Service-oriented Architecture INT 3505 1

Homework 1: Socket programming

1. Team member

	Student name	ID
1	Nguyễn Việt Minh Nghĩa	15021358

2. How to run

Run the server first,

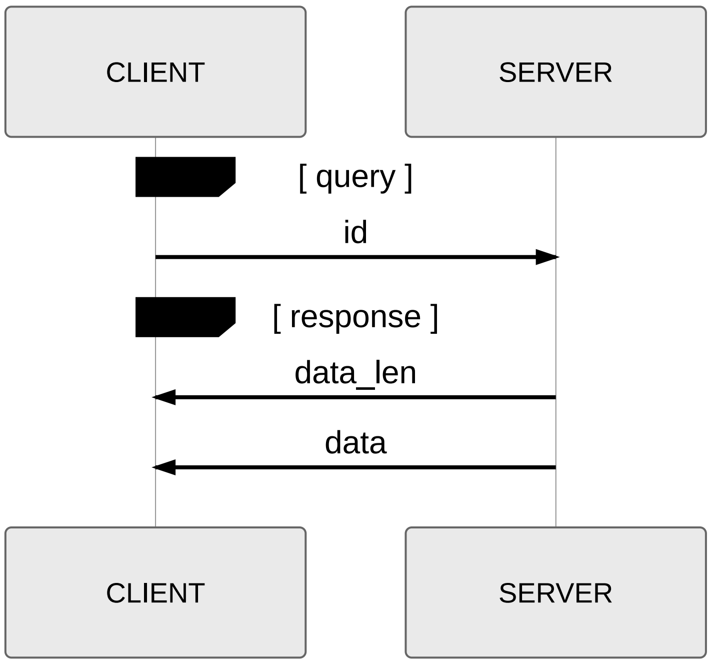
```
python3 server.py
```

then the client

```
python3 client.py
```

Enter the ID to get the information.

3. Description



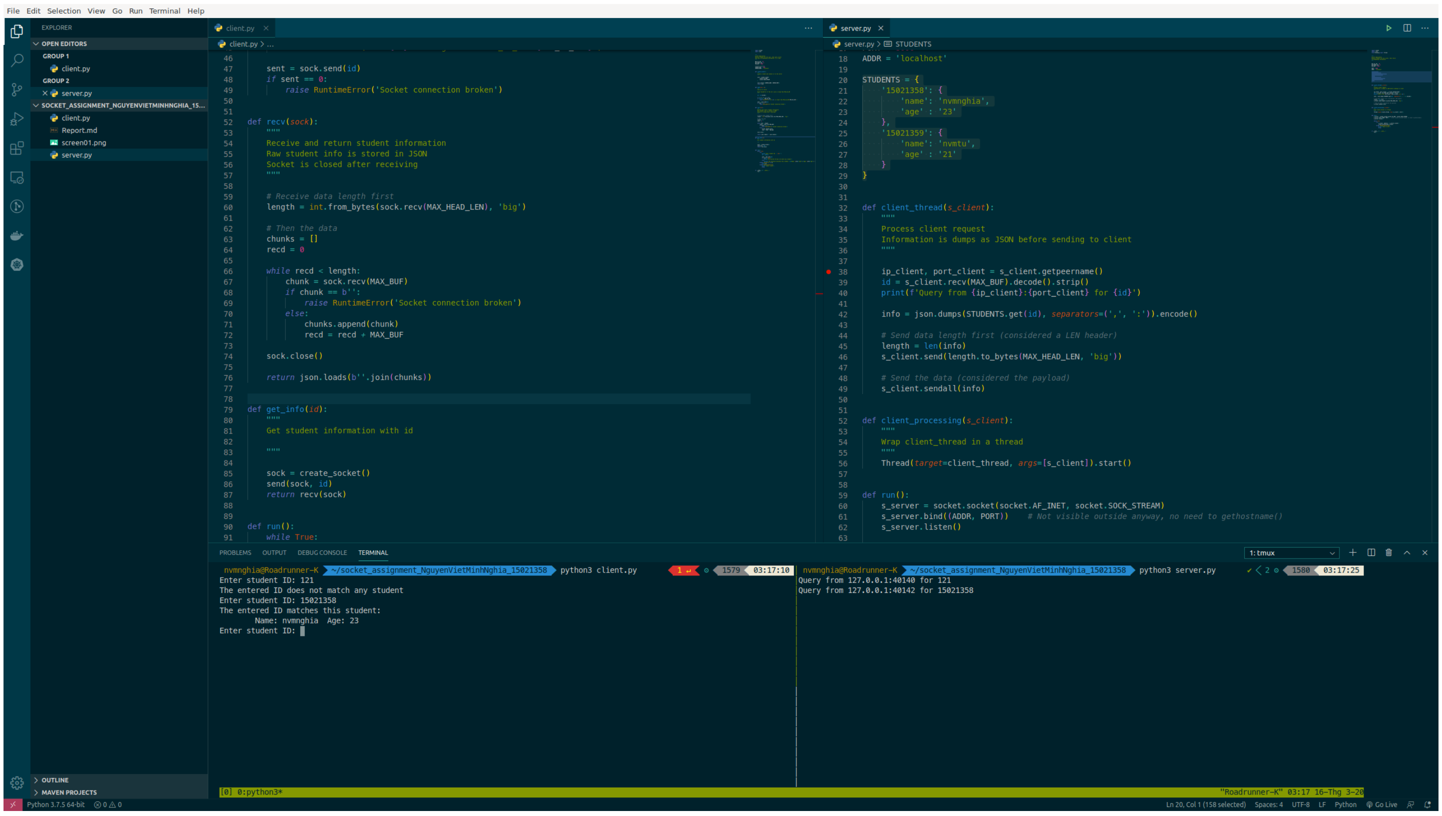
As TCP transmission doesn't have `EOT` , the size of the data must be known beforehand.

First, the client sends the id (`MAX_ID_LEN=64` bytes max) to the server.

Then the server replies with the length of the data (itself `MAX_HEAD_LEN=4` bytes max), simulating a LEN header.

Finally, after obtaining the data size to read, the client receives data from the server, in `MAX_BUF=4096` -byte chunk.

4. Screenshot



```python

    sent = sock.send(id)
    if sent == 0:
        raise RuntimeError('Socket connection broken')


def recv(sock):
    """
    Receive and return student information
    Raw student info is stored in JSON
    Socket is closed after receiving
    """

    # Receive data length first
    length = int.from_bytes(sock.recv(MAX_HEAD_LEN), 'big')

    # Then the data
    chunks = []
    recd = 0

    while recd < length:
        chunk = sock.recv(MAX_BUF)
        if chunk == b'':
            raise RuntimeError('Socket connection broken')
        else:
            chunks.append(chunk)
            recd = recd + MAX_BUF

    sock.close()

    return json.loads(b''.join(chunks))


def get_info(id):
    """
    Get student information with id

    """

    sock = create_socket()
    send(sock, id)
    return recv(sock)


def run():
    while True:
```

```python
ADDR = 'localhost'

STUDENTS = {
    '15021358': {
        'name': 'nvmnghia',
        'age' : '23'
    },
    '15021359': {
        'name': 'nvmtu',
        'age' : '21'
    }
}


def client_thread(s_client):
    """
    Process client request
    Information is dumps as JSON before sending to client
    """

    ip_client, port_client = s_client.getpeername()
    id = s_client.recv(MAX_BUF).decode().strip()
    print(f'Query from {ip_client}:{port_client} for {id}')

    info = json.dumps(STUDENTS.get(id), separators=(',', ':')).encode()

    # Send data length first (considered a LEN header)
    length = len(info)
    s_client.send(length.to_bytes(MAX_HEAD_LEN, 'big'))

    # Send the data (considered the payload)
    s_client.sendall(info)


def client_processing(s_client):
    """
    Wrap client_thread in a thread
    """
    Thread(target=client_thread, args=[s_client]).start()


def run():
    s_server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s_server.bind((ADDR, PORT))    # Not visible outside anyway, no need to gethostname()
    s_server.listen()

```

```
nvmnghia@Roadrunner-K  ~/socket_assignment_NguyenVietMinhNghia_15021358  python3 client.py
Enter student ID: 121
The entered ID does not match any student
Enter student ID: 15021358
The entered ID matches this student:
        Name: nvmnghia  Age: 23
Enter student ID: 
```

```
nvmnghia@Roadrunner-K  ~/socket_assignment_NguyenVietMinhNghia_15021358  python3 server.py
Query from 127.0.0.1:40140 for 121
Query from 127.0.0.1:40142 for 15021358
```