

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



BÁO CÁO ĐO ÁN
MÔN LẬP TRÌNH MẠNG CĂN BẢN
LỚP: NT106.N22.MMCL
Đề tài: STUDY CHAT APPLICATION

GVHD: ThS. Lê Minh Khánh Hội

Nhóm sinh viên thực hiện:

- | | |
|--------------------|----------------|
| 1. Đặng Trung Hậu | MSSV: 21520833 |
| 2. Hồ Mạnh Đạt | MSSV: 21520695 |
| 3. Nguyễn Gia Quân | MSSV: 21521327 |

TP. Hồ Chí Minh, 06/2023

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

....., ngày tháng năm 20...

Người nhận xét

(Ký tên và ghi rõ họ tên)

LỜI MỞ ĐẦU

Đồ án của nhóm chúng em là một nghiên cứu về một ứng dụng khá phổ biến trong lĩnh vực công nghệ thông tin. Sau khi tìm hiểu và thảo luận về chủ đề này, chúng em đã tiến hành phân tích và thiết kế cũng như triển khai một số tính năng trên app.

Đồ án này đã đặt ra một thách thức lớn đối với nhóm chúng em về việc tìm kiếm thông tin và tài liệu về chủ đề cũng như việc thiết kế và triển khai một số tính năng mới. Tuy nhiên, nhờ có Cô Lê Minh Khánh Hội đã tận tình giảng dạy, giúp đỡ chúng em về mặt lý thuyết để chúng em có thêm kiến thức và kỹ năng cần thiết để làm việc hiệu quả và giải quyết vấn đề.

Chúng em đã nỗ lực và cố gắng hoàn thành tốt nhất đề tài đã chọn, nhưng vẫn không thể tránh khỏi được những thiếu sót và những hạn chế trong quá trình hoàn thành đồ án. Mong Cô và các bạn thông cảm, góp ý thêm để đồ án của nhóm chúng em được hoàn thiện hơn.

Sau đây, nhóm chúng em xin trình bày báo cáo tiền độ của đồ án mà nhóm chúng em đã thực hiện. Phần báo cáo nói về tổng quan, quy trình phân tích thiết kế hệ thống, các tính năng của hệ thống cũng như cách triển khai ứng dụng qua các chương sau:

- Chương 1: Tổng quan
- Chương 2: Phân tích thiết kế hệ thống
- Chương 3: Hiện thực hệ thống
- Chương 4: Kết luận

DANH MỤC CÁC HÌNH ẢNH

Hình 2.1. 1 Mô hình kiến trúc hệ thống	12
Hình 2.1. 2 Databases chương trình	12
Hình 2.1. 3 Mô hình truyền gói tin đi giữa Client – Server	12
Hình 2.3. 1 Sơ đồ use case	14
Hình 2.4. 1 Luồng của ứng dụng	14
Hình 3.1. 1 Giao diện vào app	15
Hình 3.1. 2 Giao diện đăng nhập	15
Hình 3.1. 3 Giao diện thông báo sai mật khẩu	16
Hình 3.1. 4 Giao diện quên mật khẩu	16
Hình 3.1. 5 Giao diện tạo mật khẩu mới	16
Hình 3.2. 1 Giao diện đăng ký	17
Hình 3.2. 2 Giao diện xác thực email sau khi đăng ký	17
Hình 3.2. 3 Giao diện thông báo đăng ký thành công tài khoản	17
Hình 3.3. 1 Giao diện màn hình chính	18
Hình 3.3. 2 Giao diện lựa chọn	18
Hình 3.3. 3 Giao diện tạo phòng	18
Hình 3.3. 4 Giao diện tham gia phòng	19
Hình 3.3. 5 Giao diện phòng chat	19
Hình 3.4. 1 Giao diện server lắng nghe kết nối	20
Hình 3.4. 2 Giao diện trạng thái của client	20
Hình 3.4. 3 File gửi qua sẽ nằm trong thư mục nhận file trong máy tính	20

MỤC LỤC

LỜI MỞ ĐẦU	3
DANH MỤC CÁC HÌNH ẢNH	4
Chương 1: TỔNG QUAN	6
1.1Giới thiệu đề tài.....	6
1.1.1 Lí do chọn đề tài	6
1.1.2 Mục đích nghiên cứu	6
1.1.3 Phạm vi nghiên cứu	6
1.2Cơ sở lí thuyết.....	6
1.2.1 Công nghệ sử dụng	6
1.2.2 Các thư viện	9
1.2.3 Lập trình Socket.....	10
Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	12
2.1Sơ đồ kiến trúc tổng quan hệ thống	12
2.2Network stack	12
2.2.1 Network stack	12
2.2.2 Message structure	13
2.3Sơ đồ use case	14
2.4Luồng của ứng dụng	14
Chương 3: HIỆN THỰC HỆ THỐNG	15
3.1Tính năng đăng nhập.....	15
3.2Tính năng đăng ký	17
3.3Tính năng chat.....	18
3.4Tính năng gửi file.....	20
Chương 4: KẾT LUẬN	21
4.1Kết quả đạt được	21
4.2Hạn chế	21
4.3Hướng phát triển	21
BẢNG PHÂN CÔNG CÔNG VIỆC	21
TÀI LIỆU THAM KHẢO	22

Chương 1: TỔNG QUAN

1.1 Giới thiệu đề tài

Trong sự phát triển của công nghệ thông tin, đặc biệt là sự phát triển của các ứng dụng Chat là một trong những ứng dụng rất phổ biến và tiện lợi cho người dùng. Việc ứng dụng và phát triển ứng dụng này là nhu cầu cần thiết của cuộc sống hiện nay. Trên thế giới, ứng dụng Chat đã phát triển khá lâu và đã được đẩy mạnh trong những năm trở lại đây như SMS, Facebook Messenger, WhatsApp, Zalo, WeChat, Line,...

Ứng dụng Chat là một ứng dụng cho phép người dùng gửi – nhận tin nhắn, hình ảnh, tài liệu, gọi điện, call video và chia sẻ các loại dữ liệu khác đến với những người dùng khác. Với sự phát triển của công nghệ hiện đại, để phát triển một ứng dụng Chat chất lượng và sáng tạo, cần phải tập trung vào việc tối ưu hóa giao diện người dùng, nâng cao cơ sở dữ liệu, cải thiện độ bảo mật và phát triển các tính năng mới trên app...

1.1.1 Lí do chọn đề tài

- Nhằm tìm hiểu về ngôn ngữ lập trình C# và nền tảng .NET, các thao tác lập trình mạng, thuật toán, cách quản lý code, xử lý lỗi.
- Ôn tập lại kiến thức về giao thức mạng của môn học Nhập môn mạng máy tính như TCP/IP, HTTP, WebSocket,...
- Thực hành lập trình phía máy chủ và máy khách để có thể giao tiếp với nhau.
- Tìm hiểu về các kỹ thuật mã hóa và bảo mật thông tin trong quá trình truyền tin qua mạng, áp dụng vào ứng dụng Chat.

1.1.2 Mục đích nghiên cứu

- Tìm hiểu các thao tác lập trình mạng, cách quản lý code, xử lý lỗi.
- Nâng vững và hiểu rõ các giao thức trong lập trình mạng.
- Thực hành và nâng cao các kỹ năng lập trình phía máy chủ và máy khách.
- Tìm hiểu về các kỹ thuật mã hóa và bảo mật thông tin.

1.1.3 Phạm vi nghiên cứu

Giới hạn trong những xử lý cơ bản, sử dụng các framework phổ biến, các giao thức truyền thông như HTTP, WebSocket..., hỗ trợ sử dụng C# trong lập trình.

1.2 Cơ sở lí thuyết

1.2.1 Công nghệ sử dụng

- C# và .NET Framework:

+ Khái niệm:

C# và .NET Framework là hai công nghệ phổ biến trong lĩnh vực phát triển phần mềm. Dưới đây là một giới thiệu tổng quan về chúng:

- C#:
 - C# (đọc là "C sharp") là một ngôn ngữ lập trình đa mô hình được phát triển bởi Microsoft.

- C# được thiết kế để xây dựng ứng dụng phần mềm cho nền tảng Windows, nhưng nó cũng có thể được sử dụng cho các ứng dụng web, di động và dịch vụ đám mây.
- C# kết hợp các tính năng từ nhiều ngôn ngữ khác nhau như C++, Java và Visual Basic để tạo ra một ngôn ngữ mạnh mẽ và dễ sử dụng.
- C# hỗ trợ lập trình hướng đối tượng, quản lý bộ nhớ tự động, xử lý ngoại lệ, đa luồng và nhiều tính năng khác, giúp các nhà phát triển xây dựng các ứng dụng phức tạp.
- .NET Framework:
 - .NET Framework là một nền tảng phát triển phần mềm được phát triển bởi Microsoft.
 - Nền tảng này cung cấp một môi trường thực thi cho các ứng dụng và các dịch vụ web dựa trên C#, VB.NET và F#.
 - .NET Framework bao gồm một tập hợp các thư viện lớn, gọi là Framework Class Library (FCL), cung cấp các lớp và phương thức tiện ích để xây dựng các ứng dụng.
 - Ngoài ra, .NET Framework còn có Common Language Runtime (CLR), một môi trường thực thi quản lý, có khả năng quản lý bộ nhớ tự động, xử lý ngoại lệ và điều phối các luồng thực thi.
 - .NET Framework hỗ trợ xây dựng ứng dụng Windows Forms, WPF (Windows Presentation Foundation), ASP.NET (ứng dụng web) và nhiều công nghệ khác.

+ Ưu điểm C# và .NET Framework:

- Đa nền tảng: C# và .NET Framework cho phép phát triển ứng dụng đa nền tảng. Với sự ra đời của .NET Core và phiên bản mới nhất là .NET 5.0, bạn có thể xây dựng ứng dụng chạy trên Windows, macOS và Linux.
- Hệ sinh thái phong phú: .NET Framework cung cấp một hệ sinh thái phát triển mạnh mẽ với các thư viện và framework như ASP.NET, WPF, WinForms và nhiều hơn nữa. Bạn có thể tận dụng các công cụ này để phát triển ứng dụng desktop, web và di động.
- Lập trình hướng đối tượng: C# hỗ trợ lập trình hướng đối tượng, cho phép bạn xây dựng ứng dụng có cấu trúc rõ ràng và dễ bảo trì. Việc sử dụng các khái niệm như lớp, đối tượng, kế thừa và giao diện giúp tái sử dụng mã và tạo ra mã dễ đọc và linh hoạt.
- An toàn và tin cậy: C# sử dụng Common Language Runtime (CLR) để quản lý bộ nhớ tự động và xử lý ngoại lệ. Điều này giúp giảm thiểu các lỗi như tràn bộ đệm và nâng cao độ tin cậy của ứng dụng.

+ Nhược điểm C# và .NET Framework:

- Hạn chế đa nền tảng ban đầu: Trước khi .NET Core ra mắt, .NET Framework chỉ hỗ trợ chạy trên nền tảng Windows. Điều này tạo ra giới hạn khi muốn phát triển ứng dụng đa nền tảng. Tuy nhiên, việc có .NET Core đã giúp khắc phục hạn chế này.

- Phụ thuộc vào Microsoft: C# và .NET Framework đều là sản phẩm của Microsoft, điều này có thể tạo ra một mức độ phụ thuộc vào hãng công nghệ này. Sự phụ thuộc này có thể tạo ra khó khăn khi muốn chuyển đổi sang một nền tảng hoặc công nghệ khác.
 - Tương thích ngược: Các phiên bản mới của .NET Framework và C# có thể không tương thích ngược với các phiên bản cũ hơn. Điều này có thể tạo ra vấn đề cho việc duy trì và nâng cấp các ứng dụng đã tồn tại.
- Sự kết hợp giữa C# và .NET Framework tạo ra một môi trường phát triển mạnh mẽ cho việc xây dựng các ứng dụng phần mềm đa nền tảng. Các nhà phát triển có thể sử dụng C# để viết mã và sử dụng các thư viện và công cụ của .NET Framework để phát triển các ứng dụng chất lượng cao và linh hoạt trên nhiều nền tảng khác nhau.
- Hệ quản trị cơ sở dữ liệu Microsoft SQL Server:
 - + Khái niệm:
 - Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS), cho phép lưu trữ, truy xuất và quản lý dữ liệu theo cấu trúc bảng và mối quan hệ giữa chúng.
 - SQL Server hỗ trợ ngôn ngữ truy vấn SQL (Structured Query Language), cho phép người dùng tương tác và thao tác với dữ liệu trong cơ sở dữ liệu.
 - + Ưu điểm:
 - Hiệu suất cao: SQL Server có khả năng xử lý tải lớn và thực hiện các truy vấn phức tạp một cách nhanh chóng. Nó cung cấp các công cụ tối ưu hóa truy vấn và chỉ mục để cải thiện hiệu suất và thời gian đáp ứng.
 - Bảo mật và quản lý dữ liệu: SQL Server cung cấp các tính năng bảo mật mạnh mẽ như kiểm soát quyền truy cập, mã hóa dữ liệu và theo dõi hoạt động người dùng. Nó cũng có các công cụ quản lý dữ liệu để sao lưu, phục hồi và giám sát cơ sở dữ liệu.
 - Độ tin cậy cao: SQL Server có tính năng khôi phục dữ liệu tự động và hỗ trợ giao thức ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo tính toàn vẹn và nhất quán của dữ liệu.
 - Hỗ trợ mở rộng: SQL Server tích hợp tốt với các công nghệ và dịch vụ của Microsoft như .NET Framework, Azure, Business Intelligence và Machine Learning. Nó cũng hỗ trợ các ngôn ngữ lập trình như C#, Java, và Python.
 - + Nhược điểm:
 - Hạn chế đa nền tảng: SQL Server chủ yếu hỗ trợ nền tảng Windows, và hỗ trợ đối với các hệ điều hành khác như Linux và macOS còn hạn chế.
 - Khả năng mở rộng: Một số phiên bản SQL Server có hạn chế về khả năng mở rộng so với các hệ quản trị cơ sở dữ liệu khác. Điều này có thể gây khó khăn trong việc xử lý các ứng dụng có lưu lượng truy vấn lớn hoặc quy mô lớn.
 - GUNA.UI2.WINFORM:

+ Khái niệm:

- Guna.UI2.WinForms cung cấp một tập hợp các thành phần và điều khiển giao diện người dùng đã được thiết kế trước (pre-designed), giúp người phát triển xây dựng giao diện ứng dụng nhanh chóng và dễ dàng hơn.
- Nó hỗ trợ các tính năng như chuyển đổi (toggle), thanh trượt (slider), nút (button), bảng điều khiển (dashboard), biểu đồ (chart) và nhiều hơn nữa, giúp tạo ra các giao diện đẹp và chuyên nghiệp.

+ Ưu điểm:

- Thiết kế giao diện tốn ít thời gian: Guna.UI2.WinForms cung cấp các thành phần đã được thiết kế trước, giúp giảm thời gian và công sức trong việc xây dựng giao diện người dùng. Bạn có thể dễ dàng kéo và thả các điều khiển vào giao diện và tùy chỉnh chúng theo ý muốn.
- Giao diện đẹp mắt và chuyên nghiệp: Guna.UI2.WinForms cung cấp các kiểu dáng và hiệu ứng đồ họa hấp dẫn, giúp tạo ra giao diện ứng dụng có vẻ ngoài chuyên nghiệp và hiện đại.
- Tích hợp dễ dàng: Guna.UI2.WinForms tích hợp tốt với Windows Forms, cho phép bạn sử dụng các điều khiển Guna.UI2.WinForms kết hợp với các điều khiển Windows Forms khác. Điều này giúp bạn mở rộng khả năng thiết kế và tùy chỉnh giao diện người dùng.

+ Nhược điểm:

- Phụ thuộc vào bên thứ ba: Guna.UI2.WinForms là một công cụ của bên thứ ba, nên bạn cần có quyền truy cập và sử dụng các thành phần và tính năng của nó. Điều này có thể tạo ra một mức độ phụ thuộc và hạn chế khi muốn tùy chỉnh hoặc mở rộng các thành phần.
- Giới hạn tính linh hoạt: Mặc dù Guna.UI2.WinForms cung cấp các điều khiển giao diện người dùng đã được thiết kế trước, nhưng nó có giới hạn về tính linh hoạt so với việc tạo giao diện người dùng tùy chỉnh từ đầu.

Tóm lại, Guna.UI2.WinForms là một bộ công cụ giao diện người dùng dành cho Windows Forms, giúp tăng tốc quá trình phát triển giao diện và tạo ra các giao diện đẹp mắt. Tuy nhiên, nó có nhược điểm liên quan đến phụ thuộc vào bên thứ ba và giới hạn tính linh hoạt.

1.2.2 Các thư viện

- Thư viện JSON

+ Khái niệm:

- JSON là một định dạng dữ liệu nhẹ và độc lập với ngôn ngữ, phổ biến trong việc truyền tải và lưu trữ dữ liệu cấu trúc. Nó sử dụng cú pháp dễ đọc và dễ hiểu cho con người, cũng như dễ dàng để phân tích và tạo dữ liệu bằng các ngôn ngữ lập trình.

+ Ưu điểm:

- Dễ đọc và dễ hiểu: JSON sử dụng cú pháp đơn giản và gần giống với cú pháp của JavaScript, dễ đọc và hiểu cho con người. Điều này giúp dễ dàng xử lý và gỡ lỗi dữ liệu JSON.
- Hỗ trợ đa ngôn ngữ: JSON là một định dạng độc lập với ngôn ngữ, được hỗ trợ bởi hầu hết các ngôn ngữ lập trình phổ biến như JavaScript, Python,

Java, C# và nhiều ngôn ngữ khác. Điều này cho phép truyền tải và chia sẻ dữ liệu giữa các ứng dụng được phát triển bằng các ngôn ngữ khác nhau.

- Dung lượng nhỏ gọn: JSON sử dụng cú pháp rõ ràng và không chứa nhiều dữ liệu phụ, điều này giúp giảm kích thước của dữ liệu so với các định dạng khác như XML. Điều này làm cho việc truyền tải và lưu trữ dữ liệu JSON hiệu quả về mặt băng thông và dung lượng đĩa.

+ Nhược điểm:

- Hạn chế về kiểu dữ liệu: JSON hỗ trợ các kiểu dữ liệu cơ bản như chuỗi, số, đối tượng, mảng, boolean và null. Tuy nhiên, nó không hỗ trợ trực tiếp các kiểu dữ liệu phức tạp như ngày giờ, địa chỉ IP, hoặc binary. Khi làm việc với các kiểu dữ liệu phức tạp, người dùng cần thực hiện chuyển đổi dữ liệu thủ công.
- Thiếu hỗ trợ cho các toán tử và xử lý dữ liệu phức tạp: JSON được thiết kế để làm việc với cấu trúc dữ liệu cơ bản, không cung cấp các toán tử hay các chức năng xử lý dữ liệu phức tạp. Điều này có nghĩa là các phép tính hay xử lý dữ liệu phức tạp phải được thực hiện bên ngoài của JSON.

Tóm lại, thư viện JSON giúp xử lý và tương tác với định dạng dữ liệu JSON một cách dễ dàng và linh hoạt. Nó có ưu điểm như dễ đọc, hỗ trợ đa ngôn ngữ và dung lượng nhỏ gọn, nhưng cũng có nhược điểm liên quan đến hạn chế kiểu dữ liệu và xử lý dữ liệu phức tạp.

1.2.3 Lập trình Socket

Lập trình socket là quá trình xây dựng ứng dụng mạng sử dụng giao thức TCP/IP để thiết lập kết nối và truyền thông giữa các thiết bị trong mạng. Dưới đây là giải thích ngắn gọn về khái niệm, ưu điểm và nhược điểm của lập trình socket:

+ Khái niệm:

Socket là một đối tượng trùu tượng trong lập trình mạng, đại diện cho một đầu cuối của một kênh truyền thông hai chiều. Nó cho phép ứng dụng truyền và nhận dữ liệu qua mạng thông qua các kết nối TCP/IP hoặc UDP.

+ Ưu điểm:

- Linh hoạt: Lập trình socket cho phép bạn tạo các ứng dụng mạng linh hoạt và tùy chỉnh theo nhu cầu cụ thể. Bạn có thể xây dựng ứng dụng mạng đa nền tảng và đa dạng, như ứng dụng chat, trò chơi trực tuyến, truyền tệp, và nhiều hơn nữa.
- Tính tin cậy: Giao thức TCP/IP sử dụng bởi lập trình socket đảm bảo tính tin cậy trong truyền dữ liệu. Dữ liệu được chia thành các gói và kiểm tra lỗi để đảm bảo việc truyền và nhận dữ liệu đúng và hoàn chỉnh.
- Tốc độ truyền dữ liệu: Lập trình socket sử dụng giao thức TCP/IP, có khả năng truyền dữ liệu một cách hiệu quả và nhanh chóng qua mạng. Điều này làm cho lập trình socket trở thành một lựa chọn phổ biến cho các ứng dụng yêu cầu truyền tải dữ liệu lớn hoặc liên tục.

+ Nhược điểm:

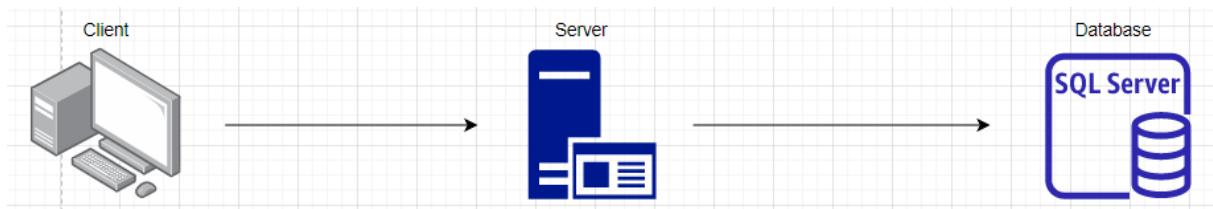
- Phức tạp: Lập trình socket yêu cầu hiểu và thực hiện các khái niệm liên quan đến giao thức TCP/IP và quản lý kết nối mạng. Điều này có thể đòi hỏi sự hiểu biết chuyên sâu và kỹ năng lập trình cao để triển khai và gỡ lỗi ứng dụng mạng.

- **Khả năng mở rộng:** Khi xây dựng các ứng dụng mạng phức tạp hoặc có số lượng lớn người dùng, việc quản lý và mở rộng các kết nối socket có thể trở nên khó khăn. Cần thiết phải có các thiết kế và quản lý cẩn thận để đảm bảo hiệu suất và khả năng mở rộng của hệ thống.
- **Firewall và NAT:** Một số ứng dụng sử dụng lập trình socket có thể gặp khó khăn khi hoạt động qua các tường lửa (firewall) hoặc các thiết bị NAT (Network Address Translation) do các hạn chế về mở kết nối và địa chỉ IP.

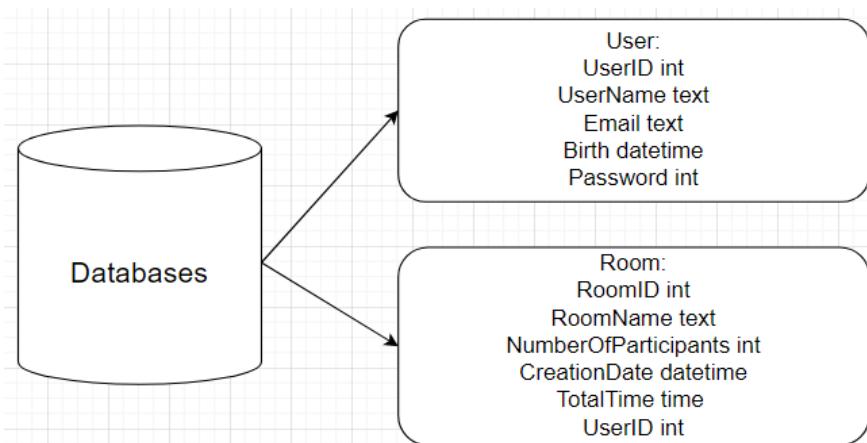
Tóm lại, lập trình socket là một phương pháp mạnh mẽ để xây dựng ứng dụng mạng. Nó có ưu điểm như tính linh hoạt, tính tin cậy và tốc độ truyền dữ liệu, nhưng cũng có nhược điểm liên quan đến sự phức tạp, khả năng mở rộng và khó khăn trong việc vượt qua tường lửa và NAT.

Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

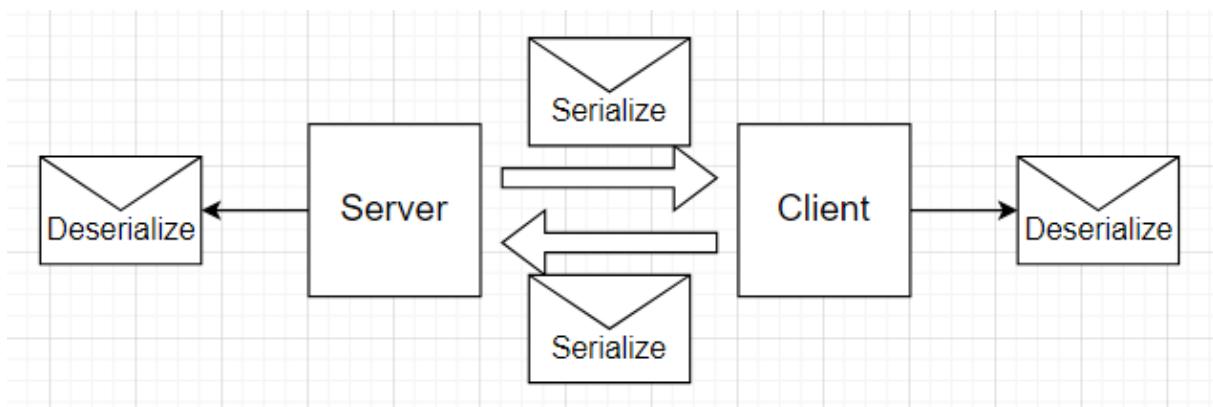
2.1 Sơ đồ kiến trúc tổng quan hệ thống



Hình 2.1. 1 Mô hình kiến trúc hệ thống



Hình 2.1. 2 Databases chương trình



Hình 2.1. 3 Mô hình truyền gói tin đi giữa Client – Server

2.2 Network stack

2.2.1 Network stack

Application	Socket: Trong tầng Application, lập trình socket được sử dụng để truyền tải dữ liệu, giao tiếp giữa các người dùng, truy cập dịch vụ mạng và kết nối ứng dụng. Nó là một công nghệ quan trọng để xây dựng các ứng dụng mạng mạnh mẽ và linh hoạt.
Presentation	Byte format: Việc sử dụng định dạng byte format trong tầng Presentation giúp mã hóa, nén, giải mã và đảm bảo tính toàn vẹn của dữ liệu trong quá trình truyền tải và hiển thị thông tin qua mạng. Nó đóng vai trò quan trọng trong việc đảm bảo dữ liệu được truyền tải và hiển thị chính xác và hiệu quả.

Session	Socket windows: Việc sử dụng socket Windows trong tầng Session cho phép quản lý quá trình thiết lập, duy trì và kết thúc phiên liên lạc giữa các ứng dụng. Nó cung cấp các phương thức để thiết lập kết nối, truyền và nhận dữ liệu, đồng bộ hóa và bảo mật. Điều này đảm bảo tính toàn vẹn, ổn định và bảo mật của phiên liên lạc giữa các ứng dụng trong tầng Session.
Transport	TCP: Tầng Transport sử dụng giao thức TCP để đảm bảo việc truyền dữ liệu tin cậy, đúng thứ tự và không bị mất mát. TCP được sử dụng để thiết lập các kết nối và quản lý luồng dữ liệu giữa các ứng dụng trên các thiết bị kết nối.
Network	IP: Việc sử dụng IP trong tầng Network có ý nghĩa quan trọng trong việc định danh địa chỉ mạng, định tuyến, giao tiếp và liên kết mạng. Nó là giao thức cốt lõi để xác định đường dẫn và truyền tải dữ liệu giữa các thiết bị trong hệ thống mạng.
Data link	Ethernet: Việc sử dụng Ethernet trong tầng Data Link có ý nghĩa quan trọng trong việc quản lý kết nối vật lý, địa chỉ MAC, truy cập vào mạng, đóng gói dữ liệu và kiểm soát lỗi trong quá trình truyền dữ liệu qua mạng. Nó đóng vai trò quan trọng trong việc xác định, định dạng và truyền tải dữ liệu giữa các thiết bị mạng.
Physical	Router: Việc sử dụng router trong tầng Physical có ý nghĩa quan trọng trong việc chuyển tiếp gói tin, định tuyến, chuyển mạch và kiểm soát lưu lượng dữ liệu trong hệ thống mạng. Nó đóng vai trò quan trọng trong việc xác định và kết nối các mạng, đồng thời tạo ra đường dẫn tối ưu và quản lý lưu lượng dữ liệu hiệu quả.

2.2.2 Message structure

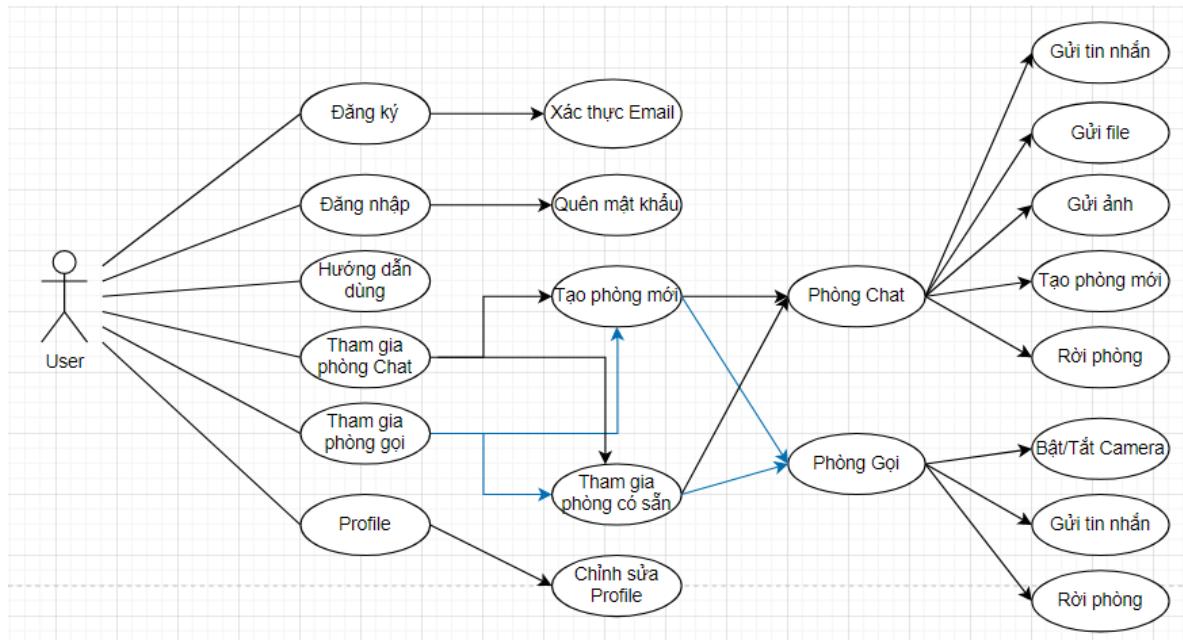
a. Server tới Client

PACKET				
Type Packet	User ID	Room ID	Content	Time
- Type Packet: Loại gói tin (như gửi tin nhắn, thông báo, tạo phòng chat hoặc đăng ký tài khoản,...).				
- User ID: Mã định danh của người dùng.				
- Room ID: Mã định danh phòng chat.				
- Content: Nội dung gói tin (như nội dung tin nhắn, kết quả đăng ký tài khoản...).				
- Time: Thời gian gửi tin.				

b. Client tới Server

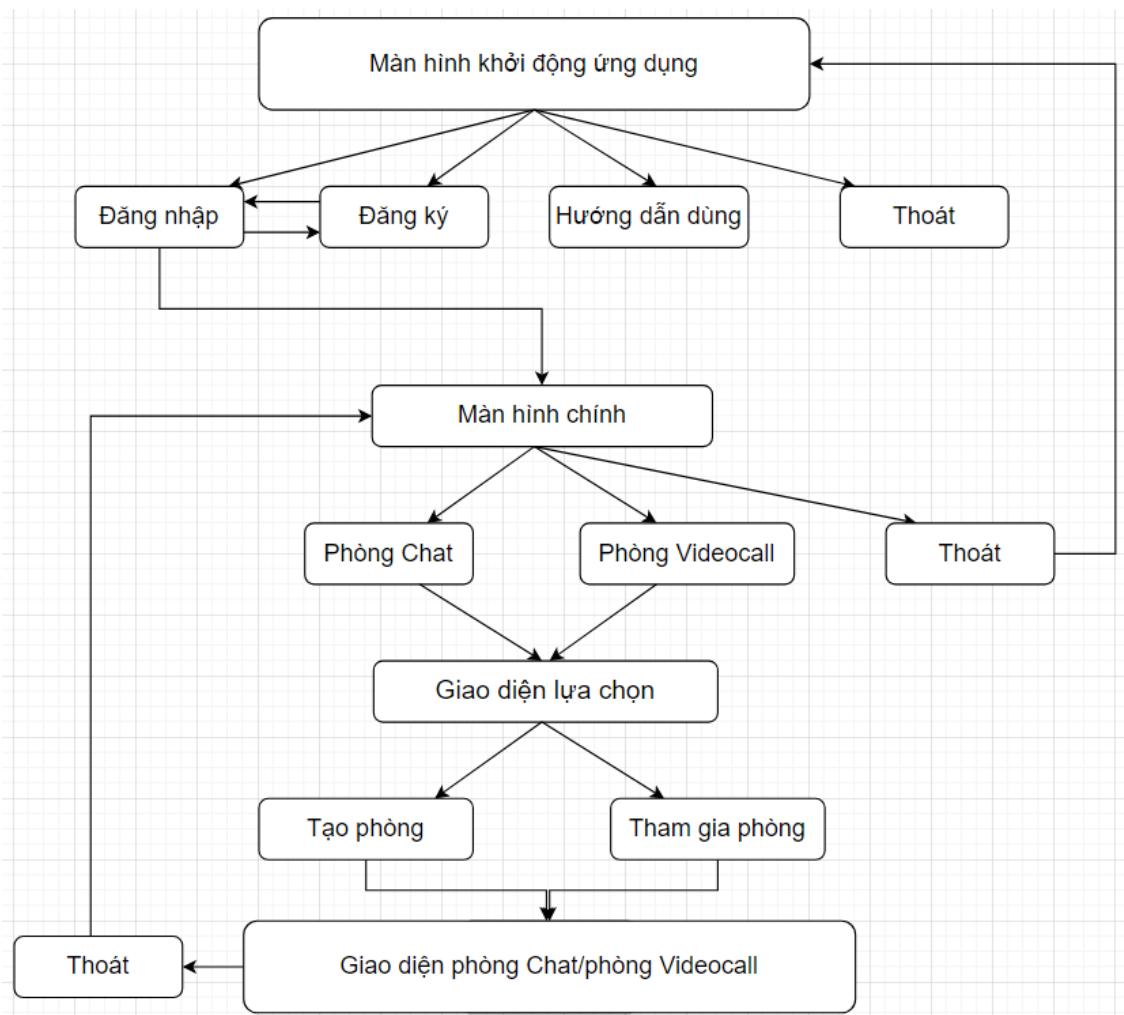
PACKET				
Type Packet	User ID	Room ID	Content	Time

2.3 Sơ đồ use case



Hình 2.3. 1 Sơ đồ use case

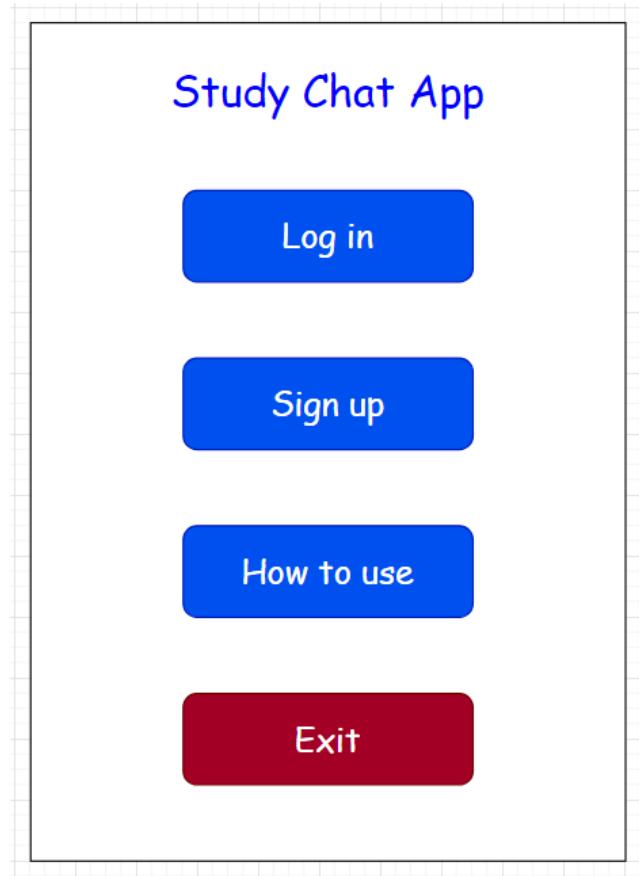
2.4 Luồng của ứng dụng



Hình 2.4. 1 Luồng của ứng dụng

Chương 3: HIỆN THỰC HỆ THỐNG

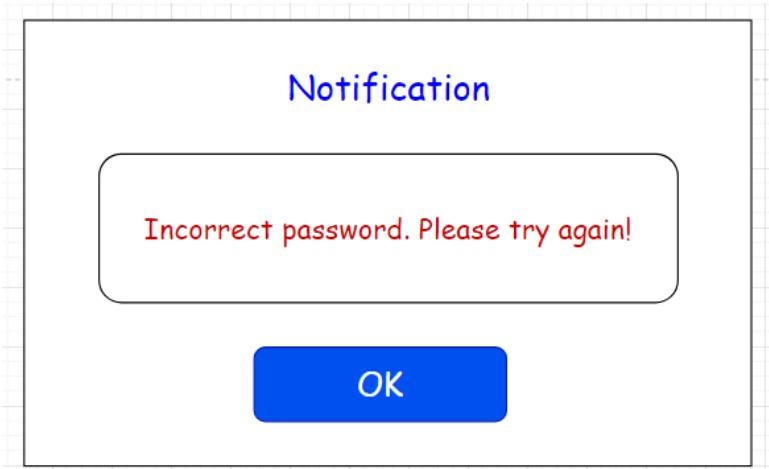
3.1 Tính năng đăng nhập



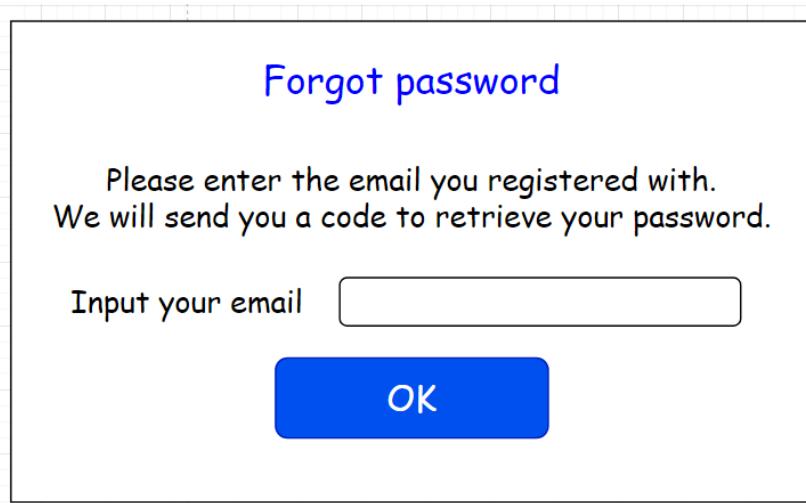
Hình 3.1. 1 Giao diện vào app

A screenshot of the Study Chat App's login screen. The screen has a white background with a thin black border. At the top center, the text "Log in" is displayed in blue. Below it is a text input field labeled "User name". Further down is another text input field labeled "Password". Underneath the password field are two links: "Forgot password?" and "Don't have an account yet? [Sign up](#)". At the bottom left is a blue button labeled "Log in", and at the bottom right is a red button labeled "Back".

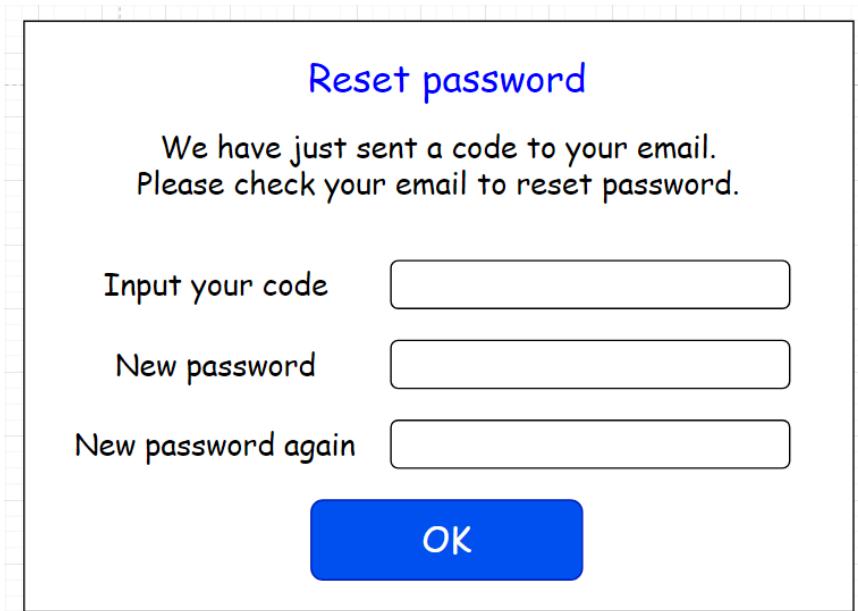
Hình 3.1. 2 Giao diện đăng nhập



Hình 3.1. 3 Giao diện thông báo sai mật khẩu



Hình 3.1. 4 Giao diện quên mật khẩu



Hình 3.1. 5 Giao diện tạo mật khẩu mới

3.2 Tính năng đăng ký

The image shows a sign-up form titled "Sign up". It contains four input fields: "Email", "User name", "Password", and "Birthday", each with a corresponding text input box. Below the input fields are two buttons: a blue "Sign up" button on the left and a red "Back" button on the right.

Hình 3.2. 1 Giao diện đăng ký

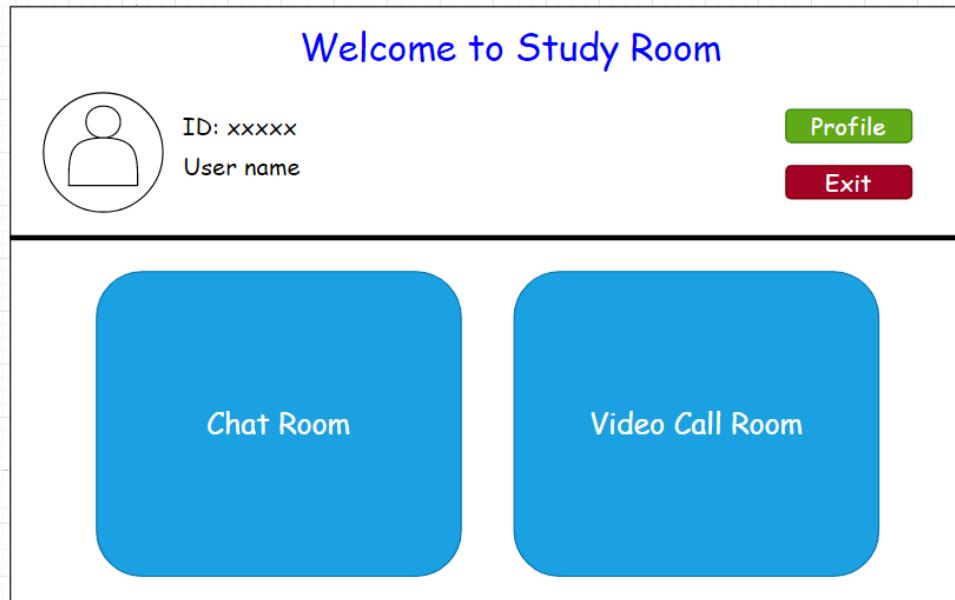
The image shows an email verification screen titled "Verify email". It displays a message: "We have just sent a code to your email. Please check your email to verify email." Below the message is an input field labeled "Input your code" with a placeholder text input box. At the bottom is a large blue "OK" button.

Hình 3.2. 2 Giao diện xác thực email sau khi đăng ký

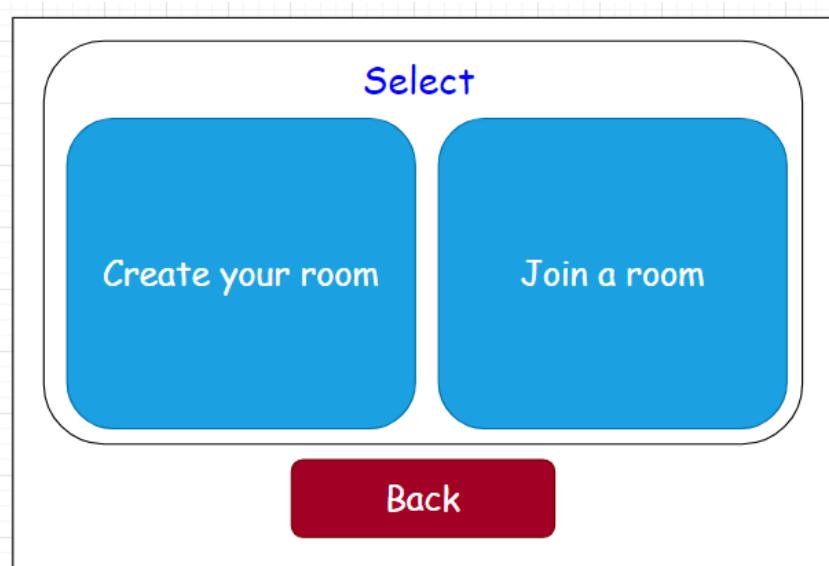
The image shows a success notification screen titled "Notification". It contains a message box with the text: "Congratulations on successfully registering your account. Please log in to use this app." At the bottom is a large blue "OK" button.

Hình 3.2. 3 Giao diện thông báo đăng ký thành công tài khoản

3.3 Tính năng chat



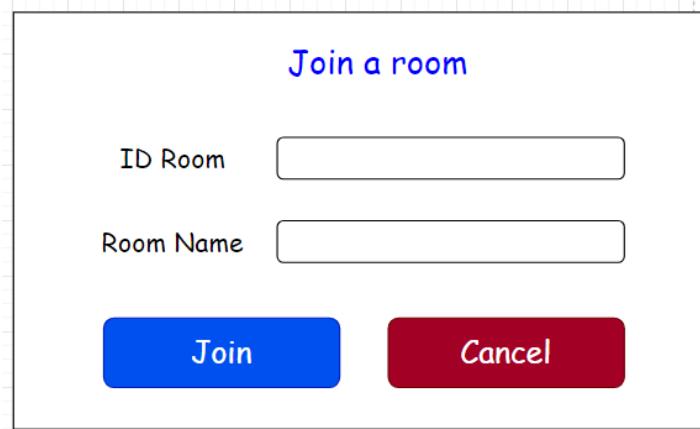
Hình 3.3. 1 Giao diện màn hình chính



Hình 3.3. 2 Giao diện lựa chọn

The image shows a form titled "Create your room". It contains three input fields: "ID Room", "Room Name", and "Number of participants", each with an associated text input box. At the bottom are two buttons: "Create" (blue) and "Cancel" (red).

Hình 3.3. 3 Giao diện tạo phòng



Hình 3.3. 4 Giao diện tham gia phòng

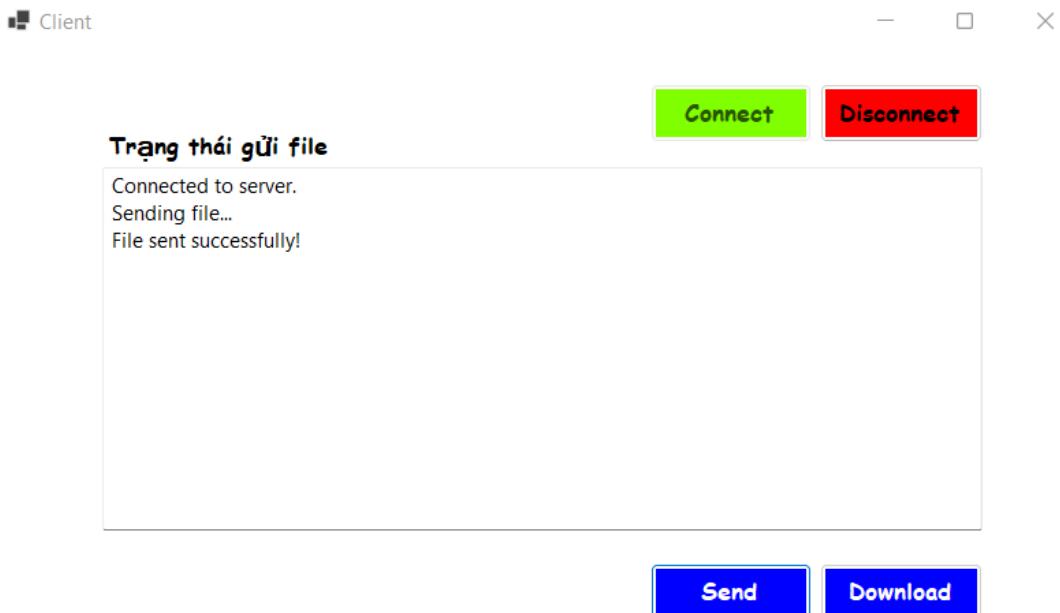
 ID User name	 Group ID Group name	<input type="button" value="Invite"/>	<input type="button" value="Leave Room"/>
 User name 1	 User 1 name <input type="text" value="Message..."/>		
 Group name 1	 User 2 name <input type="text" value="Message..."/>		
 User name 2		<input type="text" value="Message..."/>	
 Group name 2	 User 3 name <input type="text" value="Message..."/>		
<input type="button" value="Create your room"/>			
<input type="button" value="File"/>	<input type="button" value="Picture"/>	<input type="text" value="Type your message..."/>	
		<input type="button" value="Send"/>	

Hình 3.3. 5 Giao diện phòng chat

3.4 Tính năng gửi file



Hình 3.4. 1 Giao diện server lắng nghe kết nối



Hình 3.4. 2 Giao diện trạng thái của client

> This PC > DATA (D:) > HocC# > Files > Files > bin > Debug > net6.0-windows > ReceivedFiles

	Name	Date modified	Type	Size
CNTT -	huong_dan	6/5/2023 4:53 AM	File	1 KB
	nfjkds	6/5/2023 5:15 AM	File	1 KB

Hình 3.4. 3 File gửi qua sẽ nằm trong thư mục nhận file trong máy tính

Chương 4: KẾT LUẬN

4.1 Kết quả đạt được

- Thiết kế và triển khai được giao diện cũng như các tính năng cần thiết của đề tài.
- Ứng dụng được kiến thức của các môn học trước (Cơ sở dữ liệu, Nhập môn mạng máy tính) vào đề tài.
- Hiểu rõ hơn về ngôn ngữ lập trình C# Winform và các thư viện hỗ trợ.

4.2 Hạn chế

- Các chức năng của chương trình chưa được linh hoạt.
- Chưa làm được tính năng Video call.
- Chưa giải quyết được vấn đề xung đột giữa các client khi cùng hoạt động song song.

4.3 Hướng phát triển

- Nâng cấp chương trình, xử lý các sự kiện một cách linh hoạt hơn.
- Giải quyết vấn đề xung đột giữa các client bằng cách lập trình bắt đồng bộ.

BẢNG PHÂN CÔNG CÔNG VIỆC

STT	MSSV	Họ và tên	Nội dung công việc	Mức độ hoàn thành	Tự đánh giá (theo thang điểm 10)
1	21520833	Đặng Trung Hậu	Code client, hỗ trợ code server, tính năng chat, hỗ trợ viết báo cáo, slide, thuyết trình.	70%	7
2	21520695	Hồ Mạnh Đạt	Thiết kế giao diện, code tương tác giao diện, code tính năng gửi file/ảnh, viết báo cáo, slide.	70%	7
3	21521327	Nguyễn Gia Quân	Code server, tính năng đăng nhập, đăng ký, truy vấn database, hỗ trợ viết báo cáo, slide.	70%	7

TÀI LIỆU THAM KHẢO

- 1.** [Xây dựng ứng dụng Client-Server với Socket trong Java - GP Coder \(Lập trình Java\)](#) [Trực tuyến].
- 2.**
- 3.**