**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN**

**KHOA KHOA HỌC MÁY TÍNH**

****

**BÁO CÁO MÔN LẬP TRÌNH MẠNG**

**ĐỀ TÀI: ỨNG DỤNG CHAT TRỰC TUYẾN**

**Sinh viên thực hiện**: Nguyễn Như Lâm 20IT273

*Đà nẵng….tháng… năm…*

MỤC LỤC

[**CHƯƠNG 1 GIỚI THIỆU ỨNG DỤNG** 3](#_Toc127461616)

[**1.1** **Tổng quan** 3](#_Toc127461617)

[**1.2** **Các chức năng chính** 3](#_Toc127461618)

[**1.2.1.** **Login** 3](#_Toc127461619)

[**1.2.2.** **Chat riêng tư** 3](#_Toc127461620)

[**1.2.3.** **Truyền gửi file** 3](#_Toc127461621)

[**1.3.** **Bộ giao thức TCP/IP:** 3](#_Toc127461622)

[**CHƯƠNG 2 ĐỊNH NGHĨA GIAO THỨC CHO TỪNG CHỨC NĂNG** 5](#_Toc127461623)

[**CHƯƠNG 3: THIẾT KẾ ỨNG DỤNG** 7](#_Toc127461624)

[**3.1.** **Công nghệ sử dụng** 7](#_Toc127461625)

[**3.2.** **Form** 7](#_Toc127461626)

[**3.3.** **Các class chính** 8](#_Toc127461627)

[**3.4.** **Hướng dẫn sử dụng** 9](#_Toc127461628)

[**CHƯƠNG 4 KẾT LUẬN** 9](#_Toc127461629)

[**4.1.** **Kết quả đạt được** 9](#_Toc127461630)

[**4.2.** **Kết quả chưa đạt được** 9](#_Toc127461631)

# **CHƯƠNG 1 GIỚI THIỆU ỨNG DỤNG**

* 1. **Tổng quan**

Phát Biểu Bài Toán

Trong thời đại 4.0, khi khoa học công nghệ ngày càng phát triển thì việc con người có thể kết nối giao tiếp, trò chuyện với nhau cho dù có ở bất kỳ đâu trên thế giới nếu đáp ứng đủ công nghệ , ví dụ như: hình thức CHAT trực tuyến theo thời gian thực, gmail,… Trong đó, hình thức chat trực tuyến là một hình thức được phát triển hơn cả. Hiện nay, mạng Lan/Internet đã có những tiến bộ vượt bậc và ngày càng phổ biến hơn trong đồi sống sinh hoat. Điều này làm cho nhu cầu liên lạc và trao đổi thông tin thông qua mạng Lan/Internet ngày càng lớn hơn. Chính vì vậy, chương trình Chat trên mạng Lan/Internet được xây dựng để đáp ứng phần nào những nhu cầu cấp thiết đó.

Ứng dụng là phần mềm cho phép hai hay nhiều người dùng có thể giao tiếp với nhau với các tính năng chính: login, tạo chat riêng tư, truyền gửi file trong lúc chat.

* 1. **Các chức năng chính**
     1. **Login**

Chức năng này cho phép người dùng đăng nhập vào ứng dụng khi biết IP và port của server.

* + 1. **Chat riêng tư**

Chức năng này cho phép user tạo hội thoại riêng tư và gửi tin trực tiếp tới user khác. Một user có thể nhắn tin trực tiếp với nhiều user trong cùng một thời điểm.

* + 1. **Truyền gửi file**

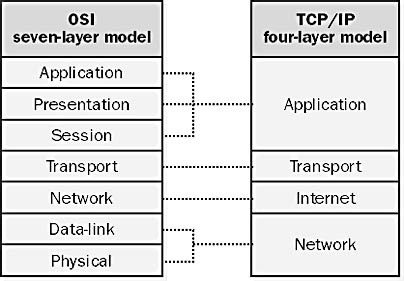
Chức năng này cho phép user gửi và nhận một số file với định dạng và kích thước được quy định trước bởi nhà phát triển trong chat riêng tư với user khác khi được cho phép.

* 1. **Bộ giao thức TCP/IP:**

Bộ giao thức TCP/IP, (tiếng Anh: Internet protocol suite hoặc IP suite hoặc TCP/IP protocol suite - bộ giao thức liên mạng), là một bộ các giao thức truyền thông cài đặt chồng giao thức (protocol stack) mà Internet và hầu hết các mạng máy tính thương mại đang chạy trên đó. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là TCP (Giao thức Điều khiển Giao vận) và IP (Giao thức Liên mạng). Chúng cũng là hai giao thức đầu tiên được định nghĩa.

Như nhiều bộ giao thức khác, bộ giao thức TCP/IP có thể được coi là một tập hợp các tầng, mỗi tầng giải quyết một tập các vấn đề có liên quan đến việc truyền dữ liệu, và cung cấp cho các giao thức tầng cấp trên một dịch vụ được định nghĩa rõ ràng dựa trên việc sử dụng các dịch vụ của các tầng thấp hơn. Về mặt logic, các tầng trên gần với người dùng hơn và làm việc với dữ liệu trừu tượng hơn, chúng dựa vào các giao thức tầng cấp dưới để biến đổi dữ liệu thành các dạng mà cuối cùng có thể được truyền đi một cách vật lý.

Mô hình OSI miêu tả một tập cố định gồm 7 tầng mà một số nhà sản xuất lựa chọn và nó có thể được so sánh tương đối với bộ giao thức TCP/IP. Sự so sánh này có thể gây nhầm lẫn hoặc mang lại sự hiểu biết sâu hơn về bộ giao thức TCP/IP.



**CHƯƠNG 2 ĐỊNH NGHĨA GIAO THỨC CHO TỪNG CHỨC NĂNG**

|  |  |  |
| --- | --- | --- |
| Nội dung | Mục đích | Ghi chú |
| <SESSION\_REQ>clientID</SESSION\_REQ> | Yêu cầu tạo một chat session từ client. clientID là id của người dùng | Sử dụng 1 lần cho mỗi chat session |
| <PEER\_NAME>userName</PEER\_NAME> | Tạo user khi người dùng đăng nhập vào server | userName là tên người dùng |
| numPort | Tạo port cho user | numPort là số hiệu port |
| <SESSION\_KEEP\_ALIVE>  </SESSION\_KEEP\_ALIVE> | Nội dung request do user gửi lên server | User request 10s/lần bao gồm tên user và trạng thái |
| statusUser | Thông báo trạng thái user | statusUser là trạng thái user |
| <SESSION\_DENY /> | Từ chối người dùng kết nối tới server | Từ chối sau khi người dùng login |
| <SESSION\_ACCEPT>  </SESSION\_ACCEPT> | Chấp nhận người dùng kết nối tới server | Chấp nhận sau khi người dùng login |
| <CHAT\_REQ>userName</CHAT\_REQ> | Yêu cầu tạo chat riêng tư từ một user với user khác | userName là tên user tạo yêu cầu chat riêng tư |
| stringIP | Định danh IP cho user | stringIP là IP của user |
| <CHAT\_DENY /> | Từ chối tạo chat session | Từ chối sau khi có yêu cầu tạo chat riêng tư |
| <CHAT\_ACCEPT /> | Chấp nhận tạo chat session | Chấp nhận sau khi có yêu cầu tạo chat riêng tư |
| <CHAT\_MSG> Message</CHAT\_MSG> | Gửi nội dung chat | Message được gửi khi user đồng ý tạo chat riêng tư |
| peerData | Xác định peer của user do server trả về | peerData là peer của user gồm name, port, IP |
| <FILE\_REQ> fileName</FILE\_REQ> | Yêu cầu gửi file | fileName là tên file muốn gửi |
| <FILE\_REQ\_NOACK /> | Từ chối yêu cầu gửi file | Sử dụng sau khi có yêu cầu gửi file |
| <FILE\_REQ\_ACK></FILE\_REQ\_ACK> | Chấp nhận yêu cầu gửi file | Sử dụng sau khi có yêu cầu gửi file |
| <FILE\_DATA\_BEGIN /> | Bắt đầu gửi file | Quá trình gửi file sau khi người gửi và nhận đồng ý |
| <FILE\_DATA>fileData</FILE\_DATA> | Nội dung file cần chuyển |  |
| <FILE\_DATA\_END /> | Kết thúc quá trình gửi file | Yêu cầu bởi người gửi file |
| <CHAT\_CLOSE /> | Kết thúc chat riêng tư | Sử dụng sau khi tạo chat riêng tư |
| MAX\_MSG\_SIZE | 102400 | Kích thước tối đa của một message(file) trong chat |
| SERVER\_ONLINE | RUNNING | User đang online |

# 

# **CHƯƠNG 3: THIẾT KẾ ỨNG DỤNG**

* 1. **Công nghệ sử dụng**
* TCP Socket: Một kĩ thuật dùng để hỗ trợ lập trình các ứng dụng giao tiếp qua mạng. TCP Socket sử dụng Stream để thực hiện quá trình truyền dữ liệu của hai máy tính đã thiết lập kết nối.
* Java Swing: Là một phần của Java Foundation Classes (JFC) được sử dụng để tạo các ứng dụng Window-Based.
  1. **Form**

Ứng dụng gồm 4 form chính:

* Server form: Form quản lí server, hiển thị thông tin server, các request user gửi lên.
* Login form: Form đăng nhập vào server. Để kết nối được, user phải cung cấp username chưa có ai đăng kí trước đó với server.
* Main form: Form để quản lí danh sách những người dùng khác đang online cùng với user hiện tại. User có thể chọn bất kì ai đang online để bắt đầu một cuộc trò chuyện.
* Chat form: Khi hai user đã chấp nhận trò chuyện với nhau thì form này sẽ xuất hiện. Hiển thị thông tin cuộc trò chuyện giữa hai người.
  1. **Các class chính**

Class cho server:

* ServerFrame.java: sử dụng để thao tác với server như bật, tắt server. Ngoài ra còn hiển thị các thông tin server như: IP, Port, số lượng người kết nối, các protocol mà client gửi lên.
* ServerCore.java: Đây là class để quản lí server, gồm đầy đủ thông tin, chức năng chính của server:
  + Tạo một SocketServer.
  + Quản lí danh sách các user đang online
  + Tạo một thread là WaitForConnect để xử lý các request kết nối từ user. Có thể là yêu cầu dăng nhập, thoát ứng dụng, lấy thông tin user đang online. Nếu đăng nhập thành công, trả về cho client danh sách user đang online, nếu không sẽ trả về một protocol từ chối đăng nhập.

Class cho Client:

* MainFrame.java: GUI class để hiển thị danh sách user khác đang online, bắt sự kiện gửi yêu cầu chat của user.
* ChatFrame.java: Xử lí việc chat, gửi nhận File giữa hai user. Hiển thị nội dung chat giữa hai user.
* ClientServer.java: Tạo và quản lí ServerSocket của mỗi user. Xử lí request chat gửi đến từ user khác.
* Client.java: Quản lí thông tin hiện tại của user. Nó sẽ tạo một thread để liên tục gửi request tới server để cập nhật danh sách user. Ngoài ra nó còn xử lí request chat tới user khác.

Các class khác:

* DataFile.java: Tạo một đối tượng File để gửi đi.
* Peer.java: Tạo một đối tượng Peer có các thuộc tính: IP, port, name.
* Tags.java: Định nghĩa các protocol được sử dụng trong ứng dụng.
* Encode.java: Định nghĩa các phương thức user đùng để gửi request lên server.
* Dedode.java: Giải mã các phương thức để lấy thông tin user, port hay ip...
  1. **Hướng dẫn sử dụng**

B1: Chạy file ServerFrame.java trong folder server để khởi chạy Server

B2: Chạy file LoginFrame trong folder login để khởi chạy giao diện đăng nhập dành cho người dùng

B3: Chọn một người dùng trong danh sách người dùng đang online để trò chuyện cùng nhau

# **CHƯƠNG 4 KẾT LUẬN**

**4.1. Kết quả đạt được**

Ứng dụng được xây dựng dựa trên mô hình kết hợp giữa client-server cho việc quản lí các user và P2P cho việc trò chuyện giữa hai user với nhau.

Ứng dụng có các tính năng đơn giản như: chat giữa hai user, một lúc đồng thời chat với nhiều user, gửi File trong quá trình chat.

**4.2. Kết quả chưa đạt được**

Mã nguồn còn chưa tối ưu cho ứng dụng.

Ứng dụng còn có thể thêm các tính năng như: chat nhóm, gọi video...