

Đại học Quốc gia Thành phố Hồ Chí Minh

Trường Đại học Khoa học Tự nhiên



ĐỒ ÁN

LẬP TRÌNH SOCKET

ĐỀ 2 - ĐỊA ĐIỂM YÊU THÍCH

Nhóm sinh viên thực hiện:

20120012 - Nguyễn Phạm Nhật Huy

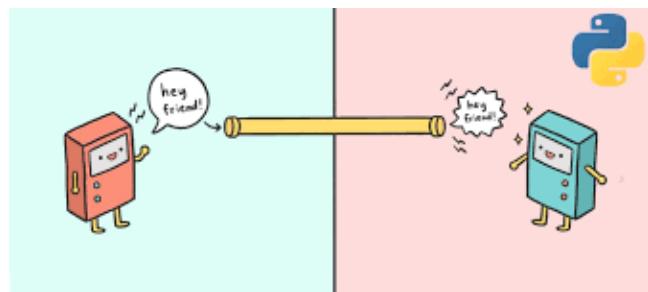
20120014 - Vương Gia Huy

20120021 - Hồ Văn Sơn

Giảng viên hướng dẫn: Lê Hà Minh, Nguyễn Thanh Quân

Lớp 20TN

Học phần: Mạng máy tính



Mục Lục

1	Tìm hiểu sơ lược	2
1.1	Socket là gì?	2
1.2	Người dùng cần đến Socket?	2
1.3	Socket có mấy kiểu?	2
1.4	Thành phần chính là gì?	2
1.4.1	Số hiệu cổng (Port Number) của socket	2
1.4.2	Địa chỉ IP	2
1.5	Các chế độ giao tiếp	3
1.5.1	Chế độ có nối kết (TCP)	3
1.5.2	Chế độ không nối kết (UDP)	3
1.6	Sơ lược về mô hình lập trình socket bằng python	3
2	Mô hình Client-Server sử dụng Socket ở chế độ không nối kết (UDP) . . .	4
2.1	Giai đoạn 1: Server tạo Socket - gán số hiệu cổng	4
2.2	Giai đoạn 2: Client tạo Socket	4
2.3	Giai đoạn 3: Trao đổi thông tin giữa Client và Server	4
3	Cách thức tổ chức chương trình quản lý các địa điểm yêu thích sử dụng Socket (giao thức UDP)	6
3.1	Kịch bản giao tiếp của chương trình	6
3.2	Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng	6
3.3	Chương trình và các hàm chức năng	6
3.3.1	Áp dụng module socket trong python	6
3.3.2	Client	7
3.3.3	Server	7
4	Các chức năng và hướng dẫn sử dụng	9
4.1	Giao diện	9
4.2	Truy vấn danh sách các địa điểm được quản lý	9
4.3	Truy vấn 1 địa điểm đang được server quản lý	12
4.4	Cho phép tải ảnh avatar về	15
4.5	Cho phép tải ảnh chi tiết của một địa điểm cụ thể	19
4.6	Hỗ trợ nhiều client truy cập đồng thời đến server	22
5	Tiêu chí đánh giá	23
5.1	Tự đánh giá	23
5.2	Bảng phân công công việc	23
6	Tài liệu tham khảo	24

1. Tìm hiểu sơ lược

1.1. Socket là gì?

Trong liên kết truyền thông 2 chiều (two-way communication), socket là điểm cuối (endpoint) biểu diễn kết nối giữa client/server. Cụ thể, khi bạn viết một ứng dụng và cần tương tác với ứng dụng khác, chúng ta sẽ dựa vào mô hình client/server. Ứng dụng có khả năng phục vụ, cung cấp thông tin cho bạn thì được gọi là server, còn ứng dụng gửi yêu cầu đến server gọi là client.

Để yêu cầu được server thực hiện điều gì đó, client phải có khả năng kết nối đến server. Cơ chế kết nối này được trừu tượng hóa gọi là socket.

1.2. Người dùng cần đến Socket?

Socket là một giao diện lập trình ứng dụng (API-Application Programming Interface). Nó được giới thiệu lần đầu tiên trong ấn bản UNIX - BSD 4.2. dưới dạng các hàm hệ thống theo cú pháp ngôn ngữ C (socket(), bind(), connect(), send(), receive(), read(), write(), close(), ...).

Ngày nay, Socket được hỗ trợ trong hầu hết các hệ điều hành như MS Windows (WinSock), Linux và được sử dụng trong nhiều ngôn ngữ lập trình khác nhau: như C, C++, Java, Visual Basic, Visual C++, ... Đặc biệt, người dùng cũng có thể chạy cùng một lúc nhiều socket liên tục, giúp nâng cao hiệu suất làm việc, cũng như tiết kiệm thêm nhiều thời gian và công sức hơn.

1.3. Socket có mấy kiểu?

Socket có 2 kiểu:

- Socket kiểu AF_UNIX chỉ cho phép giao tiếp giữa các tiến trình trong cùng một máy tính.
- Socket kiểu AF_INET cho phép giao tiếp giữa các tiến trình trên những máy tính khác nhau trên mạng.

1.4. Thành phần chính là gì?

Có 2 thành phần chính là số hiệu cổng (Port Number) và địa chỉ IP.

1.4.1. Số hiệu cổng (Port Number) của socket

Để có thể thực hiện các cuộc giao tiếp, một trong hai tiến trình phải công bố số hiệu cổng của socket mà mình sử dụng. Mỗi cổng giao tiếp thể hiện một địa chỉ xác định trong một hệ thống. Khi tiến trình được gán một số hiệu cổng, nó có thể nhận dữ liệu gửi đến cổng này từ các tiến trình khác. Tiến trình còn lại cũng yêu cầu tạo ra một socket.

1.4.2. Địa chỉ IP

Ngoài số hiệu cổng, hai bên giao tiếp còn phải biết địa chỉ IP của nhau. Địa chỉ IP giúp phân biệt máy tính này với máy tính kia trên mạng TCP/IP. Trong khi số hiệu cổng dùng để phân biệt các quá trình khác nhau trên cùng một máy tính.

1.5. Các chế độ giao tiếp

Tầng vận chuyển (transport layer) giúp chuyển tiếp các thông điệp giữa các chương trình ứng dụng với nhau. Nó có thể hoạt động theo hai chế độ:

- TCP (Transmission Control Protocol): có nối kết
- UDP (User Datagram Protocol): không nối kết

Socket là giao diện giữa chương trình ứng dụng với tầng vận chuyển. Nó cho phép ta chọn giao thức sử dụng ở tầng vận chuyển là TCP hay UDP cho chương trình ứng dụng của mình.

1.5.1. Chế độ có nối kết (TCP)

- Tồn tại kênh giao tiếp ảo giữa hai bên giao tiếp
- Dữ liệu được gởi đi theo chế độ bảo đảm: có kiểm tra lỗi, truyền lại gói tin lỗi hay mất, bảo đảm thứ tự đến của các gói tin ...
- Dữ liệu chính xác, Tốc độ truyền chậm.

1.5.2. Chế độ không nối kết (UDP)

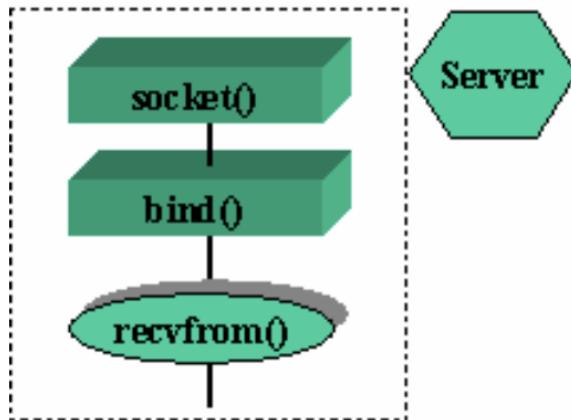
- Không tồn tại kênh giao tiếp ảo giữa hai bên giao tiếp
- Dữ liệu được gởi đi theo chế độ không bảo đảm: Không kiểm tra lỗi, không phát hiện không truyền lại gói tin bị lỗi hay mất, không bảo đảm thứ tự đến của các gói tin . . .
- Dữ liệu không chính xác, tốc độ truyền nhanh.
- Thích hợp cho các ứng dụng cần tốc độ, không cần chính xác cao: truyền âm thanh, hình ảnh . . .

1.6. Sơ lược về mô hình lập trình socket bằng python

- 1 - Chúng ta mở một socket – socket() để tạo 1 ô cắm socket cho Server. Đây là quá trình Hệ điều hành phân bổ tài nguyên, chuẩn bị kết nối. Bạn cần chỉ định tên hoặc số hiệu port cho socket để Client biết đến ô cắm của Server.
- 2 - Chúng ta liên kết máy chủ với host hoặc một máy và một port – bind().
- 3 - Server sẽ bắt đầu lắng nghe các kết nối từ Client đưa đến trên port – listen().
- 4 - Một yêu cầu kết nối được gửi từ client tới server – connect(). Server chấp nhận yêu cầu của client, kết nối từ đó được thiết lập – accept().
- 5 - Đã có thể gửi và nhận tin – read() / write() tương tự dùng lệnh read/write để đọc ghi trên tập tin. Socket dựa vào số mô tả (socket descriptor) để xác định cần đọc ghi cho hàm read/write.
- 6 - Đóng kết nối – close()

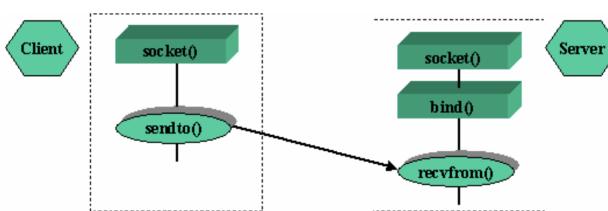
2. Mô hình Client-Server sử dụng Socket ở chế độ không nối kết (UDP)

2.1. Giai đoạn 1: Server tạo Socket - gán số hiệu cổng



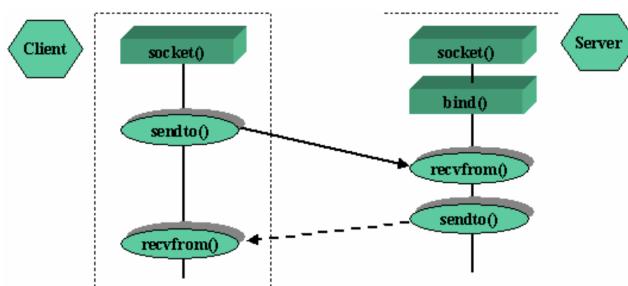
- `socket()`: Server yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển.
- `bind()`: Server yêu cầu gán số hiệu cổng cho socket.

2.2. Giai đoạn 2: Client tạo Socket



- `socket()`: Client yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển, thông thường hệ thống tự động gán một số hiệu cổng còn rảnh cho socket của Client.

2.3. Giai đoạn 3: Trao đổi thông tin giữa Client và Server



- Sau khi tạo Socket xong, Client và Server có thể trao đổi thông tin qua lại với nhau thông qua hai hàm `send()` và `receive()`.

- Đơn vị dữ liệu trao đổi giữa Client và Server là các Datagram Package (Gói tin thư tín).
- Protocol của ứng dụng phải định nghĩa khuôn dạng và ý nghĩa của các Datagram Package. Mỗi Datagram Package có chứa thông tin về địa chỉ người gửi và người nhận (IP, Port).

3. Cách thức tổ chức chương trình quản lý các địa điểm yêu thích sử dụng Socket (giao thức UDP)

3.1. Kịch bản giao tiếp của chương trình

- Giao thức trao đổi giữa client và server là UDP (User Datagram Protocol) đã được nêu rõ ở phần trên.
- Cấu trúc thông điệp bao gồm những câu lệnh:
 - get all information: Dùng để lấy tất cả các địa điểm có trong cơ sở dữ liệu (database).
 - get information id ID: Dùng để lấy thông tin bao quát của địa điểm có mã số là ID.
 - get avatar ID, IMGSIZE: Dùng để xuất ra ảnh của địa điểm có mã số là ID và với kích thước là IMGSIZE.
 - get single detail image ID, INDEX_IMG, IMGSIZE: Dùng để xuất ra ảnh cụ thể của địa điểm có mã số là ID, số thứ tự là INDEX_IMG và kích thước IMGSIZE.
- Kiểu dữ liệu của các thông điệp là kiểu **string**.
- Cách tổ chức cơ sở của nhóm là lưu trong file **json**, địa điểm A bất kì trong database gồm các thông tin sau:
 - id: là mã số trong cơ sở dữ liệu của địa điểm A.
 - name: là tên của địa điểm A.
 - latitude: là vĩ độ của địa điểm A.
 - longitude: là kinh độ của địa điểm A.
 - description: là mô tả về địa điểm A.
 - avatar: là ảnh đại diện cho địa điểm A.
 - detail_image: gồm các ảnh mô tả cụ thể về địa điểm A.

3.2. Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng

- Môi trường lập trình mà nhóm sử dụng là visual code (Windows), terminal (MacOs).
- **socket**, **numpy**, **cv2**, **PIL**, **io** là các framework/package hỗ trợ để thực thi ứng dụng được nhóm áp dụng.

3.3. Chương trình và các hàm chức năng

3.3.1. Áp dụng module socket trong python

```
1 socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

- AF_INET: là thiết lập dưới dạng ipv4.
- SOCK_DGRAM: là thiết lập giao thức UDP cho socket.

Các phương thức và mô tả:

- bind(address, port): Phương thức này được dùng để lắng nghe đến địa chỉ address và port.
- listen(backlog): Phương thức này thiết lập kết nối trên server, với tham số truyền vào là số kết nối được phép (nhỏ nhất là 0 và lớn nhất là do cấu hình của server).
- accept(): Phương thức này thiết lập chấp nhận một kết nối, và nó sẽ trả về một tuple gồm 2 thông số (conn, address) để chúng ta có thể gửi ngược về client.
- connect(address): Phương thức này thiết lập một kết nối từ client đến server.
- recvfrom(bufsize, flag): Phương thức này dùng để nhận dữ liệu qua giao thức UDP.
- sendto(bytes, address): Phương thức này dùng để gửi dữ liệu qua giao thức UDP.
- close(): Phương thức này dùng để đóng một kết nối.

3.3.2. Client

Ở sâu bên trong chương trình chúng tôi tạo ra lớp đối tượng client áp dụng các module socket ở trên bao gồm các khả năng gửi yêu cầu (request) đến server và nhận phản hồi (respond) từ server. Nó áp dụng phương thức **sendto** để gửi yêu cầu đến server và dùng **recvfrom** để nhận phản hồi từ server. Cụ thể hơn bao gồm các chức năng:

- Client truy vấn danh sách các địa điểm đang được server quản lý thông qua hàm **request_all_information**. Khi đó, client sẽ gửi thông điệp **get all information** đến server và hàm sẽ trả về là phản hồi mà server gửi lại.
- Client truy vấn 1 địa điểm A bất kỳ đang được server quản lý thông qua hàm **request_information_id**. Khi đó, client sẽ gửi thông điệp **get information id ID_A** đến server và hàm sẽ trả về là phản hồi mà server gửi lại.
- Client có thể gửi đề nghị xem ảnh đại diện của một địa điểm bất kỳ A với kích thước **IMG_SIZE** thông qua hàm **request_avatar**. Khi đó, client sẽ gửi thông điệp **get avatar ID_A IMG_SIZE** qua server để yêu cầu server phản hồi lại ảnh đại diện của địa điểm A với kích thước **IMG_SIZE** mà client mong muốn.
- Client có thể gửi đề nghị xem ảnh chi tiết của một địa điểm bất kỳ A với kích thước **IMG_SIZE** thông qua hàm **request_detail_image**. Khi đó, client sẽ gửi thông điệp **get detail image ID_A IMG_SIZE** qua server để yêu cầu server phản hồi lại ảnh chi tiết của địa điểm A với kích thước **IMG_SIZE** mà client mong muốn. Hơn nữa, có thể không xem toàn bộ ảnh chi tiết mà chỉ xem một ảnh chi tiết nào đó dựa vào việc gọi hàm **send_detail_image_id**, nó sẽ mang thông điệp **get single detail image** đến cho server xử lý.

3.3.3. Server

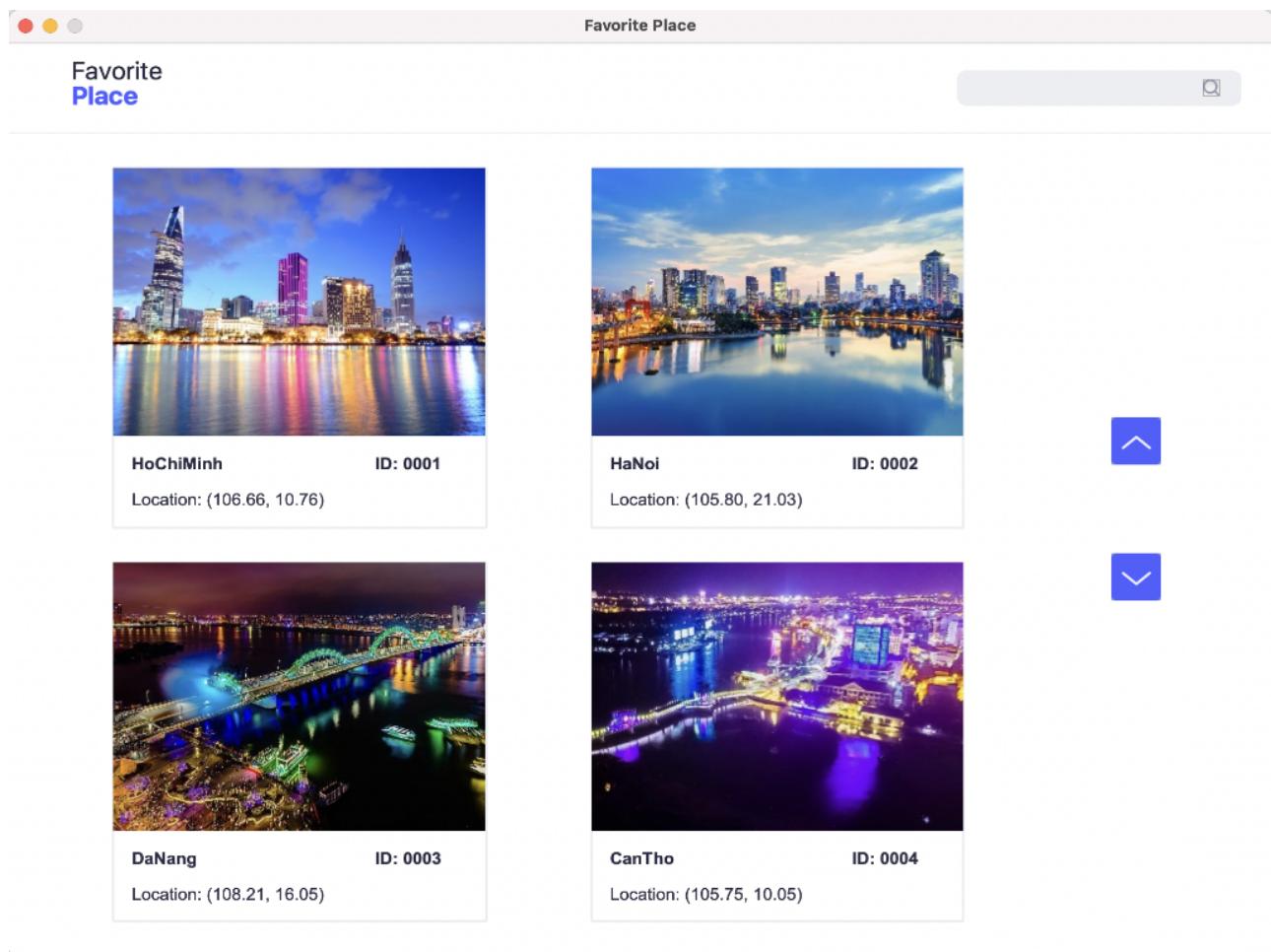
Hoàn toàn tương tự, ta cũng áp dụng các module socket ở trên sâu bên trong chương trình tạo ra lớp đối tượng server bao gồm các khả năng nhận yêu cầu (request) từ client và gửi phản hồi (respond) đến client. Nó áp dụng phương thức **recvfrom** để nhận yêu cầu từ client và dùng

sendto để gửi phản hồi đến client. Thông điệp được server gửi đến client dạng bytes và gửi tuần tự theo một độ lớn **BUFFER_SIZE** cho trước (ở chương trình của chúng tôi thì con số này là 2^{15}). Cụ thể hơn bao gồm các chức năng:

- Server nhận đề nghị truy vấn danh sách các địa điểm đang quản lý từ client thông qua hàm **recv_request**. Khi đó, server sẽ gửi phản hồi lại cho client bằng cách gọi hàm **send_all_information**.
- Server cho phép client truy vấn 1 địa điểm A bất kì đang server quản lý thông qua hàm **recv_request**. Khi đó, server sẽ gửi phản hồi lại cho client bằng việc gọi hàm **send_information_id**.
- Có thể gửi phản hồi về ảnh đại diện của địa điểm A với kích thước **IMG_SIZE** bất kì khi được client gửi yêu cầu. Và sẽ gửi phản hồi lại cho client thông qua hàm **send_avatar_id**.
- Client được phép gửi yêu cầu cho server xem ảnh chi tiết của một địa điểm bất kì A với kích thước **IMG_SIZE**. Khi đó, server sẽ gửi phản hồi lại cho client dựa vào hàm **send_detail_image**. Hơn nữa, server có thể nhận yêu cầu chỉ gửi 1 ảnh chi tiết thay vì gửi toàn bộ ảnh chi tiết, việc phản hồi sẽ được giải quyết thông qua hàm **send_single_image_id**.

4. Các chức năng và hướng dẫn sử dụng

4.1. Giao diện

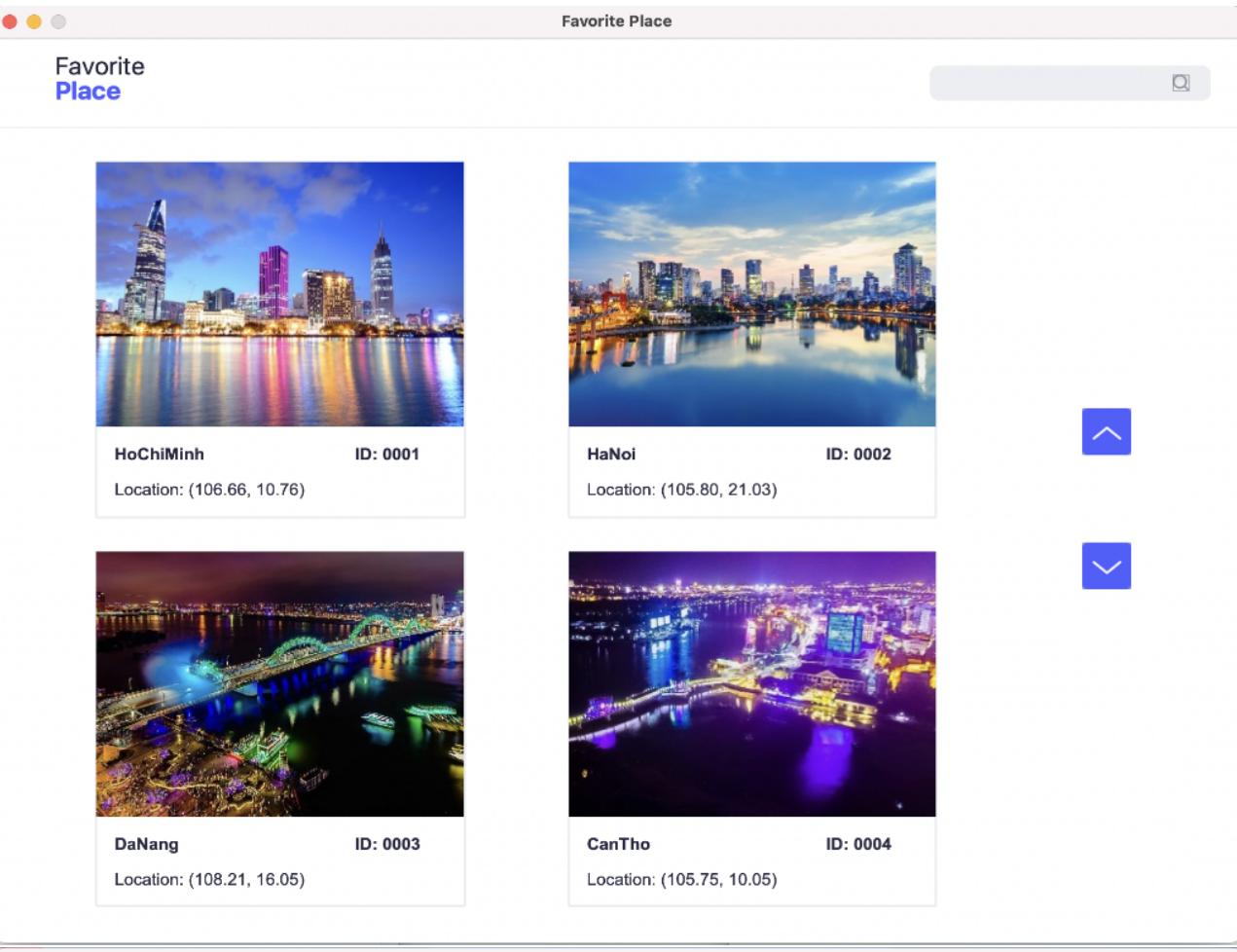


4.2. Truy vấn danh sách các địa điểm được quản lý

Client có thể truy vấn danh sách các địa điểm được quản lý với server thông qua việc nhấn vào mũi tên lén xuống. Chẳng hạn như trong database của nhóm chúng tôi có 20 địa điểm thì ta hoàn toàn có thể truy vấn được hết.

Favorite Place

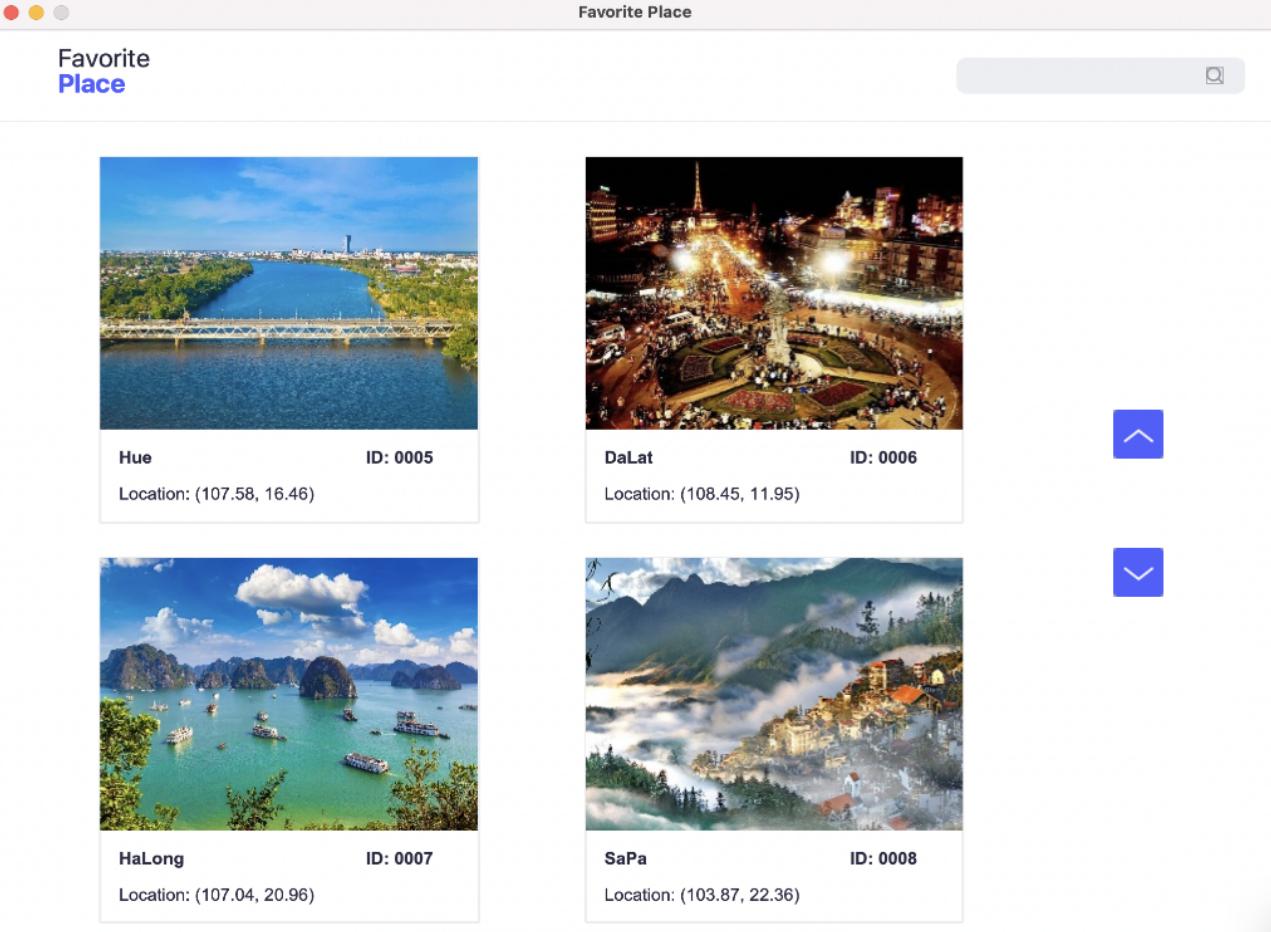
Favorite Place



City	ID	Location
HoChiMinh	ID: 0001	(106.66, 10.76)
HaNoi	ID: 0002	(105.80, 21.03)
DaNang	ID: 0003	(108.21, 16.05)
CanTho	ID: 0004	(105.75, 10.05)

Favorite Place

Favorite Place



City	ID	Location
Hue	ID: 0005	(107.58, 16.46)
DaLat	ID: 0006	(108.45, 11.95)
HaLong	ID: 0007	(107.04, 20.96)
SaPa	ID: 0008	(103.87, 22.36)

Favorite Place



NhaTrang ID: 0009
Location: (109.20, 12.25)



PhuQuoc ID: 0010
Location: (103.98, 10.17)



NewYork ID: 0011
Location: (-74.01, 40.71)



ShangHai ID: 0012
Location: (121.47, 31.23)



Beijing ID: 0013
Location: (116.41, 39.90)



Chongqing ID: 0014
Location: (106.55, 29.56)

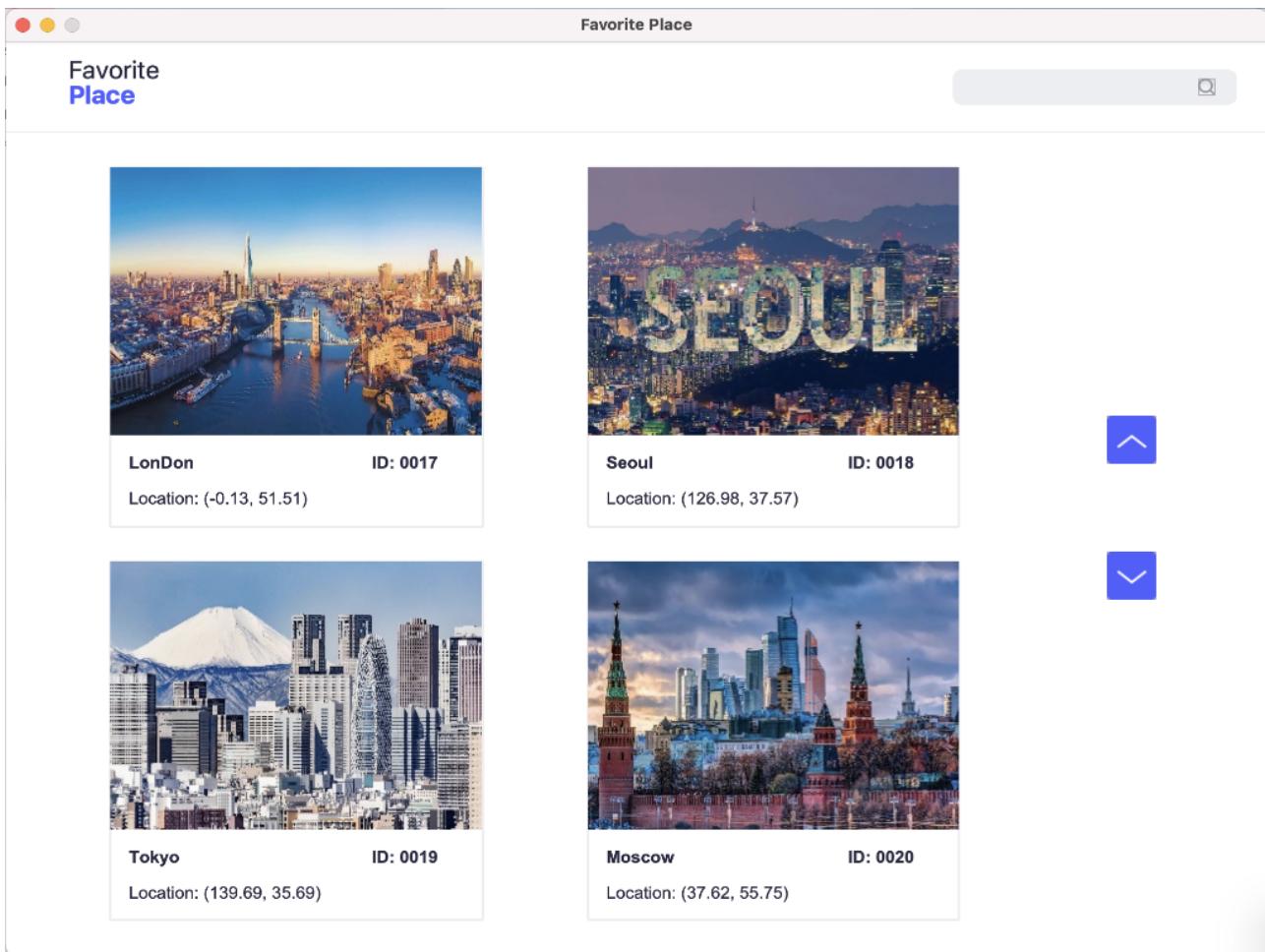
Favorite Place



HongKong ID: 0015
Location: (114.11, 22.40)



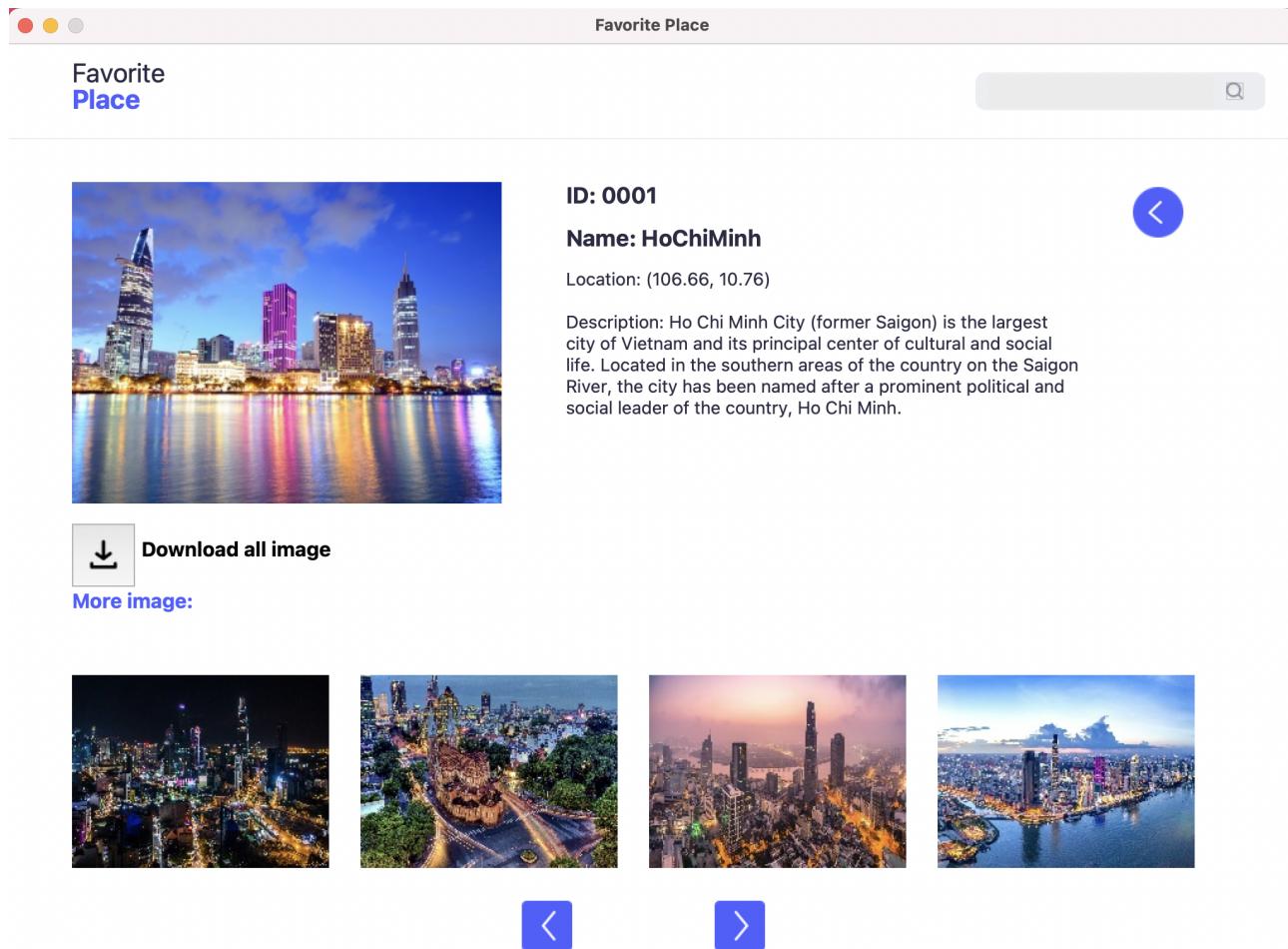
Paris ID: 0016
Location: (2.35, 48.86)



4.3. Truy vấn 1 địa điểm đang được server quản lý

Client có thể truy vấn 1 địa điểm cụ thể được quản lý với server thông qua 2 cách:

- Dựa con trỏ chuột nhấp vào hình ảnh địa điểm đó.



Favorite Place

ID: 0001

Name: HoChiMinh

Location: (106.66, 10.76)

Description: Ho Chi Minh City (former Saigon) is the largest city of Vietnam and its principal center of cultural and social life. Located in the southern areas of the country on the Saigon River, the city has been named after a prominent political and social leader of the country, Ho Chi Minh.

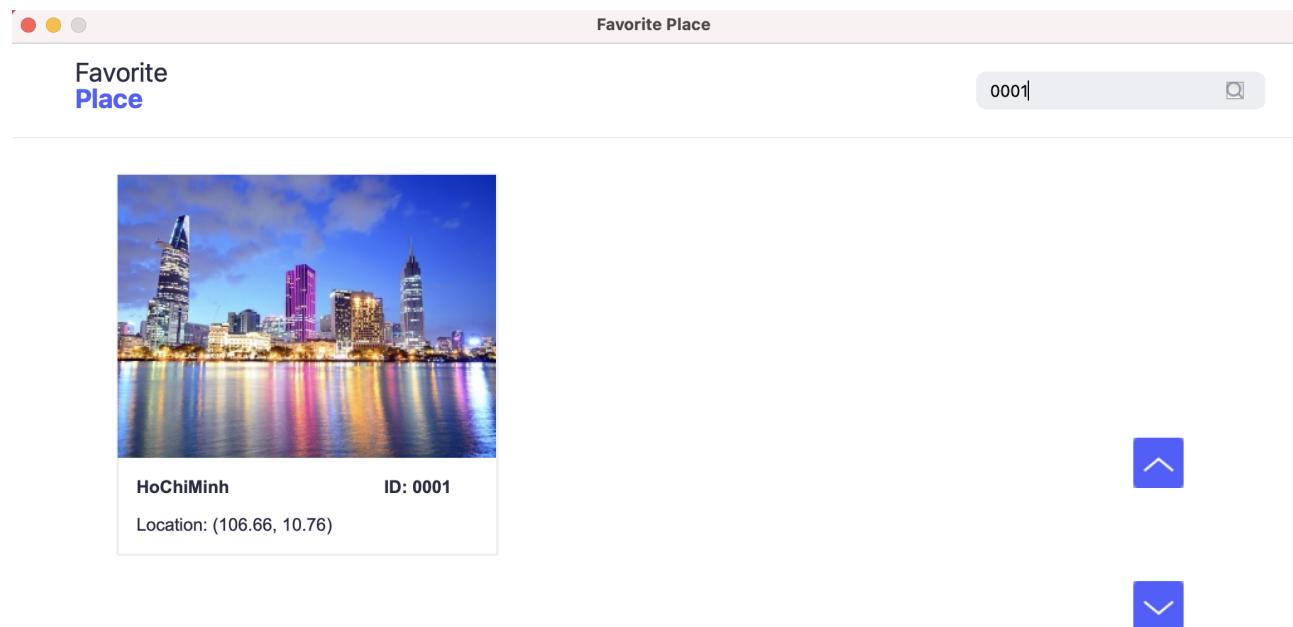
 [Download all image](#)

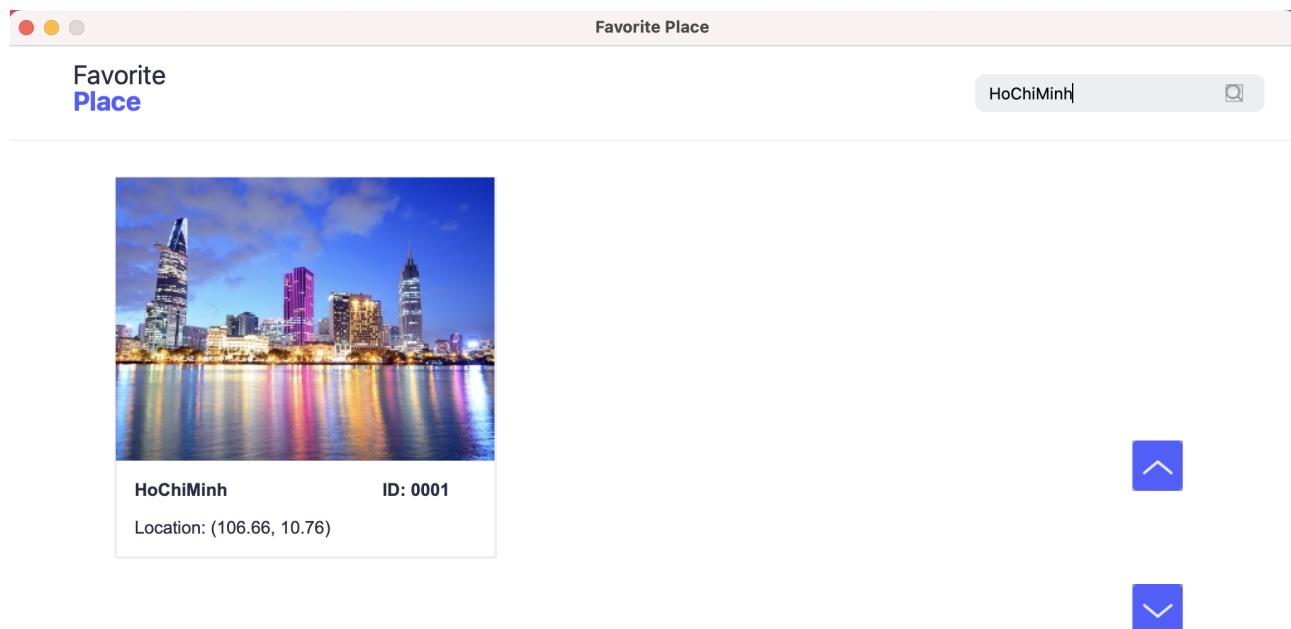
[More image:](#)

[<](#) [>](#)

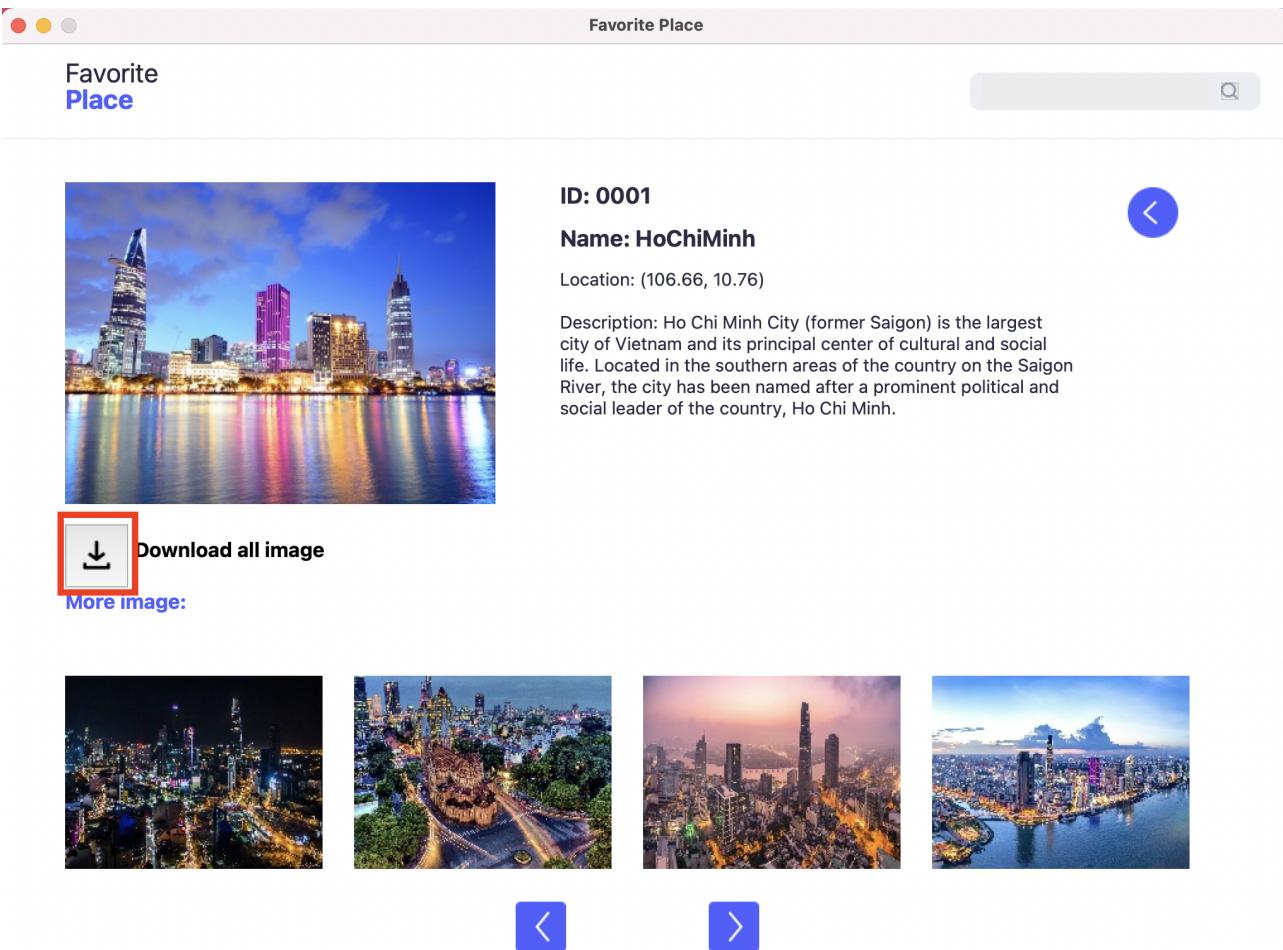
- Nhập mã số hoặc tên địa điểm trên thanh tìm kiếm. Khi đó ảnh đại diện của địa điểm cần tìm sẽ hiện lên và click vào như cách đầu tiên.





4.4. Cho phép tải ảnh avatar về

Khi ta đã vào xem thông tin của một địa điểm cụ thể nào đó và ta muốn tải về toàn bộ ảnh thì ta có thể click vào nút được đóng khung bên dưới có tên là **Download all image**.



Favorite Place

ID: 0001

Name: HoChiMinh

Location: (106.66, 10.76)

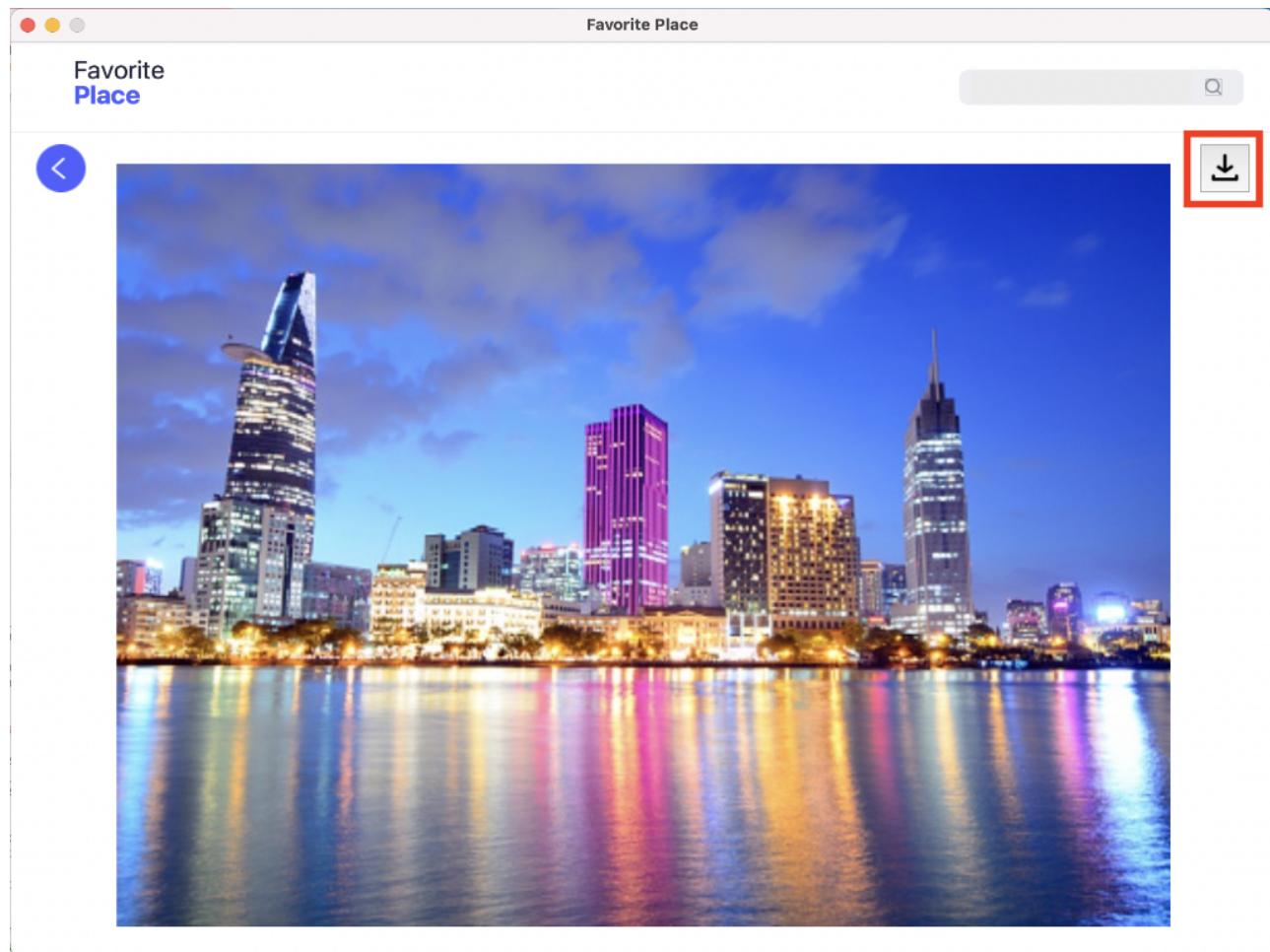
Description: Ho Chi Minh City (former Saigon) is the largest city of Vietnam and its principal center of cultural and social life. Located in the southern areas of the country on the Saigon River, the city has been named after a prominent political and social leader of the country, Ho Chi Minh.

Download all image

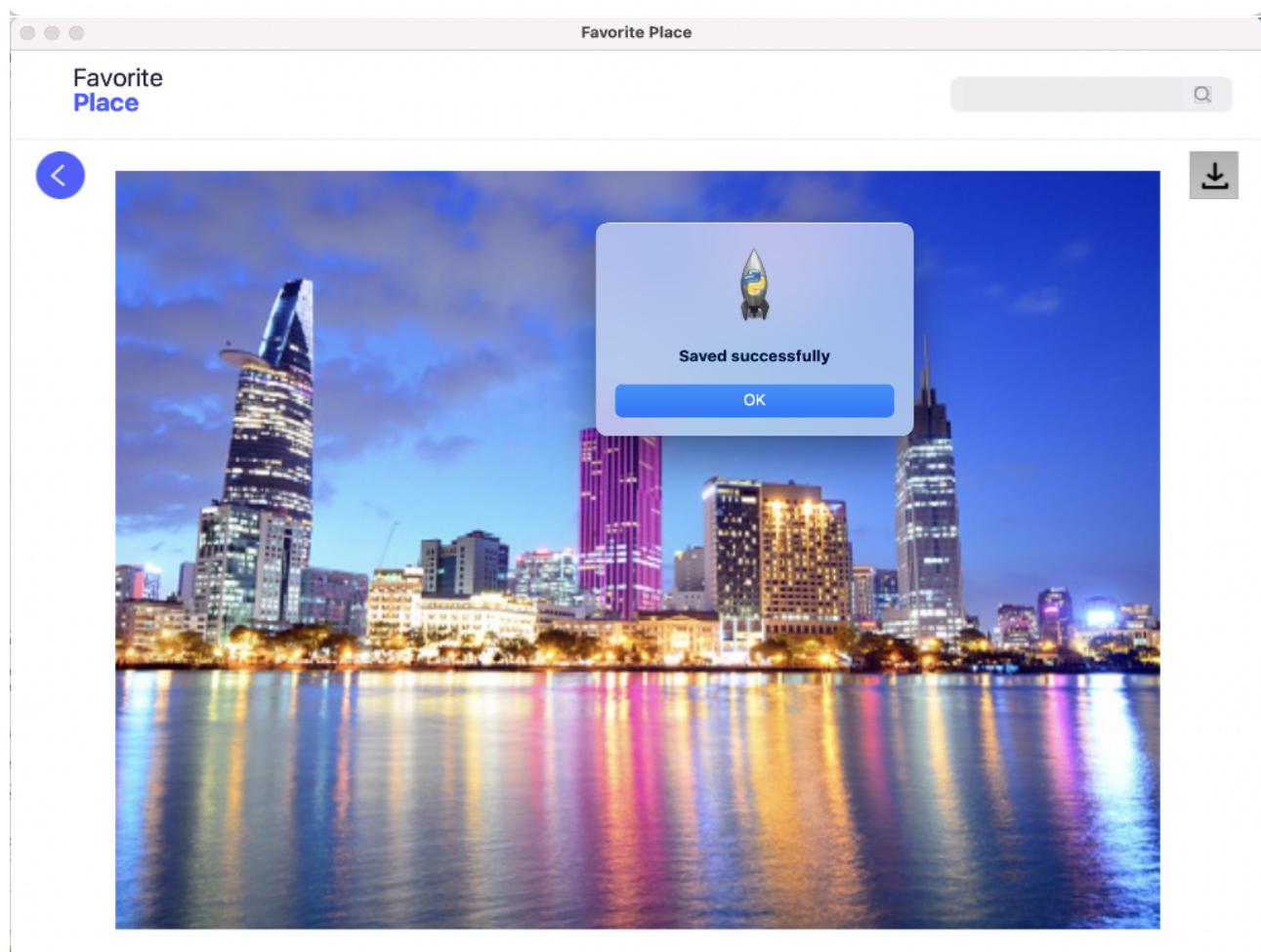
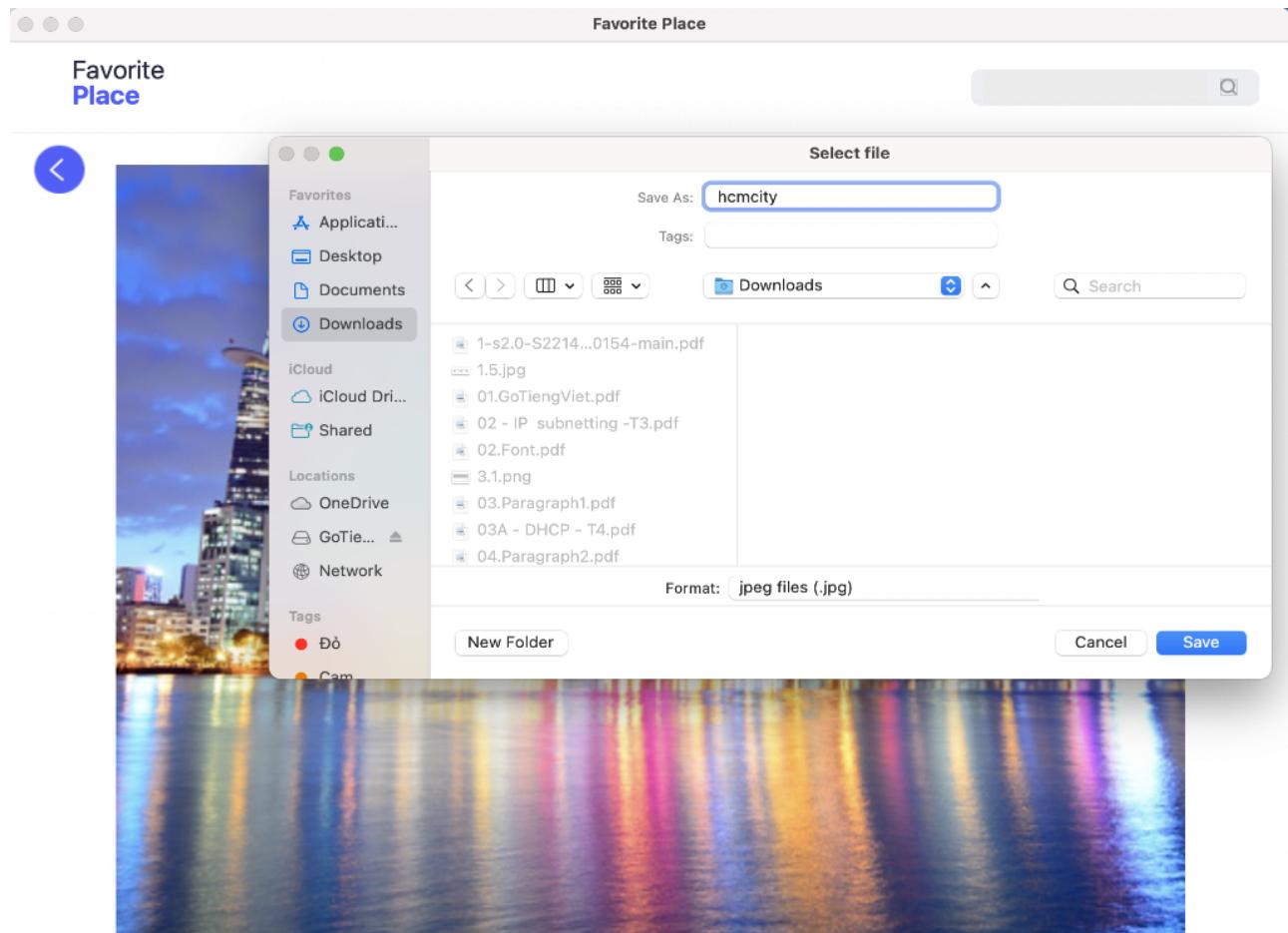
More image:

◀ ▶

Hoặc là ta có thể click vào một ảnh bất kì rồi tải về như sau:



Ta nhấn vào biểu tượng tải về được đóng khung lại ở trên thì tự động ảnh sẽ được tải về và ta chỉ cần nhập vào địa chỉ mà ta muốn lưu là xong. Ví dụ:

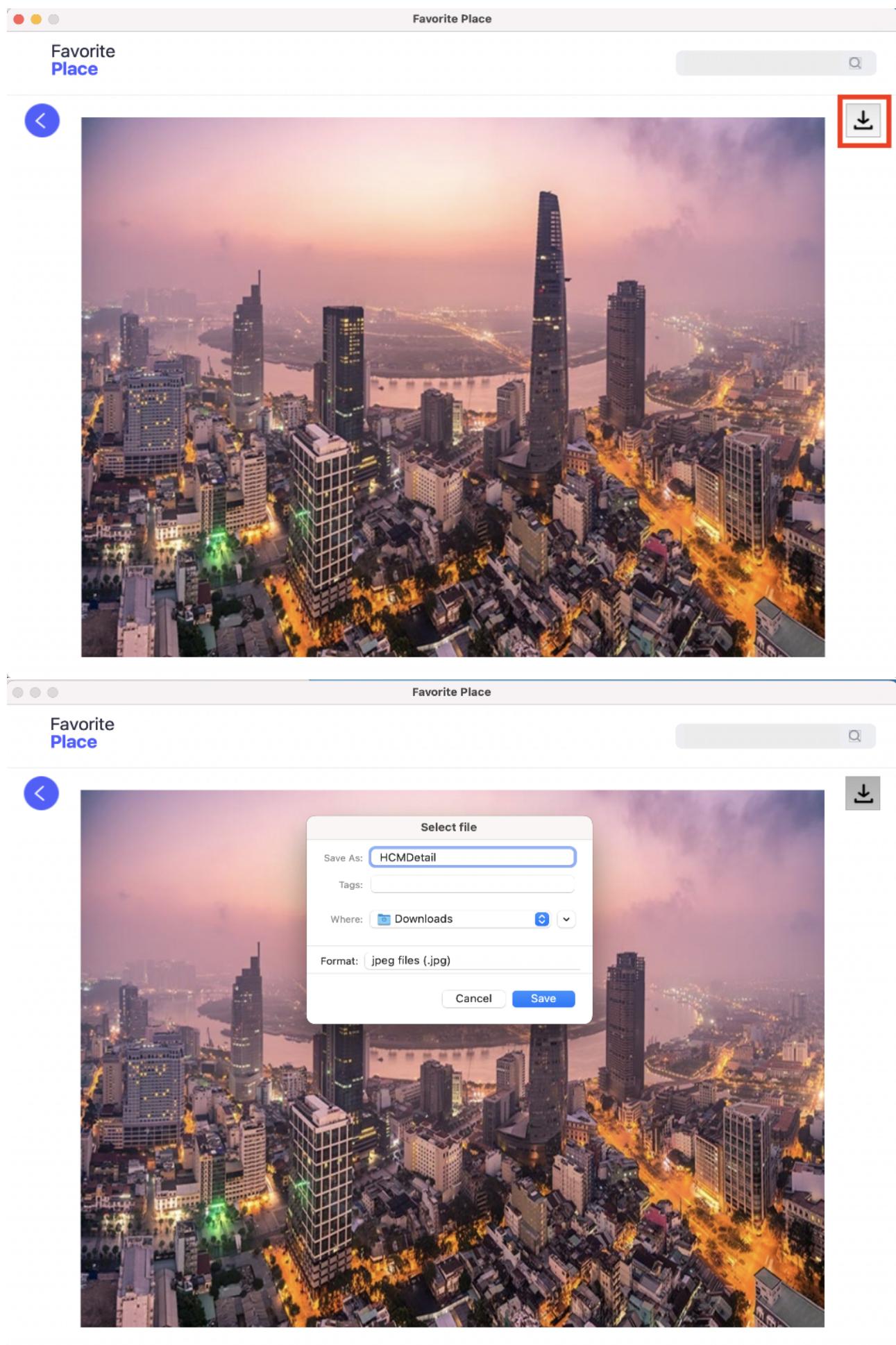


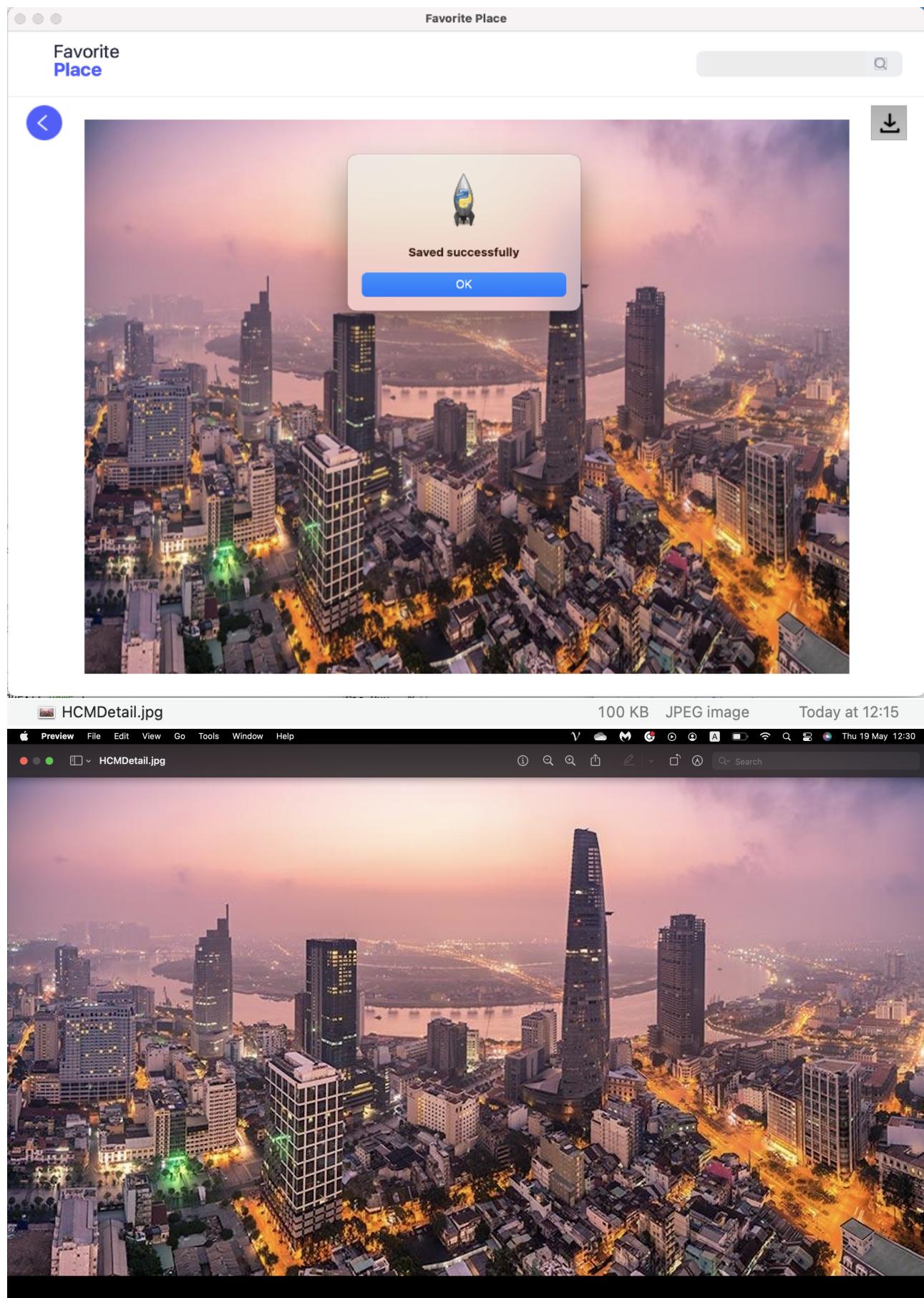
Và khi đó ở mục Downloads thì ảnh có tên **hcmc city.jpg** được tải về.



4.5. Cho phép tải ảnh chi tiết của một địa điểm cụ thể

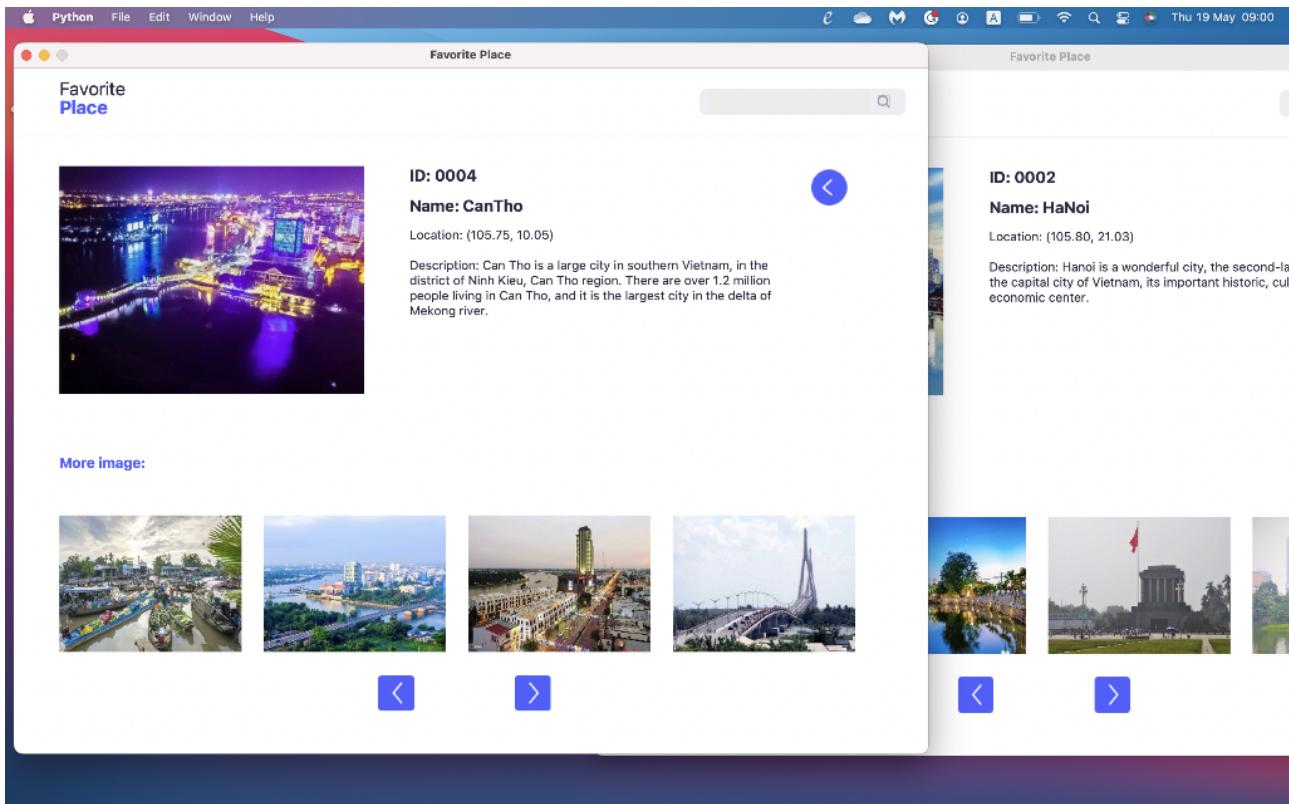
Hoàn toàn tương tự trên, client có thể tải ảnh chi tiết của một địa điểm cụ thể.





4.6. Hỗ trợ nhiều client truy cập đồng thời đến server

Ngoài ra, ứng dụng có thể hỗ trợ nhiều client truy cập đồng thời đến server



5. Tiêu chí đánh giá

5.1. Tự đánh giá

- Thái độ làm việc của từng thành viên trong nhóm chuyên nghiệp, không trì trệ, có thái độ nghiêm túc, tích cực tìm hiểu về các câu hỏi của đồ án.
- Phối hợp công việc tốt, ăn ý.
- Phân công công việc công bằng, rõ ràng, rành mạch.

5.2. Bảng phân công công việc

MSSV	Họ và tên	Nội dung công việc	Mức độ hoàn thành
20120010	Nguyễn Phạm Nhật Huy	Code (làm GUI)	100%
20120014	Vương Gia Huy	Thu thập dữ liệu và đánh báo cáo (latex)	100%
20120021	Hồ Văn Sơn	Code (làm các hàm chức năng)	100%

6. Tài liệu tham khảo

Các nguồn tài liệu tham khảo:

- *Giáo trình mạng máy tính - ĐH KHTN TPHCM*. NXB Khoa học và Kỹ thuật, 2016
- *Tài liệu lý thuyết, thực hành do giảng viên cung cấp trên Moodle môn học*