**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙣🙢🕮🙠🙡**



**ĐỒ ÁN MÔN HỌC CÔNG NGHỆ JAVA**

***Đề tài:***

**XÂY DỰNG ỨNG DỤNG CHAT SỬ DỤNG SOCKET THEO MÔ HÌNH CLIENT – SERVER**

**Thành phố Hồ Chí Minh, tháng 6 năm 2024**

GVHD: Huỳnh Hữu Nghĩa

Sinh viên thực hiện:

1. 2001215969 - Lê Hoài Nam

2. 2001210812 - Nguyễn Tấn Thịnh

3. 2001202010 - Dương Thiên Anh

4. 2001215670 – Huỳnh Trương Bảo Duy

**BẢNG PHÂN CÔNG CÔNG VIỆC**

**LỜI CAM ĐOAN**

Nhóm em xin cam đoan rằng đề tài "Xây dựng ứng dụng chat sử dụng socket theo mô hình client-server" là công trình nghiên cứu của riêng nhóm, được thực hiện dưới sự hướng dẫn của Thầy Huỳnh Hữu Nghĩa. Các kết quả nghiên cứu và các nội dung trình bày trong đề tài này là trung thực và không sao chép từ bất kỳ nguồn nào khác ngoài những tài liệu đã được trích dẫn trong phần Tài liệu tham khảo.

Nhóm em cũng xin cam đoan rằng mọi sự trợ giúp cho việc thực hiện đề tài này đã được cảm ơn một cách thích đáng. Nếu có bất kỳ sự gian lận nào được phát hiện trong đề tài này, nhóm xin hoàn toàn chịu trách nhiệm.

**Sinh viên thực hiện**

(Ký và ghi rõ họ tên)

Lê Hoài Nam

Nguyễn Tấn Thịnh

Dương Thiên Anh

Huỳnh Trương Bảo Duy

**LỜI CẢM ƠN**

Trong suốt quá trình thực hiện đề tài, chúng em xin bày tỏ lòng biết ơn sâu sắc đến Thầy Huỳnh Hữu Nghĩa, người đã tận tình hướng dẫn, góp ý và giúp đỡ chúng em trong suốt quá trình nghiên cứu và hoàn thiện đề tài này. Sự chỉ dẫn và kiến thức quý báu của Thầy đã giúp nhóm vượt qua những khó khăn và thử thách trong quá trình thực hiện đề tài.

**Nhóm**

**MỞ ĐẦU**

Trong thời đại số hóa hiện nay, ứng dụng chat không chỉ là phương tiện giao tiếp thông thường mà còn trở thành một phần không thể thiếu trong cuộc sống hàng ngày và trong môi trường làm việc. Việc xây dựng một ứng dụng chat sử dụng công nghệ socket theo mô hình client-server không chỉ là một dự án thú vị mà còn là một cơ hội để tìm hiểu sâu hơn về lập trình mạng và phát triển phần mềm.

Đặc biệt, việc áp dụng ngôn ngữ lập trình Java trong xây dựng ứng dụng này không chỉ mang lại tính linh hoạt và khả năng mở rộng cao mà còn giúp sinh viên hiểu rõ hơn về cách thức hoạt động của các công nghệ và mô hình được sử dụng.

Ở phần sau, chúng ta sẽ xem xét chi tiết hơn về lý do lựa chọn đề tài này, mục tiêu nghiên cứu, cũng như cơ sở lý thuyết và công nghệ cần thiết để thực hiện dự án một cách thành công. Đồng thời, cũng sẽ trình bày một cái nhìn tổng quan về cấu trúc và nội dung của đề tài.

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc168353989)

[1.1. Giới thiệu đề tài 1](#_Toc168353990)

[1.2. Mục tiêu đề tài 1](#_Toc168353991)

[1.3. Tổng quan lý thuyết 2](#_Toc168353992)

[1.3.1. Socket 2](#_Toc168353993)

[a. Khái niệm Socket 2](#_Toc168353994)

[b. Phân loại Socket 2](#_Toc168353995)

[c. Hoạt động của Socket 2](#_Toc168353996)

[1.3.2. Mô hình Client – Server 2](#_Toc168353997)

[a. Khái niệm 2](#_Toc168353998)

[b. Đặc điểm của mô hình Client-Server 2](#_Toc168353999)

[c. Ưu và nhược điểm của mô hình Client-Server 3](#_Toc168354000)

[1.4. Công cụ và thiết bị sử dụng 3](#_Toc168354001)

[CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU 4](#_Toc168354002)

[2.1. Giới thiệu 4](#_Toc168354003)

[2.2. Thiết kế cơ sở dữ liệu 4](#_Toc168354004)

[2.2.1. Sơ đồ cơ sở dữ liệu 4](#_Toc168354005)

[2.2.2. Mô tả 4](#_Toc168354006)

[a. Bảng user 4](#_Toc168354007)

[b. Bảng user\_account 4](#_Toc168354008)

[c. Mối quan hệ giữa các bảng 5](#_Toc168354009)

[CHƯƠNG 3: THIẾT KẾ GIAO DIỆN 6](#_Toc168354010)

[3.1. Giao diện khởi động 6](#_Toc168354011)

[3.2. Giao diện đăng nhập 7](#_Toc168354012)

[3.3. Giao diện đăng ký 8](#_Toc168354013)

[3.4. Giao diện chat 9](#_Toc168354014)

[TỔNG KẾT 12](#_Toc168354015)

[TÀI LIỆU THAM KHẢO 13](#_Toc168354016)

**DANH MỤC HÌNH ẢNH**

[Hình 1. Sơ đồ cơ sở dữ liệu ứng dụng 4](#_Toc168354446)

[Hình 2. Sever đang được khởi động 6](#_Toc168354447)

[Hình 3. Sever hiển thị 1 client kết nối khi mở ứng dụng 7](#_Toc168354448)

[Hình 4. Giao diện đăng nhập 7](#_Toc168354449)

[Hình 5. Giao diện đăng ký 8](#_Toc168354450)

[Hình 6. Thông tin user được gửi về sever sau khi đăng ký 8](#_Toc168354451)

[Hình 7. Giao diện sau khi đăng nhập 9](#_Toc168354452)

[Hình 8. Giao diện khi chọn 1 người chat 10](#_Toc168354453)

[Hình 9. Gửi và nhận tin nhắn văn bản 10](#_Toc168354454)

[Hình 10. Tính năng gửi emoji 11](#_Toc168354455)

# CHƯƠNG 1: TỔNG QUAN

## 1.1. Giới thiệu đề tài

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, giao tiếp qua mạng đã trở thành một nhu cầu thiết yếu trong cuộc sống hàng ngày và công việc. Các ứng dụng chat đã trở thành công cụ phổ biến, giúp con người kết nối với nhau dễ dàng và nhanh chóng hơn. Từ những cuộc trò chuyện cá nhân đến các cuộc họp nhóm, từ làm việc từ xa đến hỗ trợ khách hàng, các ứng dụng chat như WhatsApp, Facebook Messenger, Slack, và Discord đều đóng vai trò quan trọng trong việc truyền tải thông tin.

Việc xây dựng một ứng dụng chat không chỉ giúp đáp ứng nhu cầu giao tiếp của người dùng mà còn mang lại nhiều lợi ích trong việc học tập và nghiên cứu. Đối với sinh viên ngành Công nghệ thông tin, phát triển một ứng dụng chat là cơ hội để nắm vững các khái niệm về mạng máy tính, lập trình mạng và các mô hình phần mềm. Đặc biệt, việc sử dụng công nghệ socket trong lập trình mạng giúp sinh viên hiểu rõ hơn về cách thức giao tiếp giữa các thiết bị trong mạng.

## 1.2. Mục tiêu đề tài

Java và công nghệ socket theo mô hình client-server. Ứng dụng này phải đáp ứng các yêu cầu cơ bản như:

* Cho phép nhiều người dùng kết nối và trao đổi tin nhắn với nhau thông qua một máy chủ trung gian.
* Đảm bảo việc truyền tải tin nhắn một cách nhanh chóng và chính xác.
* Cung cấp giao diện người dùng thân thiện và dễ sử dụng.
* Đảm bảo an toàn và bảo mật thông tin trong quá trình trao đổi dữ liệu.

Bên cạnh các yêu cầu cơ bản, đề tài cũng hướng đến việc mở rộng và phát triển thêm các tính năng nâng cao như lưu trữ lịch sử trò chuyện, hỗ trợ gửi tệp tin, và tạo các phòng trò chuyện riêng biệt.

## 1.3. Tổng quan lý thuyết

### 1.3.1. Socket

#### a. Khái niệm Socket

Socket là một điểm cuối của kênh giao tiếp hai chiều giữa hai chương trình chạy trên mạng. Đó là một giao diện lập trình cho phép các chương trình ứng dụng giao tiếp qua mạng máy tính. Socket thường được sử dụng để thiết lập các kết nối mạng, gửi và nhận dữ liệu qua mạng.

#### b. Phân loại Socket

* **Socket TCP (Transmission Control Protocol):** Là loại socket phổ biến nhất, cung cấp một kết nối đáng tin cậy, định hướng kết nối và đảm bảo thứ tự của dữ liệu. TCP đảm bảo rằng tất cả các gói tin được gửi đi sẽ đến đích mà không bị mất mát.
* **Socket UDP (User Datagram Protocol):** Là loại socket không định hướng kết nối, không đảm bảo việc truyền tải đáng tin cậy và thứ tự của dữ liệu. UDP thường được sử dụng trong các ứng dụng yêu cầu tốc độ cao nhưng không cần đảm bảo toàn vẹn dữ liệu, như truyền thông tin âm thanh hoặc video trực tiếp.

#### c. Hoạt động của Socket

Socket hoạt động dựa trên mô hình client-server. Trong mô hình này, client tạo một yêu cầu kết nối đến server, server chấp nhận kết nối và hai bên bắt đầu trao đổi dữ liệu.

### 1.3.2. Mô hình Client – Server

#### a. Khái niệm

Mô hình client-server là một mô hình mạng trong đó client gửi yêu cầu dịch vụ đến server, server xử lý và gửi lại phản hồi cho client. Client thường là các thiết bị đầu cuối (máy tính, điện thoại, tablet) sử dụng dịch vụ, còn server là máy chủ cung cấp dịch vụ.

#### b. Đặc điểm của mô hình Client-Server

* **Phân cấp:** Server chịu trách nhiệm cung cấp dịch vụ và quản lý các kết nối từ client. Client gửi yêu cầu và nhận phản hồi từ server.
* **Tập trung hóa:** Server quản lý tài nguyên và dữ liệu tập trung, giúp dễ dàng bảo trì và quản lý.
* **Khả năng mở rộng:** Hệ thống có thể mở rộng bằng cách tăng số lượng server hoặc cải thiện hiệu năng của server.

#### c. Ưu và nhược điểm của mô hình Client-Server

* **Ưu điểm:** Quản lý tài nguyên tập trung, dễ dàng bảo trì và nâng cấp, bảo mật tốt hơn.
* **Nhược điểm:** Nếu server gặp sự cố, toàn bộ hệ thống có thể bị ảnh hưởng. Cần đầu tư vào phần cứng và phần mềm cho server.

## 1.4. Công cụ và thiết bị sử dụng

Ngôn ngữ lập trình Java.

Apache NetBeans IDE.

MySQL Workbench.

# CHƯƠNG 2: THIẾT KẾ CƠ SỞ DỮ LIỆU

## 2.1. Giới thiệu

Thiết kế cơ sở dữ liệu (CSDL) đóng vai trò quan trọng trong việc lưu trữ và quản lý dữ liệu trong ứng dụng chat. Đối với một ứng dụng chat, CSDL không chỉ giữ thông tin về người dùng và tin nhắn mà còn phải đảm bảo tính toàn vẹn và hiệu suất của hệ thống. Một thiết kế cơ sở dữ liệu tốt sẽ giúp ứng dụng hoạt động mượt mà, dễ dàng mở rộng và bảo mật dữ liệu.

## 2.2. Thiết kế cơ sở dữ liệu

### A screenshot of a computer Description automatically generated2.2.1. Sơ đồ cơ sở dữ liệu

Hình 1. Sơ đồ cơ sở dữ liệu ứng dụng

### 2.2.2. Mô tả

#### a. Bảng user

Bảng user lưu trữ thông tin cơ bản về người dùng của ứng dụng chat. Mỗi dòng trong bảng này tương ứng với một người dùng và bao gồm các trường sau:

* UserID: Một số duy nhất đại diện cho ID của người dùng.
* UserName: Tên đăng nhập của người dùng.
* Password: Mật khẩu được mã hóa của người dùng.

#### b. Bảng user\_account

Bảng user\_account chứa thông tin phụ về người dùng và được liên kết với bảng user thông qua trường UserID. Các trường trong bảng này bao gồm:

* UserID: Một số duy nhất đại diện cho ID của người dùng, là khóa ngoại tham chiếu đến bảng user.
* UserName: Tên đăng nhập của người dùng.
* Gender: Giới tính của người dùng, có thể là 'M' (Nam), 'F' (Nữ) hoặc trống (không xác định).
* Image: Hình ảnh của người dùng, được lưu dưới dạng dữ liệu nhị phân (BLOB).
* ImageString: Đường dẫn hoặc chuỗi định danh của hình ảnh.
* Status: Trạng thái của người dùng, có thể là '1' (hoạt động) hoặc '0' (không hoạt động).

#### c. Mối quan hệ giữa các bảng

Bảng user\_account có một mối quan hệ nối (foreign key) với bảng user thông qua trường UserID, đảm bảo rằng thông tin về người dùng trong hai bảng này được đồng bộ và không có sự mất mát dữ liệu khi xóa một người dùng từ bảng user.

Cấu trúc cơ sở dữ liệu này được thiết kế để lưu trữ thông tin về người dùng và hỗ trợ tính năng như xác thực người dùng, quản lý hồ sơ cá nhân và trạng thái trực tuyến của người dùng trong ứng dụng chat.

# CHƯƠNG 3: THIẾT KẾ GIAO DIỆN

## A screenshot of a computer Description automatically generated3.1. Giao diện khởi động

Hình 2. Sever đang được khởi động

Giao diện này đóng vai trò quan trọng trong việc khởi động và giám sát hoạt động của server. Người dùng có thể dễ dàng biết được server đã khởi động hay chưa và theo dõi các kết nối cũng như hoạt động của server thông qua các thông báo hiển thị trong cửa sổ chính.

A screenshot of a computer

Description automatically generatedHình 3. Sever hiển thị 1 client kết nối khi mở ứng dụng

Giao diện này cung cấp thông tin trực quan về trạng thái kết nối của server với các client. Khi một client kết nối, thông báo sẽ được cập nhật ngay lập tức trong khung thông báo, giúp người dùng dễ dàng giám sát các kết nối hiện tại.

## A person holding a login screen Description automatically generated3.2. Giao diện đăng nhập

Hình 4. Giao diện đăng nhập

Giao diện này cung cấp một cách thức đơn giản và trực quan để người dùng đăng nhập vào hệ thống chat. Các trường thông tin được sắp xếp dễ nhìn và dễ sử dụng. Hình minh họa và thiết kế màu sắc nhẹ nhàng giúp tạo cảm giác thân thiện, thoải mái cho người dùng.

## A person holding a login form Description automatically generated3.3. Giao diện đăng ký

Hình 5. Giao diện đăng ký

Giao diện cung cấp một cách thức đơn giản và trực quan để người dùng đăng ký tài khoản mới.

A screenshot of a computer error

Description automatically generated

Hình 6. Thông tin user được gửi về sever sau khi đăng ký

Khi người dùng đăng ký, thông tin tài khoản sẽ được gửi từ client đến server nhằm đảm bảo rằng thông tin đăng ký của người dùng được gửi đến server một cách an toàn và được xử lý kịp thời.

## A white background with black text Description automatically generated3.4. Giao diện chat

Hình 7. Giao diện sau khi đăng nhập

Giao diện chính sau khi đăng nhập được thiết kế để người dùng có thể dễ dàng giao tiếp với nhau thông qua các tin nhắn trong thời gian thực. Danh sách người dùng trực tuyến giúp tạo kết nối và tương tác tốt hơn giữa các người dùng.

A screenshot of a computer

Description automatically generatedHình 8. Giao diện khi chọn 1 người chat

A screenshot of a computer

Description automatically generatedGiao diện này giúp người dùng dễ dàng chọn một người từ danh sách người dùng trực tuyến để bắt đầu cuộc trò chuyện riêng tư.

Hình 9. Gửi và nhận tin nhắn văn bản

Giao diện và chức năng gửi và nhận tin nhắn văn bản cho phép người dùng giao tiếp dễ dàng và tức thì. Tin nhắn được hiển thị ngay lập tức trong cửa sổ chat, giúp duy trì cuộc trò chuyện liên tục và mạch lạc.

A screenshot of a computer

Description automatically generatedHình 10. Tính năng gửi emoji

Giao diện và chức năng gửi emoji giúp cuộc trò chuyện trở nên sinh động và thú vị hơn. Người dùng có thể dễ dàng chọn và gửi emoji để biểu đạt cảm xúc một cách trực quan.

# TỔNG KẾT

Dưới sự hướng dẫn của giảng viên và sự nỗ lực của cả nhóm, nhóm chúng em rất tự hào khi hoàn thành đề tài xây dựng ứng dụng chat này. Mặc dù đã đối diện với một số thách thức, nhưng chúng em đã cố gắng để mang lại một sản phẩm hoàn thiện và hiệu quả.

Ứng dụng chat này được phát triển với các tính năng cơ bản như gửi và nhận tin nhắn, đăng nhập, đăng ký tài khoản và quản lý danh sách bạn bè. Tính năng này không chỉ mang lại trải nghiệm giao tiếp tốt cho người dùng mà còn tạo ra một nền tảng linh hoạt để mở rộng và phát triển trong tương lai.

Qua quá trình này, chúng em nhận thấy rằng việc xây dựng một ứng dụng chat không chỉ là một dự án kỹ thuật mà còn là một cơ hội để thể hiện sự sáng tạo và sự kỹ năng trong lập trình. Bằng việc áp dụng kiến thức về socket và mô hình client – sever cùng ngôn ngữ lập trình java, chúng em đã tạo ra một sản phẩm có giá trị và có khả năng phát triển trong tương lai.

# TÀI LIỆU THAM KHẢO

1. Beej's Guide to Network Programming: A comprehensive guide to socket programming in C. Link: http://beej.us/guide/bgnet/
2. "Java Socket Programming" by Richard G. Baldwin: A tutorial on socket programming using Java. Link: https://www.developer.com/java/data/working-with-sockets-in-java.html
3. Oracle Java Documentation: Official documentation for Java programming language, including networking APIs. Link: <https://docs.oracle.com/en/java/>
4. "Head First Java" by Kathy Sierra and Bert Bates: A beginner-friendly book on Java programming with examples and exercises. Link: https://www.oreilly.com/library/view/head-first-java/0596009208/
5. "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross: A textbook covering networking concepts and protocols. Link: <https://www.amazon.com/Computer-Networking-Top-Down-Approach-7th/dp/0133594149/>
6. "TCP/IP Illustrated, Volume 1: The Protocols" by W. Richard Stevens: A comprehensive book on TCP/IP protocols with detailed explanations. Link: <https://www.amazon.com/TCP-Illustrated-Vol-Addison-Wesley-Professional-ebook/dp/B009X1DATC/>
7. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: A classic book on software design patterns. Link: <https://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612/>
8. Stack Overflow: A popular online community for programmers to ask and answer questions on various topics, including socket programming. Link: <https://stackoverflow.com/>
9. GitHub: An online platform for version control using Git. Many open-source projects related to socket programming can be found here. Link: <https://github.com/>
10. Online tutorials and forums: Websites like TutorialsPoint, GeeksforGeeks, and Reddit's programming communities provide valuable resources and discussions on socket programming.

Links:

* https://www.tutorialspoint.com/java/java\_networking.htm
* https://www.geeksforgeeks.org/socket-programming-in-java/
* <https://www.reddit.com/r/learnprogramming/>