ thống, đánh giá chung và cải tiến.</li> <li>- Hoàn tất chương 4, cập nhật chương 3 và thảo luận với giảng viên.</li> </ul>	Tất cả thành viên
01/02/2022 - 28/02/2022	<ul style="list-style-type: none"> <li>- Thực hiện kiểm thử, triển khai phiên bản chính thức đầu tiên.</li> <li>- Hoàn tất chương 3, cập nhật làm xong chương 2, kiểm tra toàn bộ báo cáo.</li> <li>- Chuẩn bị và nộp đơn đăng ký bảo vệ đồ án 23/02.</li> </ul>	Tất cả thành viên

Thời gian	Công việc	Người thực hiện
01/03/2021 - Kết thúc	- 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 NGƯỜI HƯỚNG DẪN**  
*(Ký và ghi rõ họ tên)*

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