

Socket programming with Python

Socket là cánh cổng ngăn cách giữa Application Layer với Transport Layer.

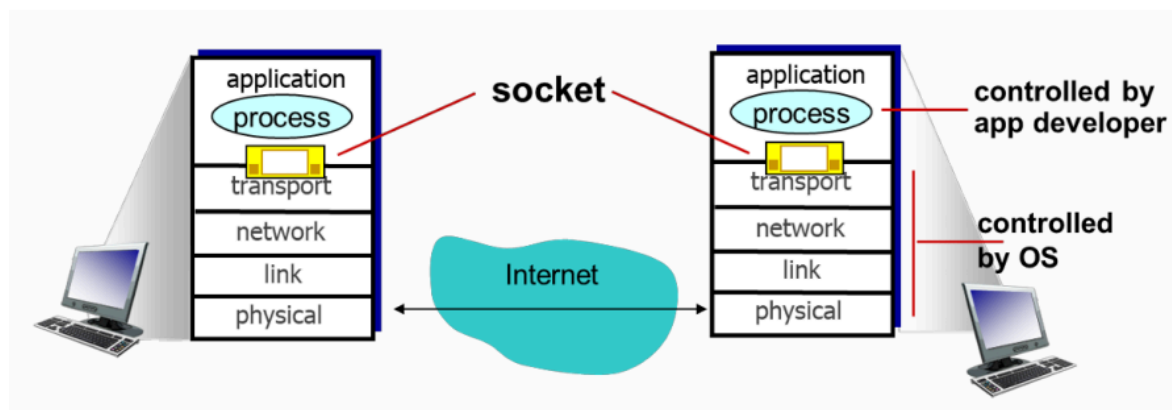


Figure 1: Mô hình socket

Thư viện socket của Python hỗ trợ một số API sau để thiết lập kết nối socket TCP/UDP:

- `socket()`
- `bind()`
- `listen()`
- `accept()`
- `connect()`
- `connect_ex()`
- `send()`
- `recv()`
- `close()`

Thư viện socket của Python hỗ trợ cả TCP socket lẫn UDP socket. Trong project này, nhóm dự định sử dụng TCP socket để hiện thực dự án.

Flow của một TCP socket connection có thể được thể hiện như hình dưới đây:

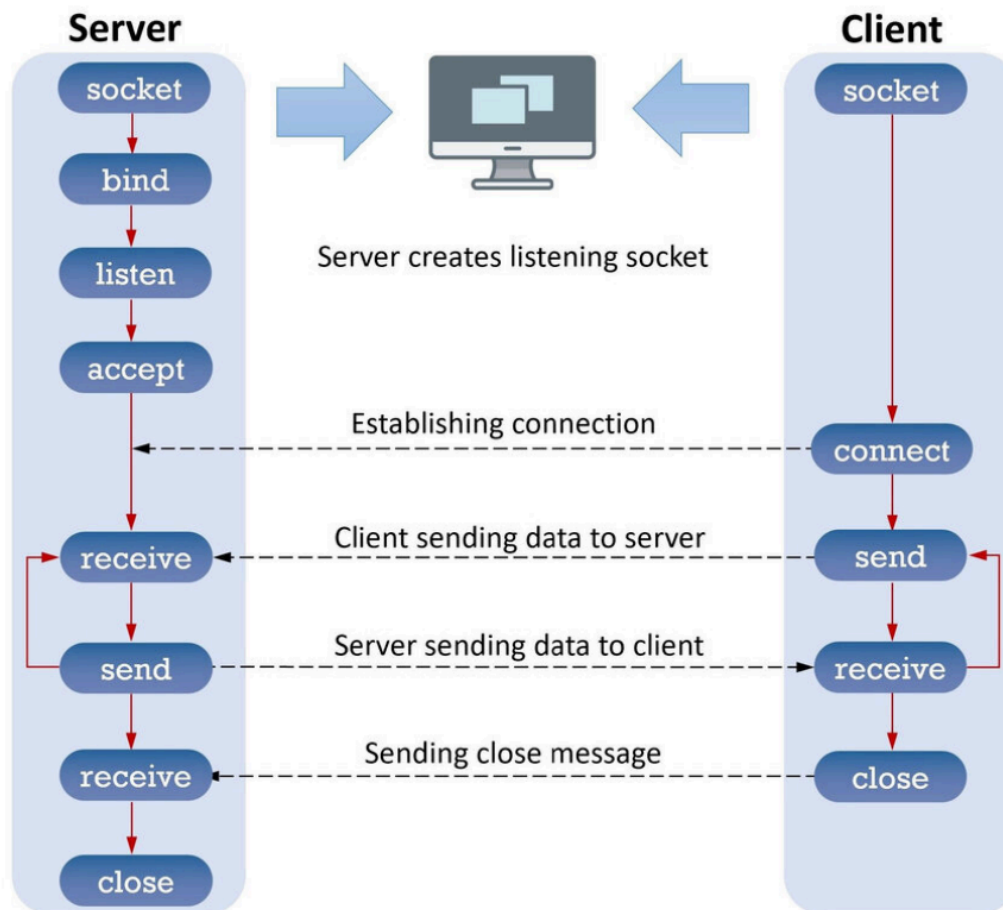


Figure 2: Quy trình một kết nối socket để gửi nhận dữ liệu (Nguồn ResearchGate)

Application design

Architecture design

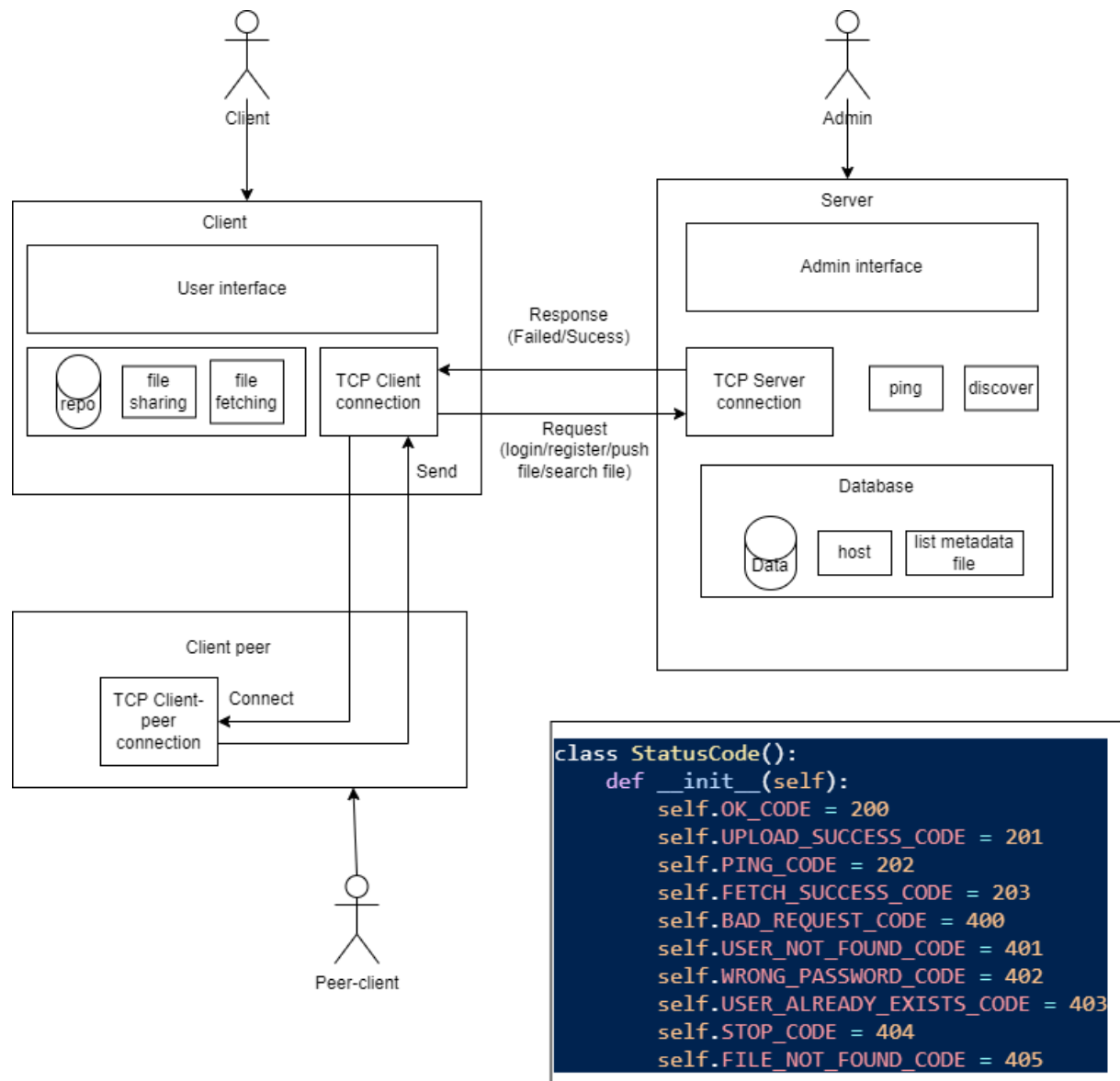


Figure 3: Kiến trúc tổng quan của hệ thống

Flow protocol design

Register

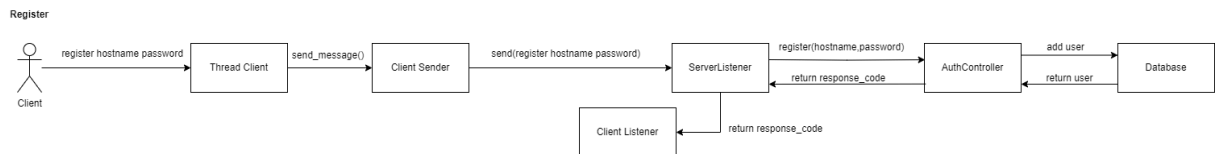


Figure 4: Sơ đồ luồng của chức năng đăng ký

Login

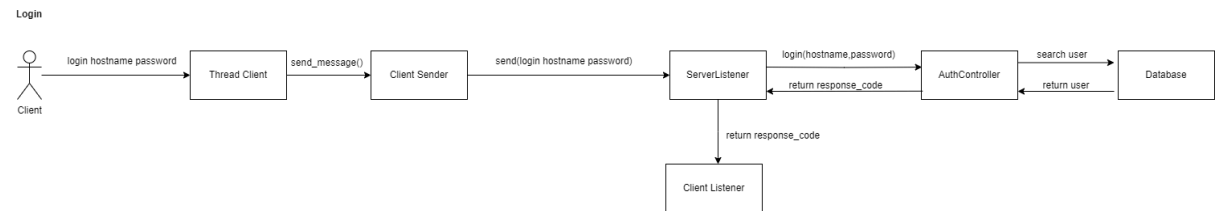


Figure 5: Sơ đồ luồng của chức năng đăng nhập

Ping

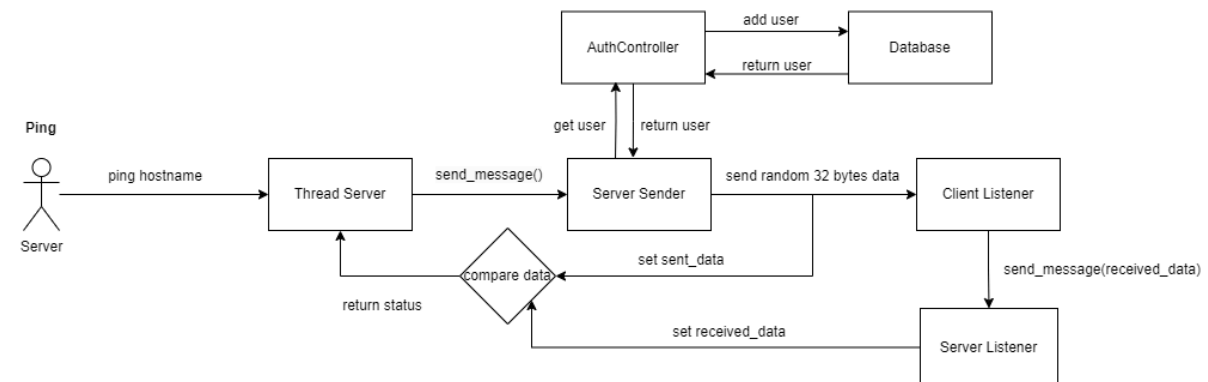


Figure 6: Sơ đồ luồng của chức năng ping

Discover

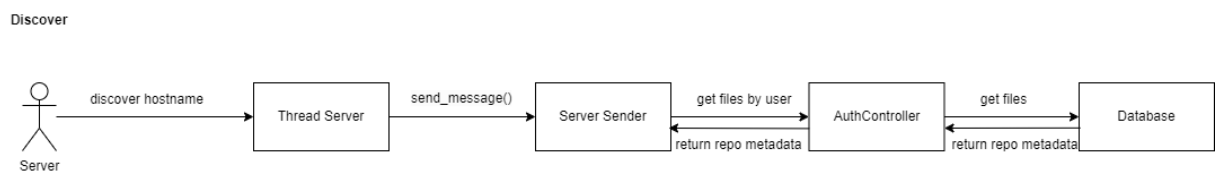


Figure 7: Sơ đồ luồng của chức năng discover

Publish

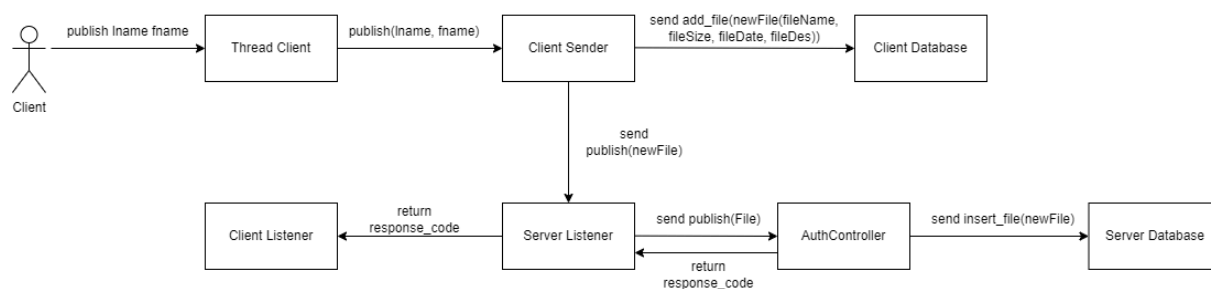


Figure 8: Sơ đồ luồng của chức năng publish

Fetch

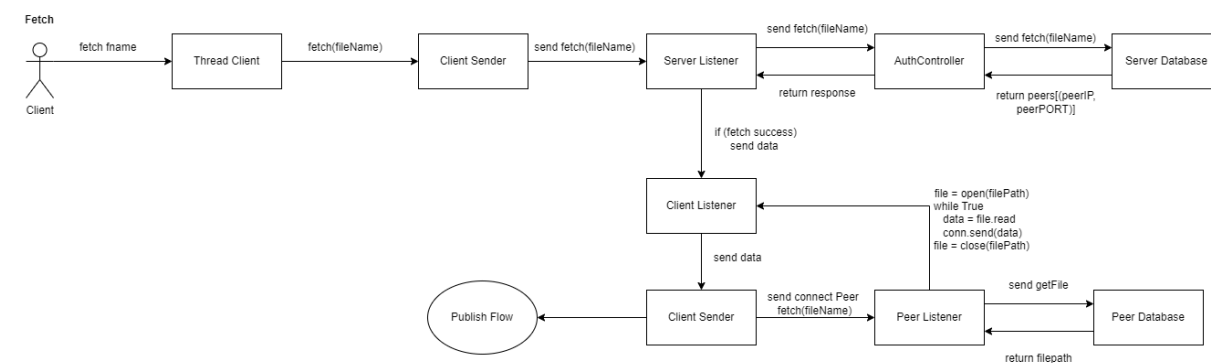


Figure 9: Sơ đồ luồng của chức năng fetch

Activity diagram

Register

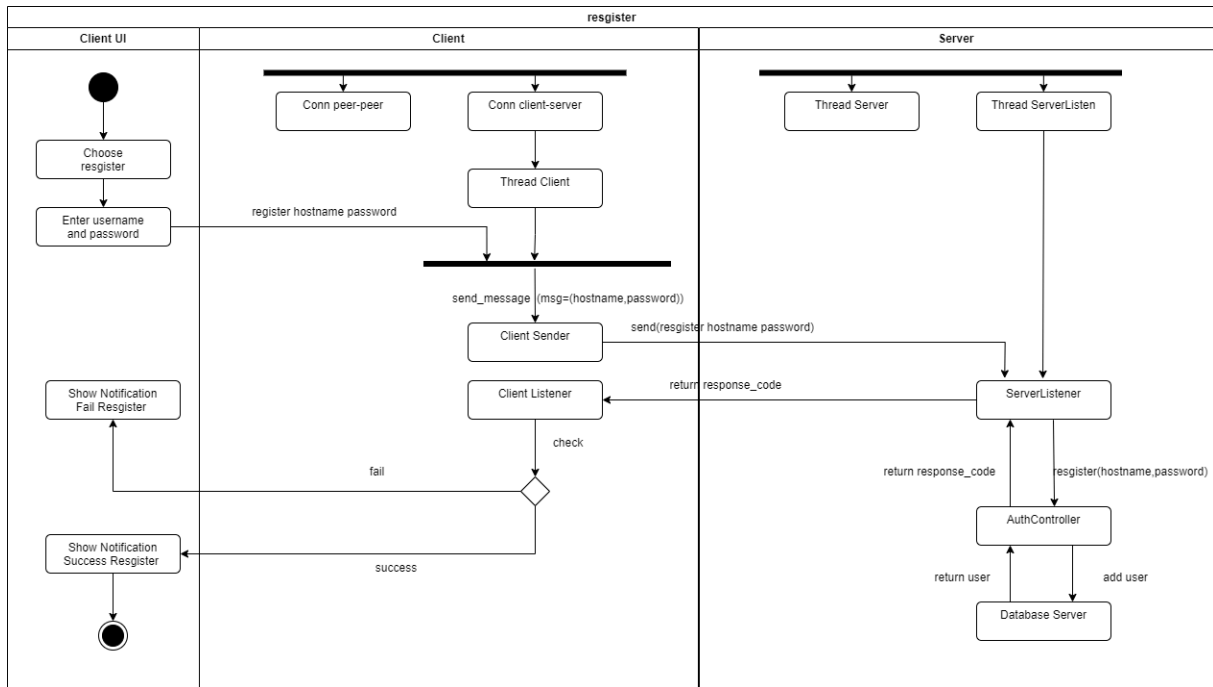


Figure 10: Sơ đồ hoạt động của chức năng đăng ký

Login

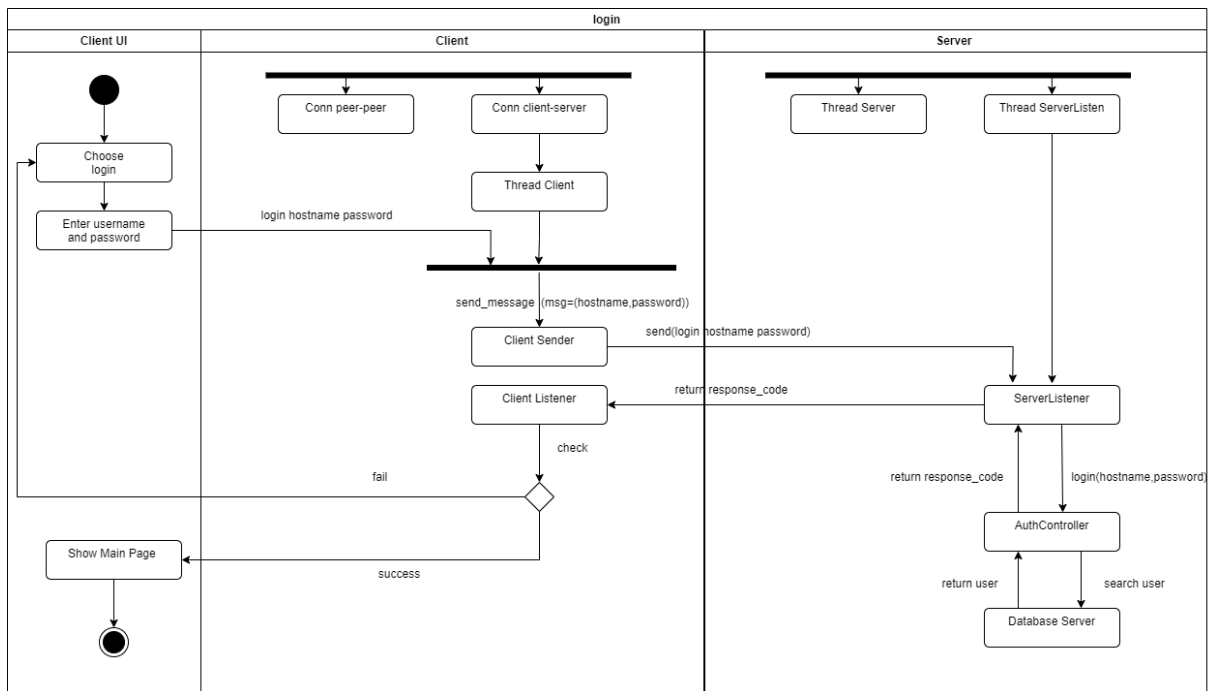


Figure 11: Sơ đồ hoạt động của chức năng đăng nhập

Ping

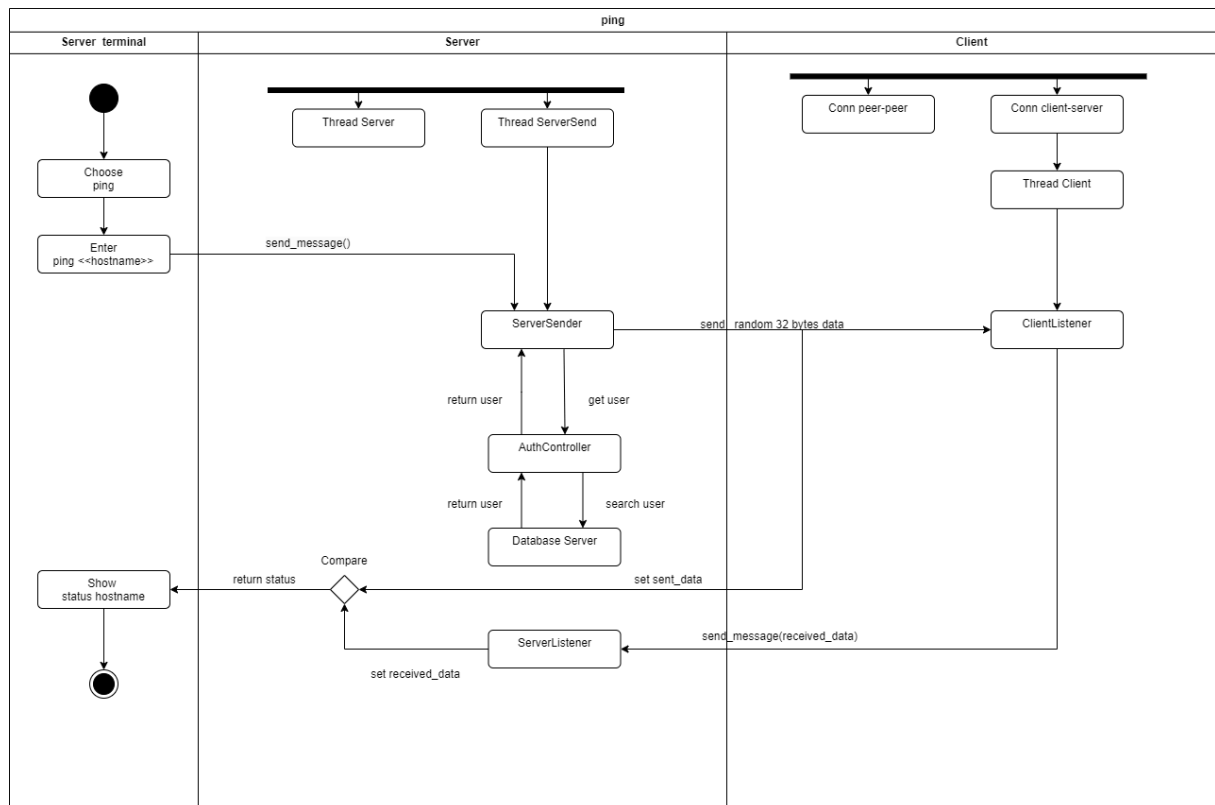


Figure 12: Sơ đồ hoạt động của chức năng ping

Discover

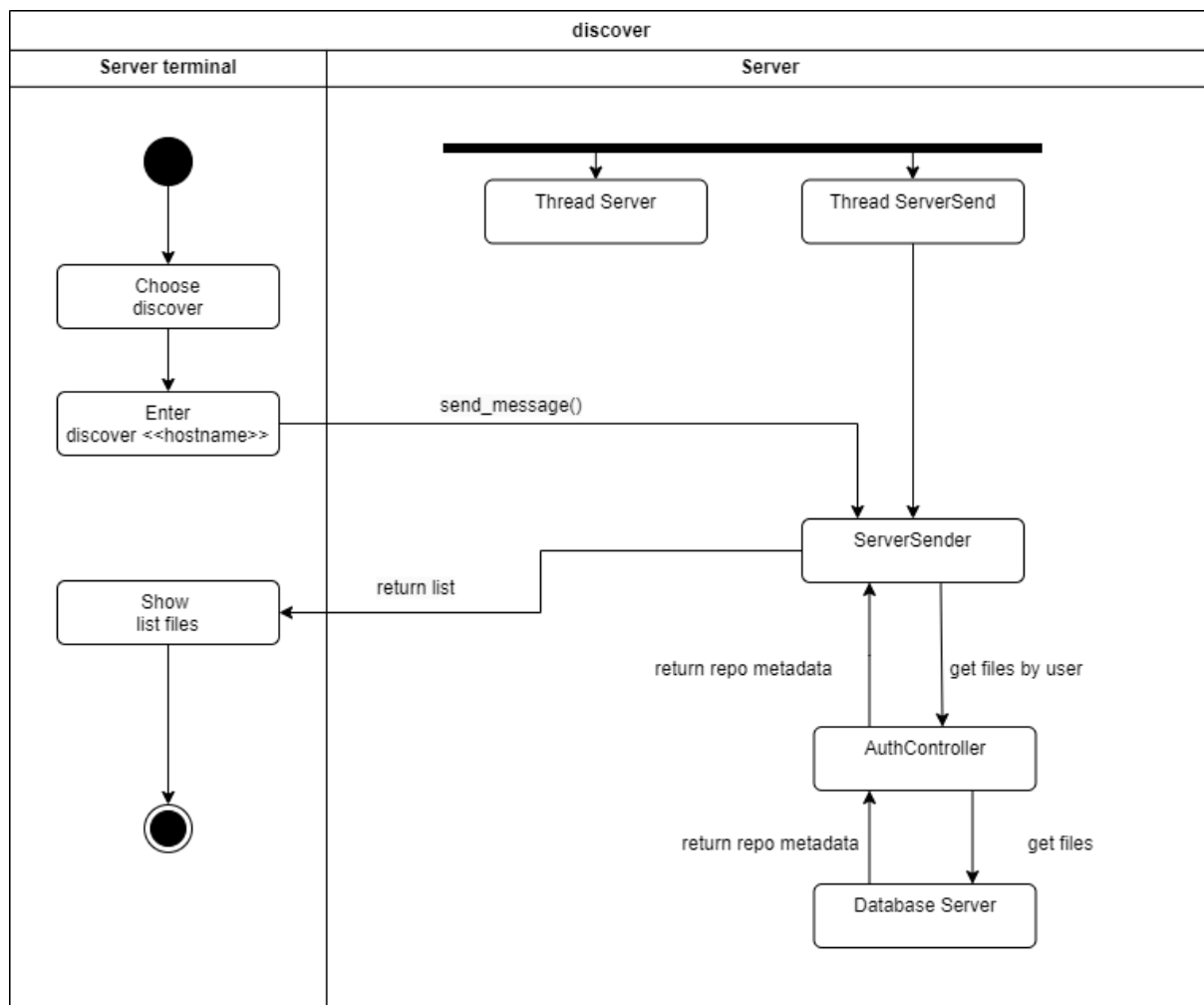


Figure 13: Sơ đồ hoạt động của chức năng discover

Publish

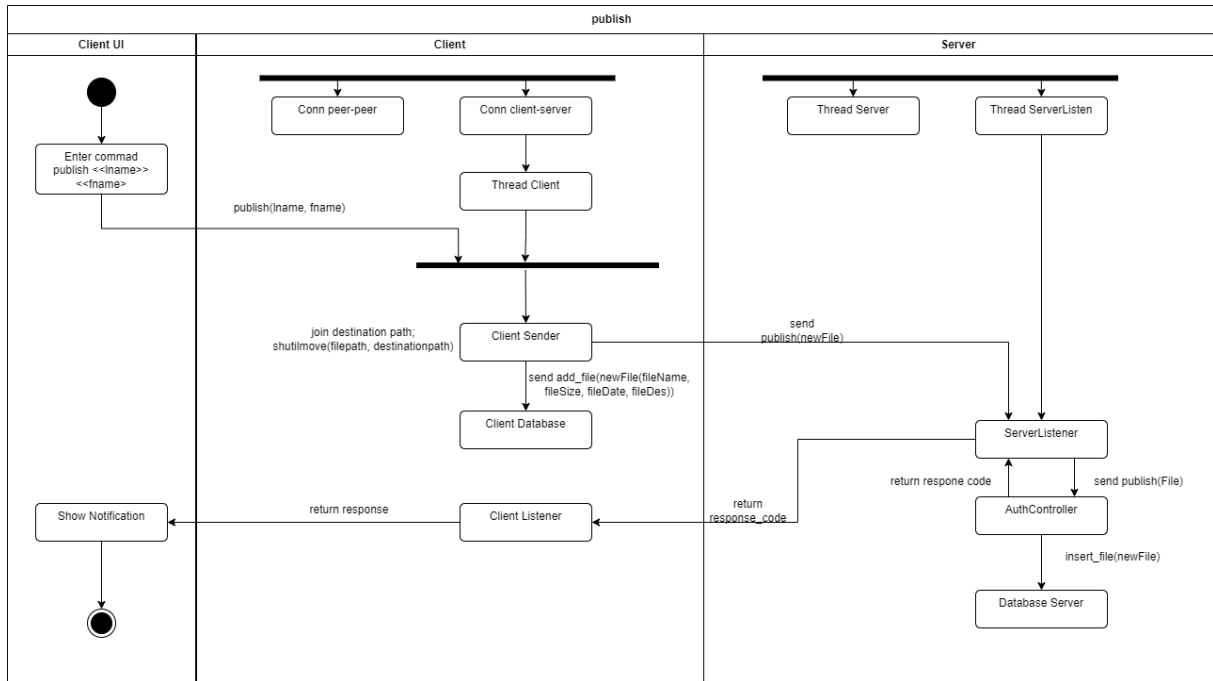


Figure 14: Sơ đồ hoạt động của chức năng publish

Fetch

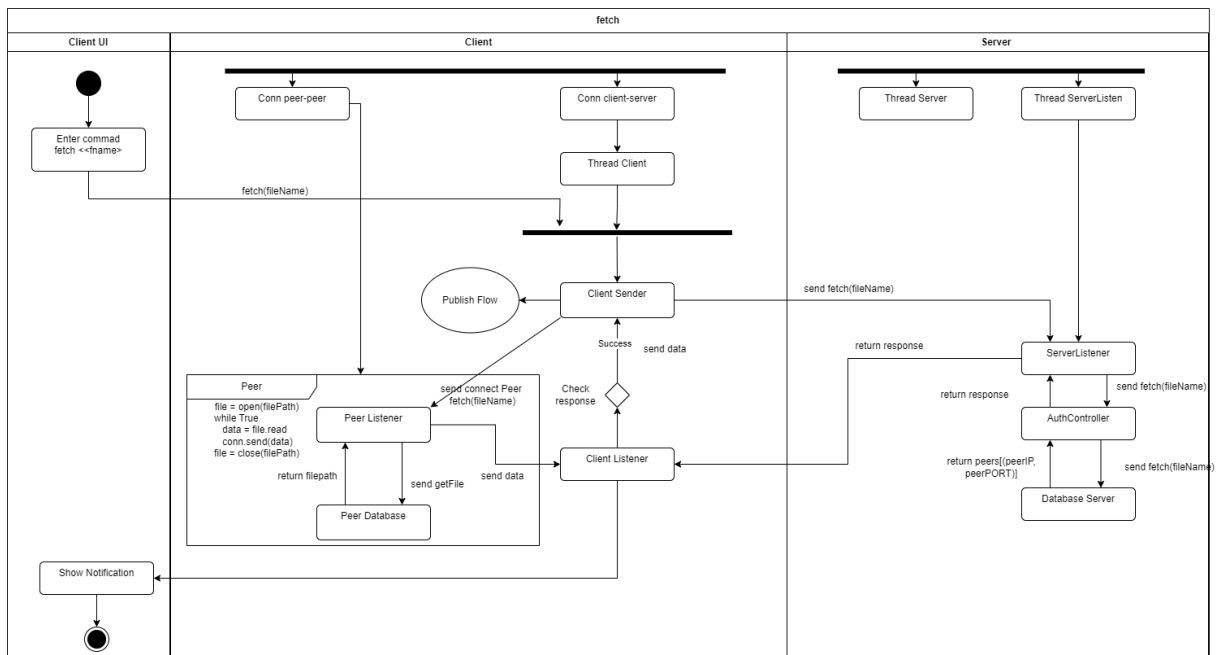


Figure 15: Sơ đồ hoạt động của chức năng fetch

