

# Mục lục

<b>1</b>	<b>Thông tin của nhóm</b>	<b>1</b>
<b>2</b>	<b>Mức độ hoàn thành</b>	<b>2</b>
<b>3</b>	<b>Kịch bản giao tiếp của chương trình</b>	<b>3</b>
3.1	Giao thức trao đổi giữa client và server: UDP . . . . .	3
3.2	Cấu trúc của thông điệp: . . . . .	3

## Danh sách hình vẽ

## Danh sách bảng

1	Bảng phân công thành viên . . . . .	1
2	Bảng mức độ hoàn thành . . . . .	2

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO ĐỒ ÁN MÔN HỌC**  
**MẠNG MÁY TÍNH**  
**ĐỀ TÀI: Ứng dụng Socket – Địa điểm yêu thích**

**Giảng viên lý thuyết:** Thầy Đỗ Hoàng Cường

**Giảng viên hướng dẫn thực hành:**

- Thầy Lê Hà Minh
- Thầy Nguyễn Thanh Quân

**Lớp:** 20TN

**Thành viên thực hiện:**

- 20120131 – Nguyễn Văn Lộc
- 20120536 – Võ Trọng Nghĩa
- 20120572 – Nguyễn Kiều Minh Tâm

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 4–5 NĂM 2022**

# 1 Thông tin của nhóm

MSSV	Họ và tên	Công việc
20120131	Nguyễn Văn Lộc	Thiết kế các hàm ở client và GUI
20120536	Võ Trọng Nghĩa	Thiết kế giao thức truyền dữ liệu giữa client và server
20120572	Nguyễn Kiều Minh Tâm	Thiết kế GUI

Bảng 1: Bảng phân công thành viên

## 2 Mức độ hoàn thành

STT	Yêu cầu	Công việc	Đánh giá	Điểm tự đánh giá
1	Truy vấn danh sách địa điểm	Clients truy vấn được danh sách địa điểm do server quản lý. Các thông tin hiển thị: mã số địa điểm, tên địa điểm.	Hoàn thành	2
2	Truy vấn thông tin 1 địa điểm	Clients truy vấn được thông tin chi tiết về 1 địa điểm do server quản lý. Các thông tin hiển thị: mã số địa điểm, tên địa điểm, tọa độ địa điểm, mô tả.	Hoàn thành	2
3	Quản lý dữ liệu tại server bằng các loại file có cấu trúc như XML, JSON hoặc CSDL quan hệ	Dữ liệu được quản lý tại server bằng file JSON.	Hoàn thành	1
4	Cho phép tải hình ảnh đại diện từ server về client cho tất cả các địa điểm đang được quản lý.	Clients có thể tải về hình ảnh đại diện của các địa điểm từ server. Hình ảnh được hiển thị lên GUI của ứng dụng sau khi tải về.	Hoàn thành	2
5	Cho phép tải về các hình ảnh của 1 địa điểm từ server về client.	Clients có thể tải về các hình ảnh của một địa điểm từ server. Hình ảnh được hiển thị lên GUI của ứng dụng sau khi tải về.	Hoàn thành	2
6	Hỗ trợ nhiều clients truy cập đồng thời đến server.	Cho phép nhiều clients truy cập đồng thời đến server.	Hoàn thành	1

Bảng 2: Bảng mức độ hoàn thành

### 3 Kịch bản giao tiếp của chương trình

#### 3.1 Giao thức trao đổi giữa client và server: UDP

#### 3.2 Cấu trúc của thông điệp:

Một thông điệp có tổng cộng 1024 bytes.

##### 1. Đối với thông điệp từ server có 2 loại sau đây:

- (a) *Thông điệp gửi số lượng thông điệp cần nhận cho data: |padding|len|*  
 Có dạng <1021 bytes 0>xyz (xyz là ba chữ số).  
 Gồm 2 trường theo thứ tự đó với các mô tả chi tiết sau đây:
  - i. padding: Gồm các null bytes (các bytes \x00) chèn vào trước trường len sao cho tổng thông điệp có đủ 1024 bytes.
  - ii. len: kiểu str, thể hiện số lượng thông điệp (tối đa 3 bytes).
- (b) *Thông điệp gửi từng phân data: |ID|data|hash|*  
 Có dạng xyz<data: 981><hash: 40> (xyz là ba chữ số).  
 Gồm 3 trường theo thứ tự đó với các mô tả chi tiết sau đây:
  - i. ID: Kiểu str gồm 3 ký tự thể hiện số trong đoạn [000; 999] (3 bytes).
  - ii. data: Kiểu byte, gồm 981 (1024 - 43) ký tự (981 bytes).
  - iii. hash: Kiểu str, gồm 40 ký tự (40 bytes) được tạo khi hash data bằng thuật toán SHA1 để kiểm lỗi. 40 bytes này mô tả một số nguyên 20 bytes = 160 bits trong hệ thập lục phân.

##### 2. Đối với thông điệp từ client có 6 loại sau đây:

- (a) *Thông điệp xác nhận số lượng: |specifier|len|padding|*  
 Có dạng ACK\_LEN\_xyz<1013 bytes 0> (xyz là ba chữ số).  
 Gồm 3 trường theo thứ tự đó với các mô tả chi tiết sau đây:
  - i. specifier: kiểu str, là chuỗi ký tự "ACK\_LEN\_" (8 bytes)
  - ii. len: kiểu str, là số lượng thông điệp đã nhận từ server, gồm 3 ký tự thể hiện số trong đoạn [000; 999] (3 bytes)
  - iii. padding: gồm 1013 (1024 - 8 - 3) bytes \x00.
- (b) *Thông điệp xác nhận phần data: |specifier|id|padding|*  
 Có dạng ACK\_xyz<1017 bytes 0> (xyz là ba chữ số).  
 Gồm 3 trường theo thứ tự đó với các mô tả chi tiết sau đây:
  - i. specifier: kiểu str, là chuỗi ký tự "ACK\_" (4 bytes)
  - ii. id: kiểu str, là thứ tự của thông điệp nhận từ server, gồm 3 ký tự thể hiện số trong đoạn [000; 999] (3 bytes)
  - iii. padding: gồm 1017 (1024 - 4 - 3) bytes \x00.
- (c) *Thông điệp yêu cầu thông tin toàn bộ địa điểm: |command|padding|*  
 Có dạng GIV\_ALL<1017 bytes 0>  
 Gồm 2 trường theo thứ tự đó với các mô tả chi tiết sau đây:

- i. **command**: kiểu **str**, là chuỗi ký tự "GIV\_ALL" (7 bytes)
  - ii. **padding**: gồm 1017 (1024 - 7) bytes \x00.
- (d) *Thông điệp yêu cầu thông tin chi tiết 1 địa điểm*: |command|id|padding|  
 Có dạng GIV\_DETAIL\_<id: x bytes><1013 - x bytes 0>  
 Gồm 3 trường theo thứ tự đó với các mô tả chi tiết sau đây:
- i. **command**: kiểu **str**, là chuỗi ký tự "GIV\_DETAIL\_" (11 bytes)
  - ii. **id**: kiểu **str**, thể hiện id của địa điểm cần lấy.
  - iii. **padding**: Gồm các bytes \x00 điền cho đủ 1024 bytes.
- (e) *Thông điệp yêu cầu ảnh đại diện 1 địa điểm*: |command|id|padding |  
 Có dạng GIV\_AVT\_<id: x bytes><1016 - x bytes 0>  
 Gồm 3 trường theo thứ tự đó với các mô tả chi tiết sau đây:
- i. **command**: kiểu **str**, là chuỗi ký tự "GIV\_AVT\_" (8 bytes)
  - ii. **id**: kiểu **str**, thể hiện id của địa điểm cần lấy.
  - iii. **padding**: Gồm các bytes \x00 điền cho đủ 1024 bytes.
- (f) *Thông điệp yêu cầu ảnh tại 1 địa điểm*: |command|position|separator|id|padding|  
 Có dạng GIV\_IMG\_xyz\_<id: t bytes><1012 - t bytes 0> (xyz là ba chữ số).  
 Gồm 5 trường theo thứ tự đó với các mô tả chi tiết sau đây:
- i. **command**: kiểu **str**, là chuỗi ký tự "GIV\_IMG\_" (8 bytes)
  - ii. **position**: kiểu **str**, là thứ tự trong danh sách ảnh cần lấy, gồm 3 ký tự thể hiện số trong đoạn [000; 999] (3 bytes)
  - iii. **separator**: kiểu **char**, là ký tự '\_'
  - iv. **id**: kiểu **str**, thể hiện id của địa điểm cần lấy.
  - v. **padding**: Gồm các bytes \x00 điền cho đủ 1024 bytes.