

ĐẠI HỌC BÁCH KHOA TP HCM  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



---

Báo cáo Bài tập lớn Môn Mạng máy tính

---

**NETWORK APPLICATION**

---

**Giảng viên:** Hoàng Lê Hải Thanh  
**Lớp:** L04

<b>Sinh viên:</b> Đỗ Ngọc Ánh	2110029
Nguyễn Đức Anh Tuấn	2115177
Nguyễn Tuấn Cường	2110029
Lê Minh Khôi	2233112

Ho Chi Minh City, 11/2023



## CONTENTS

<b>1</b>	<b>Phân công</b>	<b>2</b>
1.1	Danh sách thành viên và bảng đánh giá . . . . .	2
<b>2</b>	<b>Phase 1</b>	<b>3</b>
2.1	Phân tích yêu cầu . . . . .	3
2.1.1	Yêu cầu chức năng . . . . .	3
2.1.1.1	Client . . . . .	3
2.1.1.2	Server . . . . .	3
2.1.2	Yêu cầu phi chức năng . . . . .	4
2.2	Giao thức . . . . .	4
2.2.1	Giao thức tạo và quản lý tài khoản . . . . .	4
2.2.2	Giao thức trao đổi File . . . . .	5
<b>3</b>	<b>Kết luận</b>	<b>8</b>



# CHAPTER 1

## PHÂN CÔNG

### 1.1 Danh sách thành viên và bảng đánh giá

No.	Fullname	Student ID	Task	Rating
1	Đỗ Ngọc Ánh	2110029	Giao thức	100%
2	Nguyễn Đức Anh Tuấn	2115177	Giao thức	100%
3	Lê Minh Khôi	2233112	Giao thức	100%
4	Nguyễn Tuấn Cường	2110029	Phân tích yêu cầu -	100%

## CHAPTER 2

### PHASE 1

## 2.1 Phân tích yêu cầu

### 2.1.1 Yêu cầu chức năng

#### 2.1.1.1 Client

- **Đăng ký tài khoản:** client có thể đăng ký tài khoản với tên tài khoản và mật khẩu, thiết lập kết nối với server trung tâm, thông báo cho server biết về sự hiện diện của họ, trạng thái kết nối của họ và các file mà họ chứa trong kho lưu trữ cục bộ.
- **Đăng nhập tài khoản:** client sẽ đăng nhập và sử dụng các chức năng của hệ thống
- **Tải lên file cục bộ:** client có thể tải file cục bộ lên kho lưu trữ và siêu dữ liệu của file đó (tên file, mô tả, định dạng, quyền truy cập,...) được chuyển đến server.
- **Yêu cầu tải file:** client có thể yêu cầu các file không có trong kho lưu trữ của họ. Từ đó server sẽ có thể xác định các client khác lưu trữ file được yêu cầu và gửi danh tính của họ cho client yêu cầu, client yêu cầu có thể tìm nạp trực tiếp file từ nút nguồn được xác định có bản sao của file mà không cần sự can thiệp thêm của server.
- **Xem danh sách các file lưu trữ:** client có thể xem danh sách các file có trong kho lưu trữ của mình
- **Xoá file ra khỏi kho lưu trữ:** client có thể xoá file ra khỏi kho lưu trữ của mình.
- **Command-shell interpreter:** client có command prompt đơn giản cho phép client thực hiện các nhiệm vụ khác nhau:
  - publish lname fname: tải lên một file cục bộ (được lưu trong file system của client với tên lname) lên kho lưu trữ của client với tên fname và gửi thông báo tới server
  - fetch fname: tải các bản sao của một file mục tiêu với tên fname và thêm vào kho lưu trữ của client

#### 2.1.1.2 Server

- **Quản lý:** server có thể thêm, xoá các client; server có thể quản lý server và tài nguyên.
- **Xem danh sách các file của client:** server xem danh sách của tất cả client đã đăng ký cũng như các file trong kho lưu trữ của từng client.
- **Xử lý yêu cầu tải file của client:** khi có yêu cầu tải file từ client, server xác định các client khác lưu trữ file được yêu cầu và gửi danh tính của họ cho client yêu cầu.

- **Kiểm tra kết nối của client:** server có thể kiểm tra xem client có đang kết nối với server hay không
- **Command-shell interpreter:** server có command prompt đơn giản cho phép server thực hiện các nhiệm vụ khác nhau:
  - discover hostname: truy xuất danh sách các file trong kho lưu trữ của client với tên hostname.
  - ping hostname: kiểm tra trạng thái hoạt động của client với tên hostname.

### 2.1.2 Yêu cầu phi chức năng

- **Được xây dựng dựa trên bộ giao thức Internet TCP/IP:** Với việc sử dụng hai giao thức này, bộ giao thức TCP/IP cho phép các máy tính kết nối với nhau và giao tiếp, truyền tải các dữ liệu và tài nguyên qua mạng.
- **Bảo mật:**
  - Ứng dụng phải ưu tiên bảo mật khi chia sẻ file, đảm bảo tính bảo mật, tính toàn vẹn và xác thực của các file được truyền cũng như liên lạc giữa client và server.
- **Khả năng mở rộng:** Ứng dụng phải có khả năng mở rộng, có thể xử lý số lượng client và file ngày càng tăng một cách hiệu quả mà không ảnh hưởng đến hiệu suất hoặc độ ổn định.
- **Ổn định:** Ứng dụng làm việc ổn định 24/7, kể cả ngày lễ và giờ cao điểm, có thể downtime trong khung giờ đêm để cập nhật hệ thống (0h-3h).
- **Xử lý đa luồng:** Ứng dụng phải có xử lý đa luồng để nhiều client có thể tải xuống các file khác nhau từ một client mục tiêu tại một thời điểm nhất định.
- **Giao diện:** Trình thông dịch command-shell của client phải có giao diện trực quan và thân thiện với người dùng, giúp người dùng dễ dàng tương tác với các lệnh chia sẻ file.

## 2.2 Giao thức

### 2.2.1 Giao thức tạo và quản lý tài khoản

- **Mô hình áp dụng:** Mô hình Client - Server
- **Giao thức sử dụng:** HTTP
- **Các loại thông điệp được trao đổi:**
  - REQUEST: yêu cầu kết nối
  - RESPONSE: trả về phản hồi kết nối
- **Cú pháp thông điệp (Syntax):**
  - **REQUEST:**
    - \* Request line: Gồm method(GET, POST, HEAD, PUT và DELETE), path và HTTP version.
    - \* Header: Cho phép client gửi thêm các thông tin bổ sung về thông điệp HTTP request và thông tin về chính client.
    - \* Body: là dữ liệu gửi từ client đến server trong gói tin HTTP request.
  - **RESPONSE:**
    - \* Request line: Gồm HTTP version, Status code, Status text.
    - \* Header: có chức năng tương tự như gói tin request.
    - \* Body: Phần Body là nơi đóng gói dữ liệu để trả về cho client.
- **Quy tắc (Rules):**

- Mã trạng thái (code) sẽ được trả về trong trường Method của thông điệp RESPONSE với những quy ước sau:
  - \* 100: Kết nối thành công
  - \* 101: Kết nối thất bại
- Trường Time trong các thông điệp trên sẽ tuân theo định dạng dd/mm/yyyy - hh:mm:ss

## 2.2.2 Giao thức trao đổi File

### 1. Tải metadata của file lên server

- Các loại thông điệp được trao đổi:
  - START (Khởi tạo kết nối)
  - RESPONSE (Phản hồi kết nối)
  - ALERT (Cảnh báo)
  - SEND (Gửi metadata)
- Cú pháp thông điệp (Syntax):
  - Thông điệp START:
    - \* **Method:** START
    - \* **From:** Địa chỉ bên gửi
    - \* **To:** Địa chỉ bên nhận
    - \* **Time:** Thời điểm thực hiện
  - Thông điệp RESPONSE:
    - \* **Method:** RESPONSE <code>
    - \* **From:** Địa chỉ bên gửi
    - \* **To:** Địa chỉ bên nhận
    - \* **Time:** Thời điểm thực hiện
    - \* **Accept:** Trả về connection có được kết nối
  - Thông điệp SEND:
    - \* **Method:** SEND
    - \* **From:** Địa chỉ bên gửi
    - \* **To:** Địa chỉ bên nhận
    - \* **Time:** Thời điểm thực hiện
    - \* **Length:** Kích thước thông điệp
    - \* **Content:** Upload:< lname (Vị trí lưu file) >:< fname (Tên file) >
  - Thông điệp ALERT:
    - \* **Method:** ALERT
    - \* **From:** Địa chỉ bên gửi
    - \* **To:** Địa chỉ bên nhận
    - \* **Time:** Thời điểm thực hiện
    - \* **Length:** Kích thước thông điệp
    - \* **Content:** Nội dung cảnh báo
- Quy tắc (Rules):
  - Mã trạng thái (code) sẽ được trả về trong trường Method của thông điệp RESPONSE với những quy ước sau:
    - \* 100: Kết nối thành công
    - \* 101: Kết nối thất bại
  - Trường Time trong các thông điệp trên sẽ tuân theo định dạng dd/mm/yyyy - hh:mm:ss

## 2. Gửi metadata của các file về client

- Các loại thông điệp được trao đổi:
  - START (Khởi tạo kết nối)
  - RESPONSE (Phản hồi kết nối)
  - ALERT (Cảnh báo)
  - SEND (Gửi, Yêu cầu file metadata)
- Cú pháp thông điệp (Syntax):
  - Thông điệp START:
    - \* **Method:** START
    - \* **From:** Địa chỉ bên gửi
    - \* **To:** Địa chỉ bên nhận
    - \* **Time:** Thời điểm thực hiện
  - Thông điệp RESPONSE:
    - \* **Method:** RESPONSE <code>
    - \* **From:** Địa chỉ bên gửi
    - \* **To:** Địa chỉ bên nhận
    - \* **Time:** Thời điểm thực hiện
    - \* **Accept:** Trả về connection có được kết nối
  - Thông điệp SEND:
    - \* **Method:** SEND
    - \* **From:** Địa chỉ bên gửi
    - \* **To:** Địa chỉ bên nhận
    - \* **Time:** Thời điểm thực hiện
    - \* **Length:** Kích thước thông điệp
    - \* **Content:** Metadata (hoặc yêu cầu Metadata)
  - Thông điệp ALERT:
    - \* **Method:** ALERT
    - \* **From:** Địa chỉ bên gửi
    - \* **To:** Địa chỉ bên nhận
    - \* **Time:** Thời điểm thực hiện
    - \* **Length:** Kích thước thông điệp
    - \* **Content:** Nội dung cảnh báo
- Quy tắc (Rules):
  - Mã trạng thái (code) sẽ được trả về trong trường Method của thông điệp RESPONSE với những quy ước sau:
    - \* 100: Kết nối thành công
    - \* 101: Kết nối thất bại
  - Trường Time trong các thông điệp trên sẽ tuân theo định dạng dd/mm/yyyy
  - hh:mm:ss

## 3. Trao đổi file

- Các loại thông điệp được trao đổi:
  - START (Khởi tạo kết nối)
  - RESPONSE (Phản hồi kết nối)
  - ALERT (Cảnh báo)
  - SEND (Gửi, Yêu cầu File)
- Cú pháp thông điệp (Syntax):

- **Thông điệp START:**
  - \* **Method:** START
  - \* **From:** Địa chỉ bên gửi
  - \* **To:** Địa chỉ bên nhận
  - \* **Time:** Thời điểm thực hiện
- **Thông điệp RESPONSE:**
  - \* **Method:** RESPONSE <code>
  - \* **From:** Địa chỉ bên gửi
  - \* **To:** Địa chỉ bên nhận
  - \* **Time:** Thời điểm thực hiện
  - \* **Accept:** Trả về connection có được kết nối
- **Thông điệp SEND:**
  - \* **Method:** SEND
  - \* **From:** Địa chỉ bên gửi
  - \* **To:** Địa chỉ bên nhận
  - \* **Time:** Thời điểm thực hiện
  - \* **Length:** Kích thước thông điệp
  - \* **Content:** Nội dung file (hoặc yêu cầu file)
  - \* **Type file:** Loại file được gửi
- **Thông điệp ALERT:**
  - \* **Method:** ALERT
  - \* **From:** Địa chỉ bên gửi
  - \* **To:** Địa chỉ bên nhận
  - \* **Time:** Thời điểm thực hiện
  - \* **Length:** Kích thước thông điệp
  - \* **Content:** Nội dung cảnh báo
- **Quy tắc (Rules):**
  - Mã trạng thái (code) sẽ được trả về trong trường Method của thông điệp RESPONSE với những quy ước sau:
    - \* 100: Kết nối thành công
    - \* 101: Kết nối thất bại
  - Trường Time trong các thông điệp trên sẽ tuân theo định dạng dd/mm/yyyy
  - hh:mm:ss





## CHAPTER 3

KẾT LUẬN