**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH**

**Đề tài 508: Xây dựng ứng dụng tự động đồng bộ dữ liệu trên nhiều máy trong mạng**

**SINH VIÊN THỰC HIỆN:**

**Đặng Văn Nhật Minh LỚP: 20TCLC-DT3 NHÓM: 20Nh13**

**Huỳnh Đinh Hoàng Viên LỚP: 20TCLC-DT3 NHÓM: 20Nh13**

**Trần Công Nguyên Hải LỚP: 20TCLC-DT3 NHÓM: 20Nh13**

**GIẢNG VIÊN HƯỚNG DẪN: ThS. Nguyễn Thế Xuân Ly**

**Đà Nẵng 12/ 2021**

MỤC LỤC

[MỤC LỤC 2](#_Toc123217613)

[DANH SÁCH HÌNH VẼ 3](#_Toc123217614)

[MỞ ĐẦU (GIỚI THIỆU ĐỀ TÀI) 4](#_Toc123217615)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 4](#_Toc123217616)

[1.1. Nêu vấn đề 4](#_Toc123217617)

[1.1.1. Đầu vào 4](#_Toc123217618)

[1.1.2. Đầu ra 4](#_Toc123217619)

[1.2. Các thành phần 4](#_Toc123217620)

[1.2.1. Mô hình Client-Server 4](#_Toc123217621)

[1.2.2. TCP/IP 5](#_Toc123217622)

[1.2.3. Mô hình MVC 7](#_Toc123217623)

[1.2.4. WebSocket 8](#_Toc123217624)

[1.2.5. Restful API 10](#_Toc123217625)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 11](#_Toc123217626)

[2.1. Các chức năng trong hệ thống 11](#_Toc123217627)

[2.2. Cơ sở dữ liệu 12](#_Toc123217628)

[2.3. Sơ đồ khối hệ thống 12](#_Toc123217629)

[2.3.1. Sơ đồ hoạt động 12](#_Toc123217630)

[2.3.2. Sơ đồ tuần tự 20](#_Toc123217631)

[CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 24](#_Toc123217632)

[3.1. Kết quả chương trình 24](#_Toc123217633)

[3.2. Đánh giá kết quả 29](#_Toc123217642)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 29](#_Toc123217643)

[TÀI LIỆU THAM KHẢO 31](#_Toc123217644)

DANH SÁCH HÌNH VẼ

[Hình 1. Mô hình Client – Server 5](#_Toc123219901)

[Hình 2. Giao thức TCP/IP 7](#_Toc123219902)

[Hình 3. So sánh HTTP và WebSocket về gửi nhần request/response 9](#_Toc123219903)

[Hình 4. Restful API 10](#_Toc123219904)

[Hình 5. Cơ sở dữ liệu 12](#_Toc123219905)

[Hình 6. Đăng nhập 13](file:///C:\Users\ASUS-FPT\Desktop\activity\PBL4-SyncFolder.docx#_Toc123219906)

[Hình 7. Đăng ký 14](#_Toc123219907)

[Hình 8. Xem thư mục user đang quản lý 15](#_Toc123219908)

[Hình 9. Xem thư mục user được chia sẻ từ user khác 16](#_Toc123219909)

[Hình 10. Download file/folder 16](#_Toc123219910)

[Hình 11. Xóa file/folder 17](#_Toc123219911)

[Hình 12. Tạo mới folder 18](#_Toc123219912)

[Hình 13. Cấp quyền file/folder cho user khác 19](#_Toc123219913)

[Hình 14. Tìm kiếm 20](#_Toc123219914)

[Hình 15. Tuần tự chức năng đăng nhập 21](#_Toc123219915)

[Hình 16. Tuần tự xem thư mục user quản lý 21](#_Toc123219916)

[Hình 17. Tuần tự chức năng tìm kiếm file/folder 22](#_Toc123219917)

[Hình 18. Tuần tự chức năng xóa file/folder 22](#_Toc123219918)

[Hình 19. Tuần tự chức năng cấp quyền 23](#_Toc123219919)

[Hình 20. Tuần tự chức năng tạo folder 23](#_Toc123219920)

[Hình 21. Giao diện đăng nhập 24](#_Toc123219921)

[Hình 22. Giao diện đăng ký 24](#_Toc123219922)

[Hình 23. Giao diện chính 25](#_Toc123219923)

[Hình 24. Giao diện chức năng upload file 25](#_Toc123219924)

[Hình 25. Giao diện hoàn thành upload file 26](#_Toc123219925)

[Hình 26. Giao diện cửa sổ cấp quyền 26](#_Toc123219926)

[Hình 27. Giao diện hoàn thành cấp quyền 27](#_Toc123219927)

[Hình 28. Giao diện xóa file 27](#_Toc123219928)

[Hình 29. Giao diện cửa sổ thêm file 28](#_Toc123219929)

[Hình 30. Giao diện hoàn thành thêm file 28](#_Toc123219930)

[Hình 31. Giao diện ô tìm kiếm 29](#_Toc123219931)

MỞ ĐẦU (GIỚI THIỆU ĐỀ TÀI)

Trong chương trình học phần Lập trình mạng và Mạng máy tính, các giao thức kết nối mạng là một trong những phần lý thuyết hết sức quan trọng. Trong chương trình này chúng em sử dụng giao diện lập trình web sử dụng WebSocket hoạt động theo mô hình TCP để truyền nhận dữ liệu giữa các client và server.

Mục tiêu của ứng dụng web chúng em là giúp nhóm người trong công ty cần chia sẽ dữ liệu với nhau và dữ liệu được đồng bộ với các thư mục chia sẻ trong mạng LAN. Người dùng có thể thực hiện những tác vụ cơ bản như thêm/sửa/xóa, cấp quyền chia sẻ file/thư mục mình quản lý.

Chúng em xin cảm ơn thầy Nguyễn Thế Xuân Ly đã hỗ trợ chúng em trong việc thực hiện đồ án lần này.

# CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

## Nêu vấn đề

Một nhóm người trong công ty cần chia sẻ dữ liệu các thư mục quản lý với nhau thông qua mạng cục bộ LAN mạng diện rộng WAN

### Đầu vào

* Người dùng cấp quyền truy cập vào thư mục dữ liệu chia sẻ trên các máy trong mạng (quyền có thể là đọc / ghi / sửa / xóa tương ứng với từng người dùng)
* Chương trình được cài đặt trên các máy

### Đầu ra

* Các tác tử của chương trình liên kết được với nhau và cho phép người dùng đăng nhập với tài khoản và quyền tưng ứng.
* Cho phép người dùng thao tác với quyền của mình trên file được đồng bộ.

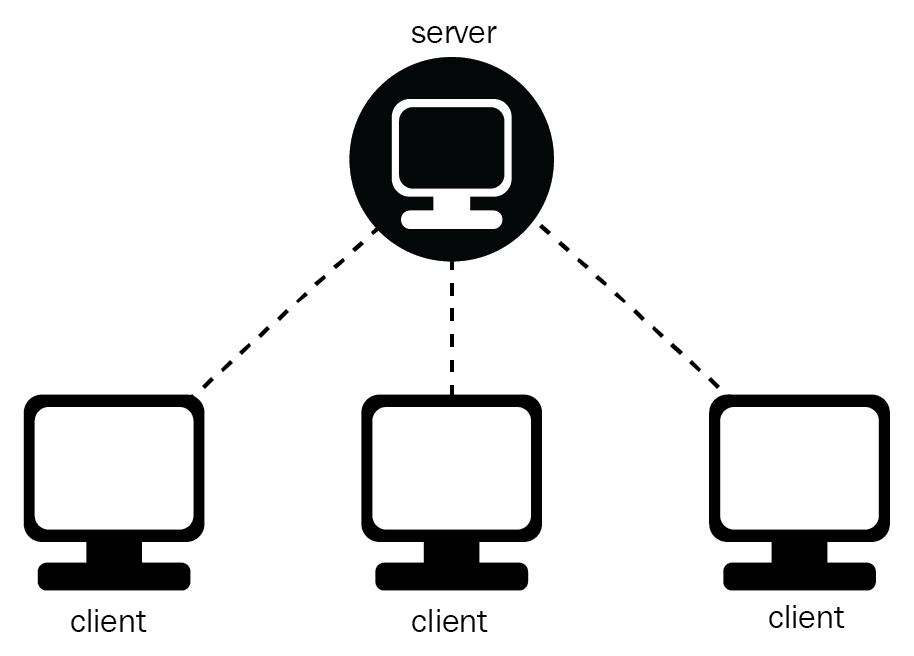
## Các thành phần

### Mô hình Client-Server

#### Định nghĩa

Client-Server là mô hình mạng máy tính bao gồm 2 thành phần chính là máy khách (client) và máy chủ (server). Trong mô hình này, server là nơi lưu trữ tài nguyên, cài đặt các chương trình dịch vụ và thực hiện các yêu cầu của client. Client đón vai trò gửi yêu cầu đến server. Client gồm máy tính và thiết bị điện tử nói chung. Với sự phát triển vô cùng mạnh mẽ của mạng lưới internet ngày nay, mô hình client-server được áp dụng rất rộng rãi và phổ biến nhất.

Chương trình server và client nói chuyện với nhau bằng các thông điệp (messages) thông qua một cổng truyền thông liên tác IPC (Interprocess Communication). Để một chương trình server và một chương trình client có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn để nói chuyện, chuẩn này được gọi là giao thức. Nếu một chương trình client nào đó muốn yêu cầu lấy thông tin từ server thì nó phải tuân theo giao thức mà server đó đưa ra.



Hình 1. Mô hình Client – Server

#### Ưu điểm

* Có khả năng chống quá tải mạng
* Đảm bảo toàn vẹn dữ liệu khi có sự cố xảy ra trên môi trường internet
* Dễ dàng mở rộng hệ thống mạng
* Chỉ cần chung định dạng giao tiếp mà không cần chung nền tảng là có thể hoạt động được
* Với mô hình Client-Server, người dùng có thể truy cập dữ liệu từ xa, thực hiện các thao tác gửi, nhận file hay tìm kiếm thông tin đơn giản

#### Nhược điểm

* Cần bảo trì, bảo dưỡng server thường xuyên.
* Khả năng bảo mật thông tin mạng là một hạn chế nữa của Client-Server. Bởi vì, nguyên lý hoạt động của Client-Server là trao đổi dữ liệu giữa server và client ở 2 khu vực địa lý khác nhau. Trong quá trình trao đổi dữ liệu, khả năng thông tin mạng bị lộ là điều dễ xảy ra.

### TCP/IP

#### Định nghĩa

**TCP/IP** là tên viết tắt của cụm từ **Transmission Control Protocol/Internet Protocol**, là một tập hợp các giao thức (protocol) trao đổi thông tin được sử dụng để truyền tải và kết nối các thiết bị trong mạng Internet. Cụ thể hơn, TCP/IP chỉ rõ cho chúng ta cách thức đóng gói thông tin (còn được gọi là gói tin ), được gửi và nhận bởi các máy tính có kết nối với nhau. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là [TCP](https://vi.wikipedia.org/wiki/TCP) (giao thức điều khiển giao vận) và [IP](https://vi.wikipedia.org/wiki/IP) (giao thức liên mạng). Trong đó hai giao thức chính trong TCP/IP phục vụ các chức năng cụ thể:

* TCP xác định cách các ứng dụng có thể tạo ra các kênh truyền thông qua mạng. Nó cũng quản lý cách một thông điệp được tập hợp thành các gói nhỏ hơn trước khi chúng được truyền qua internet và được tập hợp lại theo đúng thứ tự tại địa chỉ đích.
* IP xác định cách định địa chỉ và định tuyến từng gói để đảm bảo rằng nó đến đúng đích. Mỗi máy tính cổng trên mạng sẽ kiểm tra địa chỉ IP này để xác định nơi chuyển tiếp thư.

Bộ giao thức này cung cấp giao tiếp dữ liệu đầu cuối, xác định cách dữ liệu được đóng gói (packetized), đánh địa chỉ (addressed), truyền (transmitted), routed (định tuyến) và nhận (received).

Chức năng này được tổ chức thành 4 tầng trừu tượng.

- Từ thấp nhất đến cao nhất, bắt đầu từ tầng liên kết, chứa các phương thức truyền thông cho dữ liệu nằm trong một phân đoạn mạng (network segment)

- Tầng internet, cung cấp kết nối giữa các mạng độc lập

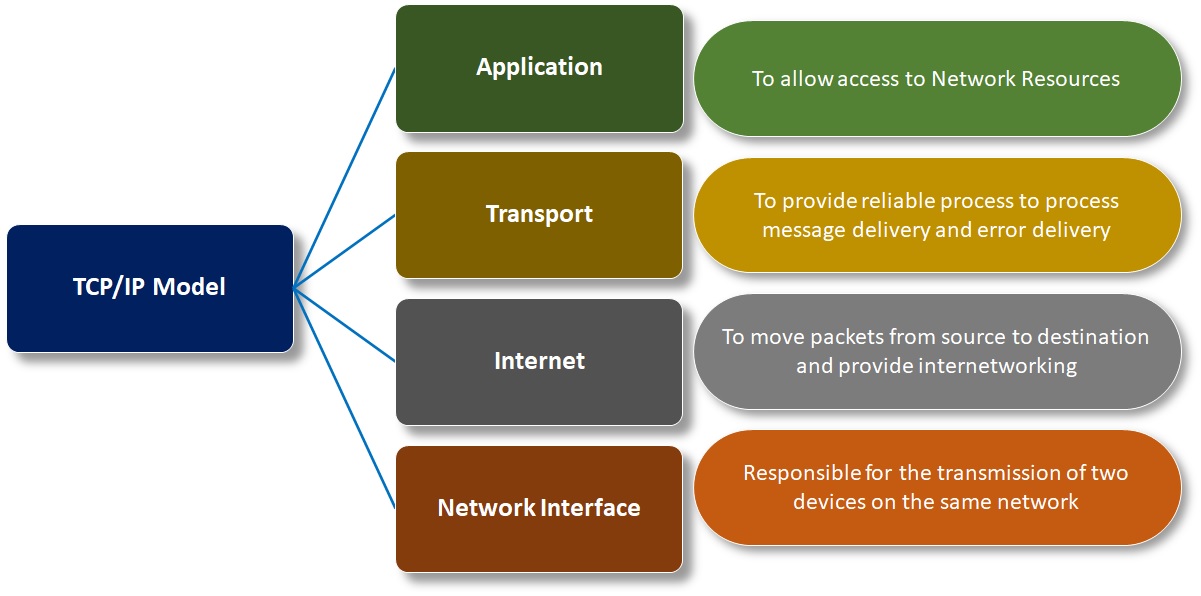
- Tầng giao vận, xử lý giao tiếp giữa các máy chủ (host)

- Tầng ứng dụng, cung cấp khả năng trao đổi dữ liệu giữa tiến trình và tiến trình cho các ứng dụng.

#### Cấu trúc mô hình TCP/IP

Mô hình TCP/IP tiêu chuẩn bao gồm 4 tầng được chồng lên nhau là:

* Tầng 1 - Tầng vật lý: Là sự kết hợp giữa tầng Vật lý và tầng liên kết dữ liệu của mô hình OSI. Chịu trách nhiệm truyền dữ liệu giữa hai thiết bị trong cùng một mạng. Tại đây, các gói dữ liệu được đóng vào khung (gọi là Frame) và được định tuyến đi đến đích đã được chỉ định ban đầu.
* Tầng 2 - Tầng mạng: chịu trách nhiệm truyền tải dữ liệu một cách logic trong mạng. Các phân đoạn dữ liệu sẽ được đóng gói (Packets) với kích thước mỗi gói phù hợp với mạng chuyển mạch mà nó dùng để truyền dữ liệu. Lúc này, các gói tin được chèn thêm phần Header chứa thông tin của tầng mạng và tiếp tục được chuyển đến tầng tiếp theo. Các giao thức chính trong tầng là IP, [ICMP](https://www.totolink.vn/article/73-icmp-la-gi-9-loai-icmp-thuong-thay.html) và ARP.
* Tầng 3 - Tầng giao vận: xử lý vấn đề giao tiếp giữa các máy chủ trong cùng một mạng hoặc khác mạng được kết nối với nhau thông qua bộ định tuyến. Trong tầng này còn bao gồm 2 giao thức cốt lõi là TCP và UDP. Trong đó, TCP đảm bảo chất lượng gói tin nhưng tiêu tốn thời gian khá lâu để kiểm tra đầy đủ thông tin từ thứ tự dữ liệu cho đến việc kiểm soát vấn đề tắc nghẽn lưu lượng dữ liệu. Trái với điều đó, UDP cho thấy tốc độ truyền tải nhanh hơn nhưng lại không đảm bảo được chất lượng dữ liệu được gửi đi.
* Tầng 4: Tầng ứng dụng: đảm nhận vai trò giao tiếp dữ liệu giữa 2 máy khác nhau thông qua các dịch vụ mạng khác nhau (duyệt web, chat, gửi email, một số giao thức trao đổi dữ liệu: SMTP, SSH, [FTP](https://www.totolink.vn/article/154-giao-thuc-ftp-la-gi-nhung-dieu-ban-chua-biet-ve-giao-thuc-ftp.html),...). Dữ liệu khi đến đây sẽ được định dạng theo kiểu Byte nối Byte, cùng với đó là các thông tin định tuyến giúp xác định đường đi đúng của một gói tin.



Hình 2. Giao thức TCP/IP

#### Ưu điểm của mô hình TCP/IP

* TCP có khả năng tương thích cao với tất cả các hệ điều hành
* Khả năng mở rộng cao, nó có thể định tuyến, xác định đường dẫn hiệu quả nhất thông qua mạng
* Nhẹ và không gây quá tải không cần thiết cho mạng hoặc máy tính.
* Tính năng phục hồi dữ liệu bị mất trên đường truyền.

### Mô hình MVC

#### Định nghĩa

MVC là viết tắt của cụm từ “Model – View – Controller” . Đây là mô hình thiết kế được sử dụng rộng rãi trong hầu hết các ngôn ngữ lập trình hướng đối tượng: Java, PHP,… thường được dùng để phát triển giao diện người dùng, cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

**MVC** có mục tiêu nhằm chia tách phần giao diện và phần code của ứng dụng để dễ dàng quản lý, bảo trì và phát triển. Bao gồm 3 thành phần chính có tương tác với nhau:

* **Model (dữ liệu)**: Tách riêng thành phần logic và phần hiển thị
* **Model.Bean**:Chứa các thực thể, gồm các thuộc tính (private), constructor, kèm các phương thức get/set
* **Model.DAO**: Thực hiện các công việc liên quan đến tương tác trực tiếp với Database như kết nối, lấy dữ liệu, truy vấn, chỉnh sửa…
* **Model.BO (service)**: Truyền yêu cầu từ Controller chuyển đến Model.DAO để xử lý, nhận dữ liệu từ Model.DAO và trả về cho Controller. Ngoài ra còn xử lý nghiệp vụ.
* **View**: là giao diện của hệ thống tương tác trực tiếp với người dùng.
* **Controller**: nhận yêu cầu từ người dùng, đưa yêu cầu và nhận dữ liệu từ tầng Model, từ đó chuyển hướng trả về cho View.

#### Ưu điểm của mô hình MVC

* Băng thông nhẹ vị không sử dụng viewstate nên khá tiết kiệm băng thông, giúp website hoạt động ổn định hơn.
* Tách biệt các phần Model, Controller, View với nhau, giúp việc kiểm tra đơn giản hơn, phát hiện các lỗi trước khi ra mắt cho người dùng.
* Chức năng Controller có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau.
* Bảo trì, tái sử dụng code dễ dàng.
* Phân chia công việc cho các developer dễ dàng hơn.

#### Nhược điểm

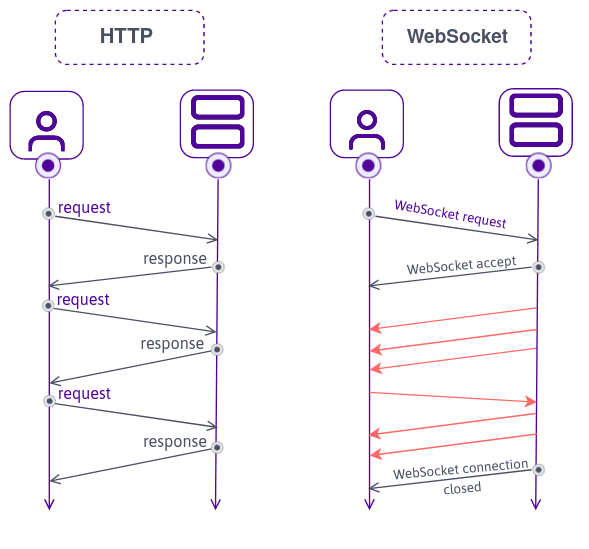
* Ít phù hợp với các dự án nhỏ vì khá là cồng kềnh và mất thời gian.
* Khó triển khai rộng rãi

### WebSocket

#### Định nghĩa

Socket là một endpoint (điểm kết thúc) của liên kết giao tiếp hai chiều giữa hai chương trình chạy trên internet, cho phép quy trình này “giao tiếp” với 1 quy trình khác. Khi muốn kết nối giữa một máy tính và máy chủ dữ liệu bắt buộc phải đi qua các lớp socket sau đó được các tầng TCP định danh.

**WebSocket** là một phương thức hỗ trợ giao tiếp 2 chiều giữa client (một chương trình hoặc một thiết bị) và server (máy chủ) bằng cách sử dụng một tầng TCP socket. Khi chưa có WebSocket client và máy chủ giao tiếp với nhau chỉ 1 chiều qua giao thức HTTP đồng thời data trả ề chứa rất nhiều dữ liệu header nên rất chậm. WebSocket giúp client và server (máy chủ) có thể cùng gửi yêu cầu và trả về dữ liệu cùng lúc song song với nhau.



Hình 3. So sánh HTTP và WebSocket về gửi nhần request/response

#### Ưu điểm của WebSocket

* Cho phép client và server giao tiếp realtime 2 chiều.
* Giảm độ trễ network cho những ứng dụng cần tính năng realtime vì header request của WebSocket có kích thước nhỏ hơn nhiều so với HTTP.
* Truyền và nhận dữ liệu trực tiếp không thông qua bất kỳ tầng bổ sung nào.

#### Nhược điểm

* WebSocket chưa được hỗ trợ trên tất cả các trình duyệt web.
* Khó sử dụng với các dịch vụ giới hạn phạm vi yêu cầu.

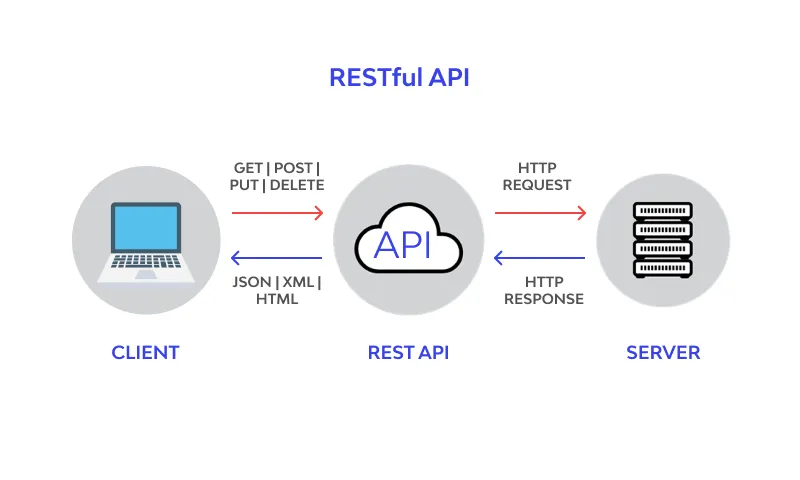
### Restful API

#### Định nghĩa

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

Các thành phần của Restful API:

* **API (Application Programming Interface)**: là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như [JSON](https://topdev.vn/blog/json-la-gi/) hay XML.
* **REST** **(Representational State Transfer)**: là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE,… đến một URL để xử lý dữ liệu.



Hình 4. Restful API

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.

#### Ưu điểm

* Dễ hiểu, đơn giản
* Cho phép tổ chức các ứng dụng phức tạp dễ dàng sử dụng tài nguyên
* Quản lý tải cao nhờ HTTP Proxy server và cache
* Các client mới có thể dễ dàng làm việc trên nhưng ứng dụng khác
* Cho phép sử dụng các lệnh gọi thủ tục HTTP tiêu chuẩn để truy xuất dữ liệu và request
* RESTful API dựa trên code và có thể sử dụng nó để đồng bộ hóa dữ liệu bằng website
* Cung cấp các định dạng linh hoạt bằng các tuần tự hóa (serialize) dữ liệu ở dạng JSON hay XML

#### Nhược điểm

* Không có trạng thái: hầu hết các ứng dụng web đều yêu cầu cơ chế stateful (có trạng thái)
* Bảo mật: REST có thể phù hợp với các URL public, nhưng không phải là một lựa chọn tốt nếu cần truyền dữ liệu nhạy cảm giữa client và server

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Các chức năng trong hệ thống

Ứng dụng bao gồm các chức năng :

* Đăng nhập
* Đăng ký
* Xem thư mục của user quản lý
* Xem thư mục của user được chia sẻ từ user khác
* Download file/folder
* Xóa file/folder
* Tạo mới folder
* Cấp quyền file/folder cho user khác
* Tìm kiếm file

## Cơ sở dữ liệu

Diagram

Description automatically generated

Hình 5. Cơ sở dữ liệu

## Sơ đồ khối hệ thống

### Sơ đồ hoạt động

#### Đăng nhập

Diagram

Description automatically generated

Hình 6. Đăng nhập

#### Đăng ký

Diagram, schematic

Description automatically generated

Hình 7. Đăng ký

#### Xem thư mục user quản lý

Diagram

Description automatically generated

Hình 8. Xem thư mục user đang quản lý

#### Xem thư mục được chia sẻ từ user khác

Diagram

Description automatically generated

Hình 9. Xem thư mục user được chia sẻ từ user khác

#### Download file/folder

Diagram, schematic

Description automatically generated

Hình 10. Download file/folder

#### Xóa file/folder

Diagram, schematic

Description automatically generated

Hình 11. Xóa file/folder

#### Tạo folder

Diagram

Description automatically generated

Hình 12. Tạo mới folder

#### Cấp quyền file/folder cho user khác

Diagram

Description automatically generated

Hình 13. Cấp quyền file/folder cho user khác

#### Tìm kiếm

Diagram

Description automatically generated

Hình 14. Tìm kiếm

### Sơ đồ tuần tự

#### Đăng nhập

Diagram, timeline

Description automatically generated

Hình 15. Tuần tự chức năng đăng nhập

#### Xem thư mục user quản lý

Diagram

Description automatically generated with medium confidence

Hình 16. Tuần tự xem thư mục user quản lý

#### Tìm kiếm file/folder

Diagram

Description automatically generated

Hình 17. Tuần tự chức năng tìm kiếm file/folder

#### Xóa file/folder

Diagram, timeline

Description automatically generated

Hình 18. Tuần tự chức năng xóa file/folder

#### Cấp quyền

Diagram, schematic

Description automatically generated

Hình 19. Tuần tự chức năng cấp quyền

#### Tạo folder

Diagram

Description automatically generated

Hình 20. Tuần tự chức năng tạo folder

# CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

## Kết quả chương trình

### Đăng nhập

Graphical user interface

Description automatically generated

Hình 21. Giao diện đăng nhập

### Đăng ký

Graphical user interface, application

Description automatically generated

Hình 22. Giao diện đăng ký

### Giao diện chính (chức năng xem thư mục)

Graphical user interface, text, application, email

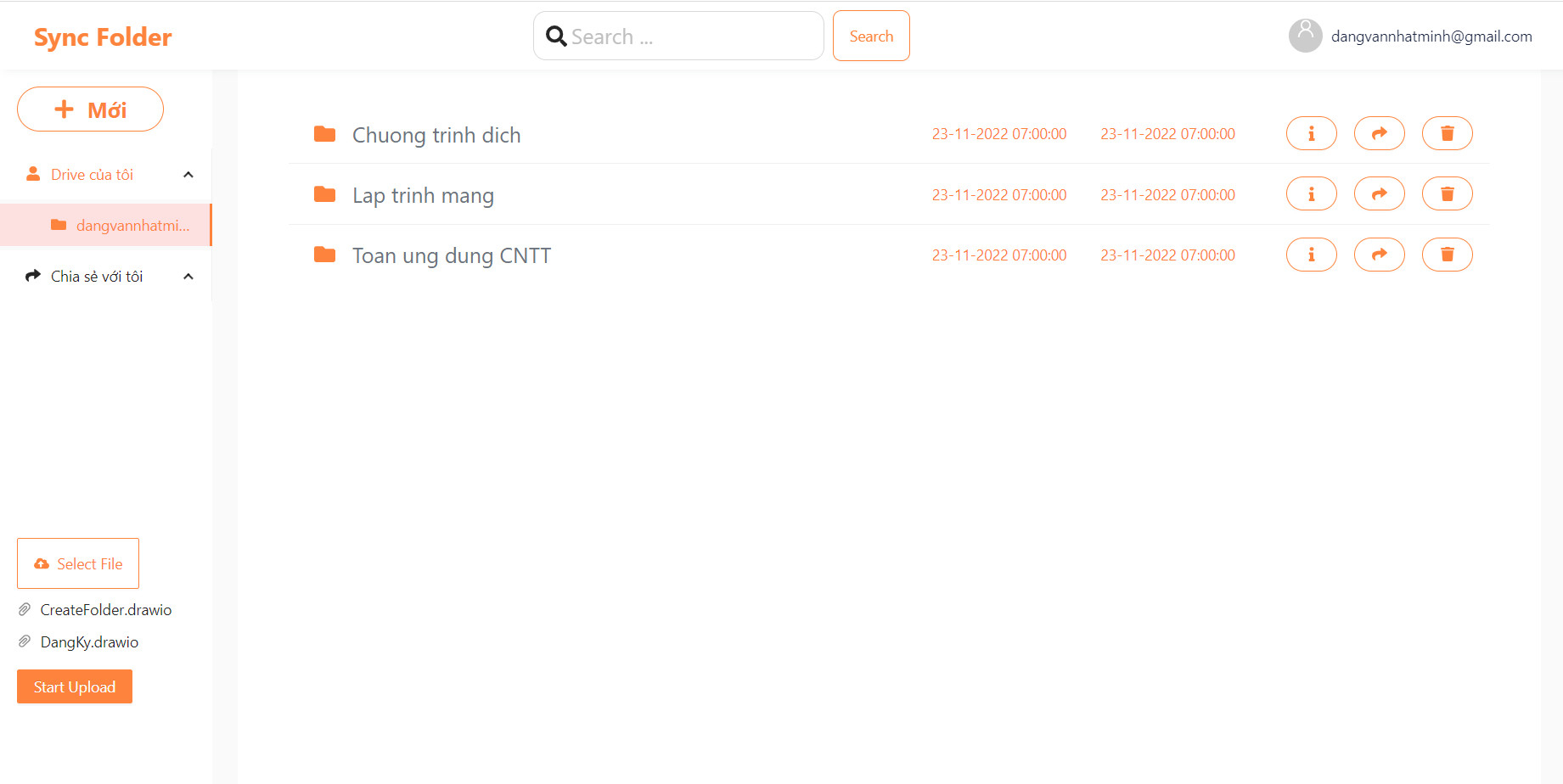
Description automatically generated

Hình 23. Giao diện chính

### Upload file

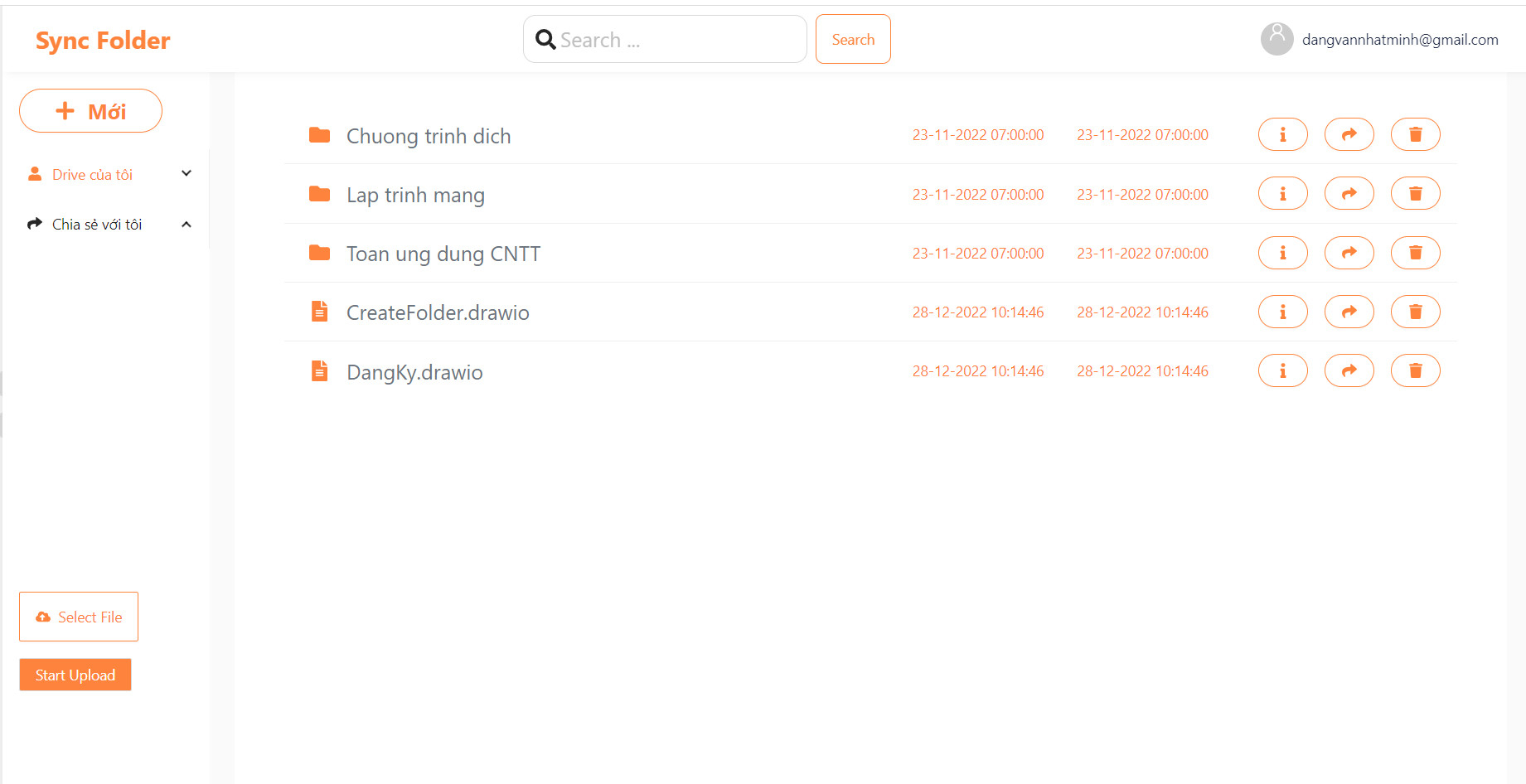
* Ban đầu





Hình 24. Giao diện chức năng upload file

* Sau khi upload 2 file (.drawio)



Hình 25. Giao diện hoàn thành upload file

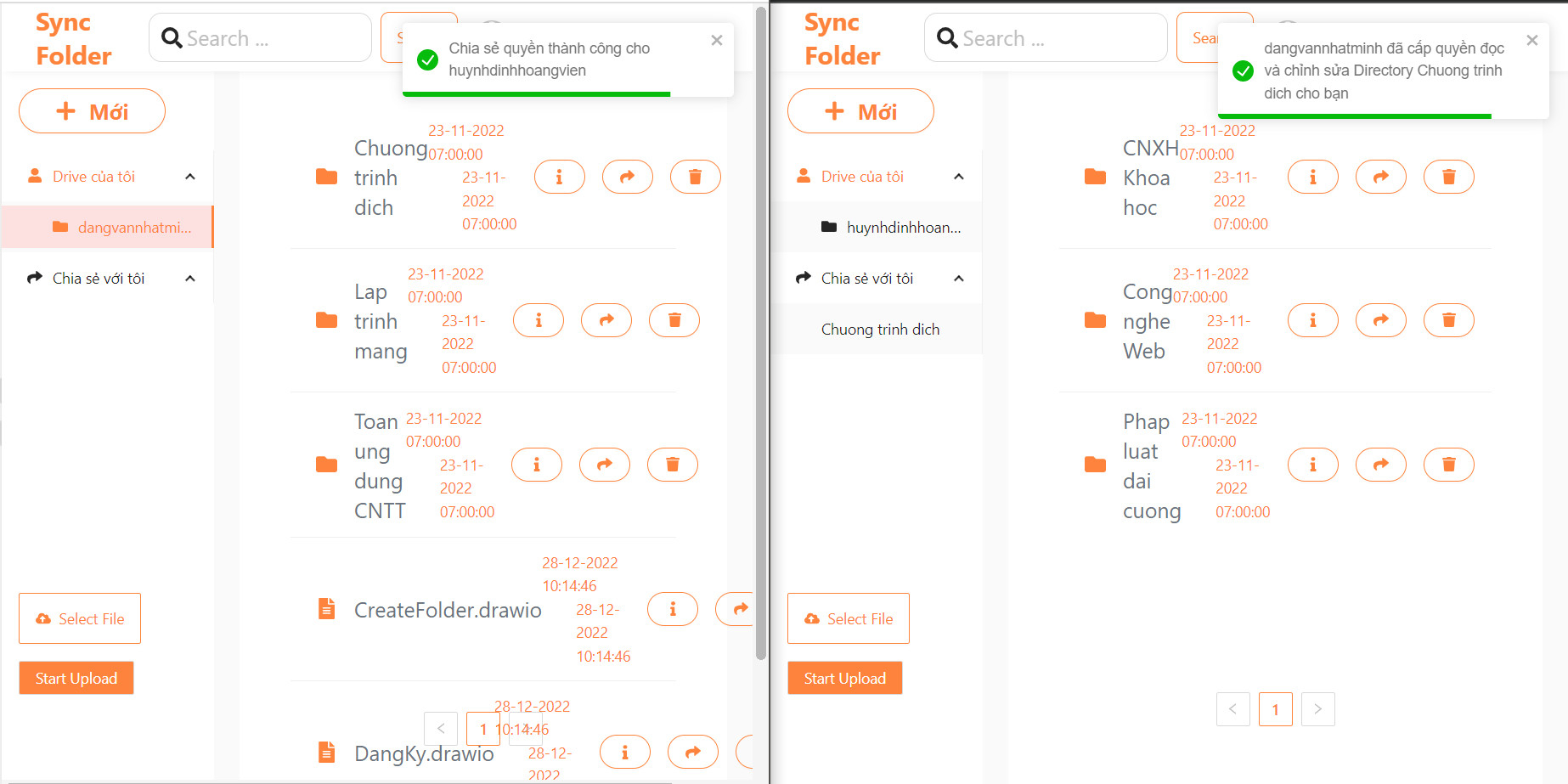
### Cấp quyền

Graphical user interface, text, application

Description automatically generated

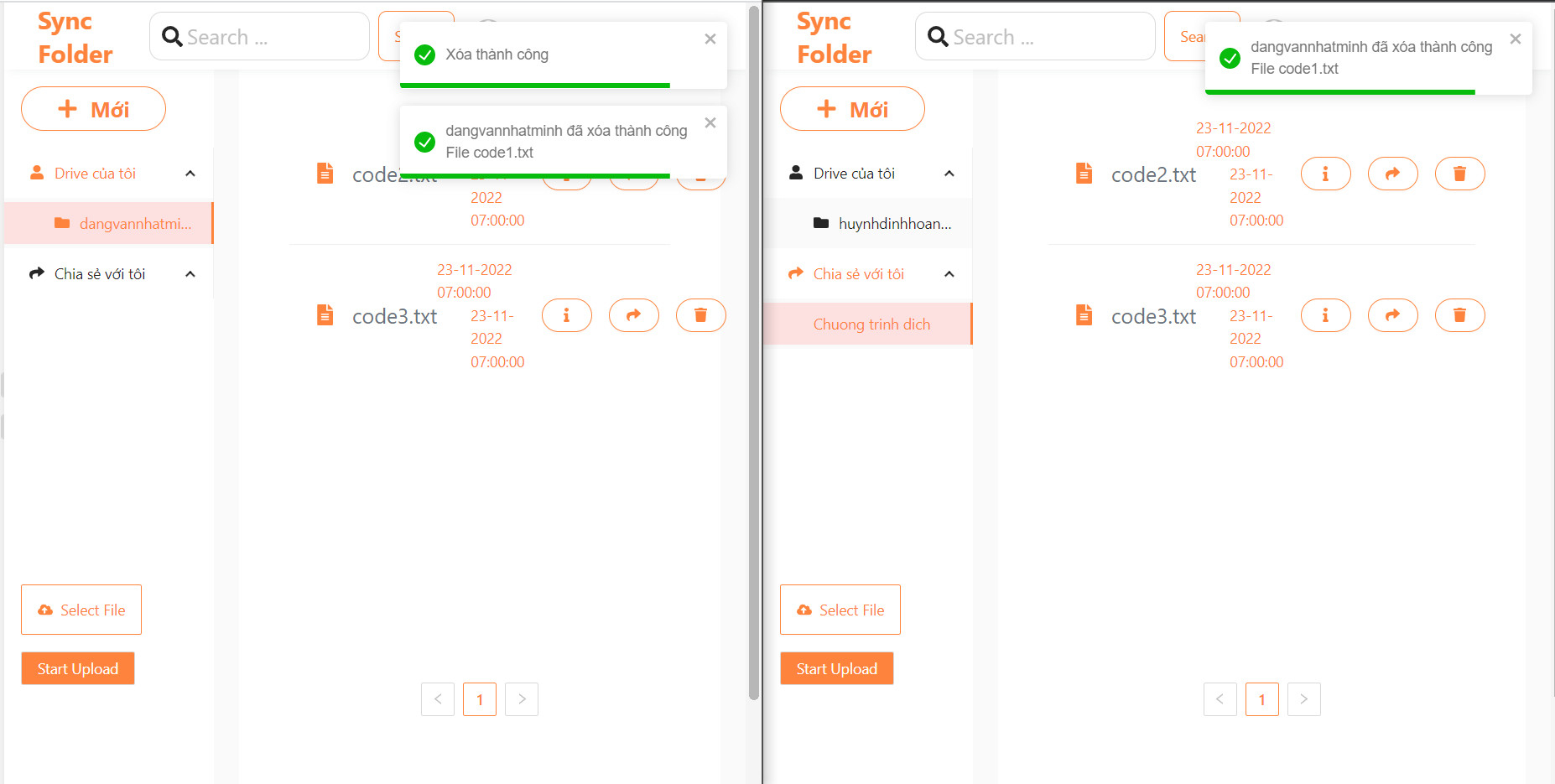
Hình 26. Giao diện cửa sổ cấp quyền

* Cấp quyền folder “Chương trình dịch”



Hình 27. Giao diện hoàn thành cấp quyền

### Xóa file



Hình 28. Giao diện xóa file

### Thêm file

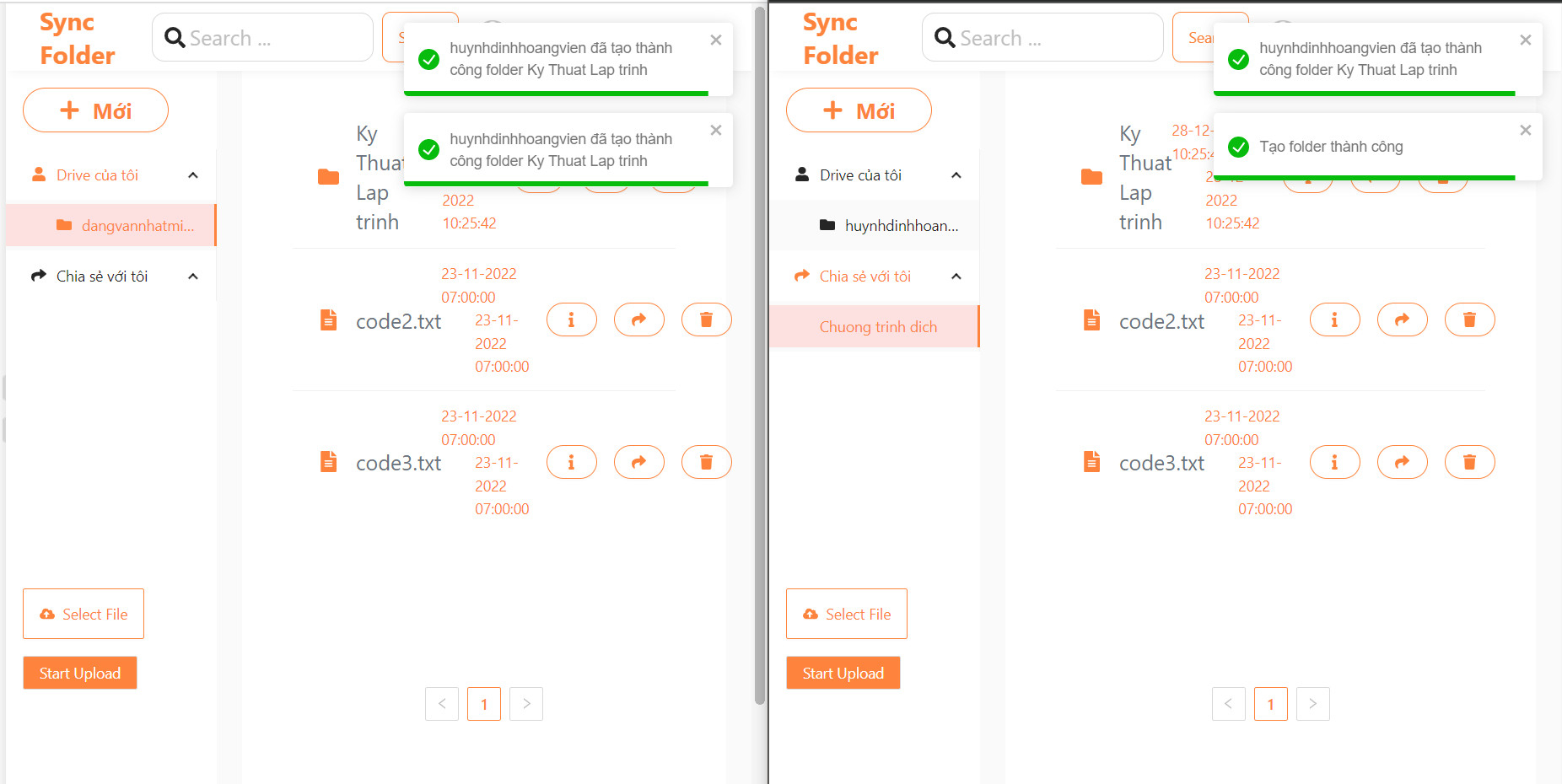
Graphical user interface, text, application

Description automatically generated

Đặt tên folder

Hình 29. Giao diện cửa sổ thêm file

* Đặt tên folder “Kỹ thuật lập trình”



Hình 30. Giao diện hoàn thành thêm file

### Tìm kiếm

Graphical user interface, text, application, email

Description automatically generated

Hình 31. Giao diện ô tìm kiếm

## Đánh giá kết quả

Chương trình hoạt động khá ổn định, giao diện dễ sử dụng. Việc áp dụng WebSocket giúp việc truyền nhận dữ liệu giữa Server và Client nhanh và bảo mật, độ trễ gần như bằng không giúp tăng trải nghiệm của người dùng. Sử dụng đa luồng giúp Server có thể xử lý độc lập nhiều Client cùng một thời điểm.

Tuy nhiên ứng dụng còn tiềm tàng những lỗi nhỏ chưa được tìm thấy.

WebSocket còn chưa tích hợp với nhiều trình duyệt web. Đồng thời WebSocket chưa thật sự tương thích với các framework như hibernate. Hibernate là một framework cung cấp một bộ lọc xung quanh một HTTP request. Khi bắt đầu một request, nó sẽ thiết lập một contest (chứa các transaction và liên kết JDBC) được ràng buộc với luồng request. Khi request đó kết thúc, bộ lọc hủy bỏ contest này.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Qua đồ án hệ điều hành và lập trình mạng kỳ này, chúng em học được cách xây dựng một ứng dụng website dựa trên mô hình MVC và học được cách viết Web service và WebSocket giúp đồng bộ file giữa các user với nhau. Hiểu hơn về bản chất của chồng giao thức TCP/IP. Đồ án còn giúp chúng em thực hành các môn đã học như Lập trình mạng, Công nghệ Web, Mạng máy tính.

Ngoài ra thông qua đồ án còn giúp chúng em nâng cao khả năng làm việc nhóm và kĩ năng sử dụng các công cụ hỗ trợ như Git, Postman, …

* Tiếp thu các ý kiến đóng góp để hoàn thiện nội dung đồ án.
* Tiếp tục nghiên cứu và phát triển thuật toán, từ đó tối ưu hóa thời gian chạy của hệ thống.
* Phát triển thêm các chức năng như chỉnh sửa thông tin người dùng.
* Phát triển thêm module Admin để quản lý website.
* Có thể phát triển ứng dụng bằng cách deploy lên các Cloud Application Platform như Heroku, đồng thời chuyển nơi lưu các file từ server sang Google Drive API vì Heroku chỉ quản lý file tĩnh.

TÀI LIỆU THAM KHẢO

[1] Slide môn học công nghệ web – Mai Văn Hà

[2] Slide môn học lập trình mạng – Phạm Minh Tuấn

[3] <https://www.baeldung.com/java-websockets>

[4] <https://socket.io/>