

**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 ĐỒ ÁN CUỐI KỲ**  
**LẬP TRÌNH MẠNG CĂN BẢN**

**ĐỀ TÀI**

**ChessMates – Cờ vua**

**Giảng viên hướng dẫn: ThS. Lê Minh Khánh Hội**

**Lớp: NT106.N21.MMCL**

<b>Sinh viên thực hiện:</b>	<b>MSSV</b>
<b>Lê Trần Phi Long</b>	<b>21522302</b>
<b>Dương Viết Đức</b>	<b>21521959</b>
<b>Hứa Hồ Gia Huy</b>	<b>21522143</b>
<b>Trương Chí Bảo</b>	<b>21521868</b>

*TP. Hồ Chí Minh, 6/2023*

## **LỜI CẢM ƠN**

Chúng em xin gửi lời cảm ơn đến **Trường Đại học Công nghệ Thông tin – ĐHQG TP.HCM** nói chung và các thầy cô trong **Khoa Mạng máy tính và Truyền thông** nói riêng đã tạo điều kiện cho chúng tôi cơ hội thực hành, tiếp xúc với những kiến thức, kinh nghiệm mới cho công việc trong thời gian tới cũng như hoàn thành đồ án về ứng dụng game cờ vua này.

Đặc biệt, chúng tôi xin chân thành cảm ơn **ThS. Lê Minh Khánh Hội**. Cô là người đã trực tiếp truyền đạt kiến thức chuyên môn sâu sắc và tạo điều kiện để chúng tôi có thể tiếp thu những kiến thức quan trọng trong suốt quá trình học tập. Sự tận tâm và những gợi ý, chỉ bảo của cô đã đóng vai trò quan trọng trong việc giúp chúng tôi hiểu rõ hơn về môn học và phát triển kỹ năng chuyên môn của mình.

Mặc dù đã cố gắng hoàn thành đề tài tốt nhất nhưng do thời gian và kiến thức có hạn nên chúng em không thể tránh khỏi những thiếu sót nhất định. Rất mong quý thầy cô thông cảm. Mong có những chia sẻ và tận tình đóng góp của quý thầy cô cũng như các bạn.

## MỤC LỤC

<b>Chương 1: Tổng quan .....</b>	<b>4</b>
I. Giới thiệu đề tài (lý do chọn đề tài, giới thiệu về ứng dụng, nếu là game thì giới thiệu luật chơi, các tính năng của ứng dụng, game).....	4
a. Lý do chọn đề tài.....	4
b. Luật chơi ( cách đi và ăn quân) :.....	4
Quân Tốt – Pawn .....	4
Quân Xe – Chariot, Rook .....	5
Quân Mã – Knight .....	8
Quân Tượng – Bishop.....	10
Quân Hậu – Queen.....	12
Quân Vua – King .....	14
c. Các tính năng của chương trình ứng dụng :.....	14
II. Cơ sở lý thuyết .....	15
a. Socket.....	15
b. TCP .....	16
c. UDP.....	17
<b>Chương 2. Phân tích thiết kế hệ thống .....</b>	<b>20</b>
I. Sơ đồ kiến trúc tổng quan hệ thống.....	20
II. Network stack.....	21
III. Sơ đồ use case .....	23
IV. Luồng của ứng dụng .....	27
<b>Chương 3. Hiện thực hệ thống .....</b>	<b>29</b>
I. Tính năng đăng nhập, đăng ký, quên mật khẩu và xác thực cho tính năng quên mật khẩu. ....	29
II. Sảnh chờ và các tính năng hiển thị thông tin. ....	32
III. Tính năng chat.....	33
<b>Chương 4. Kết luận, hướng phát triển .....</b>	<b>37</b>
<b>Bảng phân công công việc.....</b>	<b>38</b>
<b>Tài liệu tham khảo :.....</b>	<b>39</b>
<b><a href="https://vi.wikipedia.org/wiki/C%E1%BB%9D_vua">https://vi.wikipedia.org/wiki/C%E1%BB%9D_vua</a>.....</b>	<b>39</b>

## Chương 1: Tổng quan

### I. Giới thiệu đề tài (lý do chọn đề tài, giới thiệu về ứng dụng, nếu là game thì giới thiệu luật chơi, các tính năng của ứng dụng, game)

#### ❖ Lý do chọn đề tài

Cờ vua là một trò chơi trí tuệ mang tính đối kháng. Hiện đang phổ biến trên thế giới cũng như ở Việt Nam.

Trò chơi sử dụng một bàn cờ hình vuông chia thành 64 ô vuông nhỏ hơn với 8 hàng ngang và 8 hàng dọc. Mỗi người chơi sẽ bắt đầu với 16 quân cờ, bao gồm 8 con tốt, 2 mã, 2 tượng, 2 xe, 1 hậu và 1 vua. Mục tiêu của người chơi là cố gắng chiếu hết vua đối phương.

#### ❖ Luật chơi ( cách đi và ăn quân) :

Quân Trắng luôn luôn đi trước.

Trong phần này, chúng ta sẽ điểm qua các cách đi quân, ăn quân, các chiêu trò đặc biệt với các quân cờ. Xem cách di quân của từng quân trong cờ vua :

### Quân Tốt – Pawn



*The Pawn*

Quân Tốt trong cờ vua là quân thường xuyên bị ngó lơ nhất trong tất cả các quân cờ vua. Một phần vì quân tốt quá đơn giản và là quân cờ nhỏ nhất trong bộ cờ vua, sức mạnh ban đầu kém nhất và đứng ở hàng đầu. Mỗi người chơi cờ vua bắt đầu bằng việc tiến tốt, có 8 quân tốt trên bàn cờ để di chuyển.

#### *Làm thế nào để đi quân Tốt*

Quân Tốt là quân cờ đơn giản nhất và cũng phức tạp trong cách nó di chuyển.

Quân Tốt trong cờ vua có rất ít sự lựa chọn trong cách di chuyển. Chỉ có Tiến lên về phía trước cho đến khi nó đến cuối bàn cờ và được thăng hạng thành những quân cờ khác. Quân Tốt có cách di chuyển có thể mô tả ngắn gọn như sau:

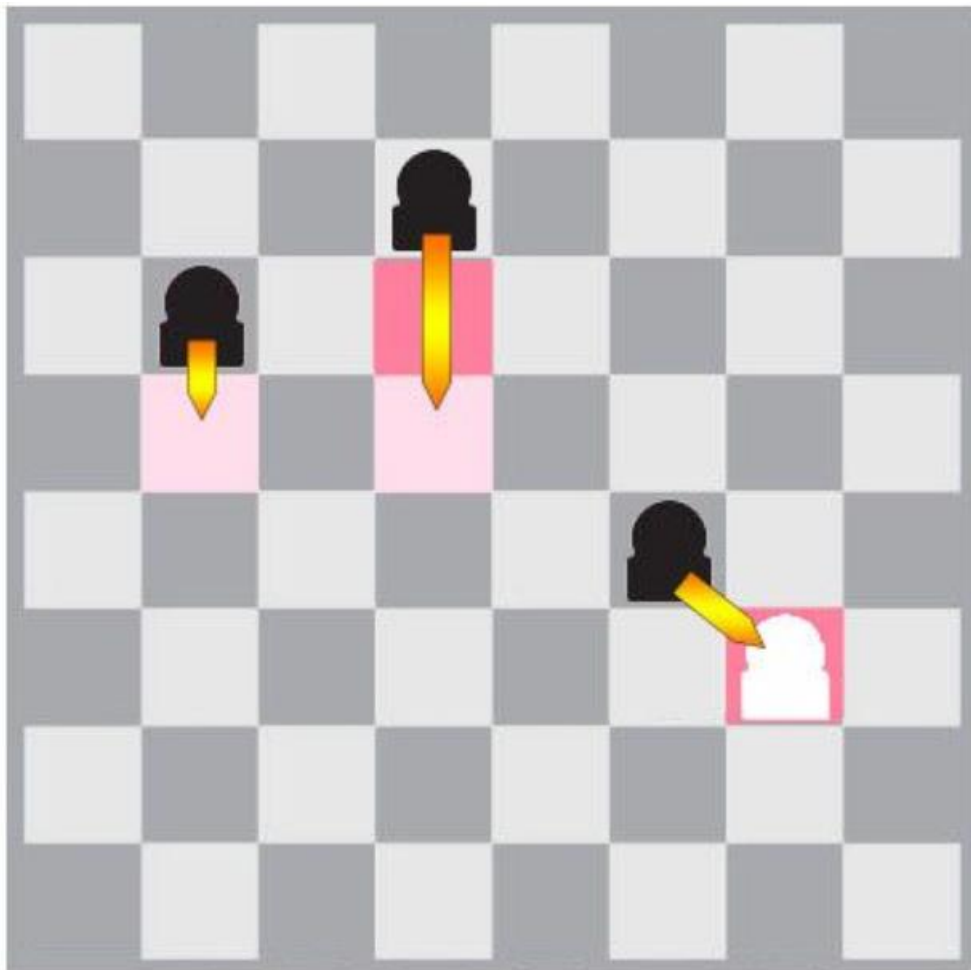
Quân tốt chỉ có thể di chuyển về phía trước 1 ô, trong số ít trường hợp khác là 2 ô.

Quân Tốt chỉ có thể tiến 2 ô trong lần di chuyển đầu tiên của nó.

Quân Tốt có thể tiến chéo để ăn quân đối thủ.

Khi quân Tốt tiến đến phía bên kia bàn cờ, nó được phong chức thành bất kỳ con nào khác nhưng không phải là vua.

Đụng Tốt ( Pawn Ram) là tình huống khi 2 Tốt của mình và đối thủ đụng nhau, làm cản đường di chuyển của cả hai. Để thoát khỏi đụng tốt, chúng ta có thể xử lý...



## Quân Xe – Chariot, Rook



## *The Rook*

Quân Xe trong cờ Vua có hình tượng như một tòa tháp, tương tự quân Xe trong cờ Tướng.

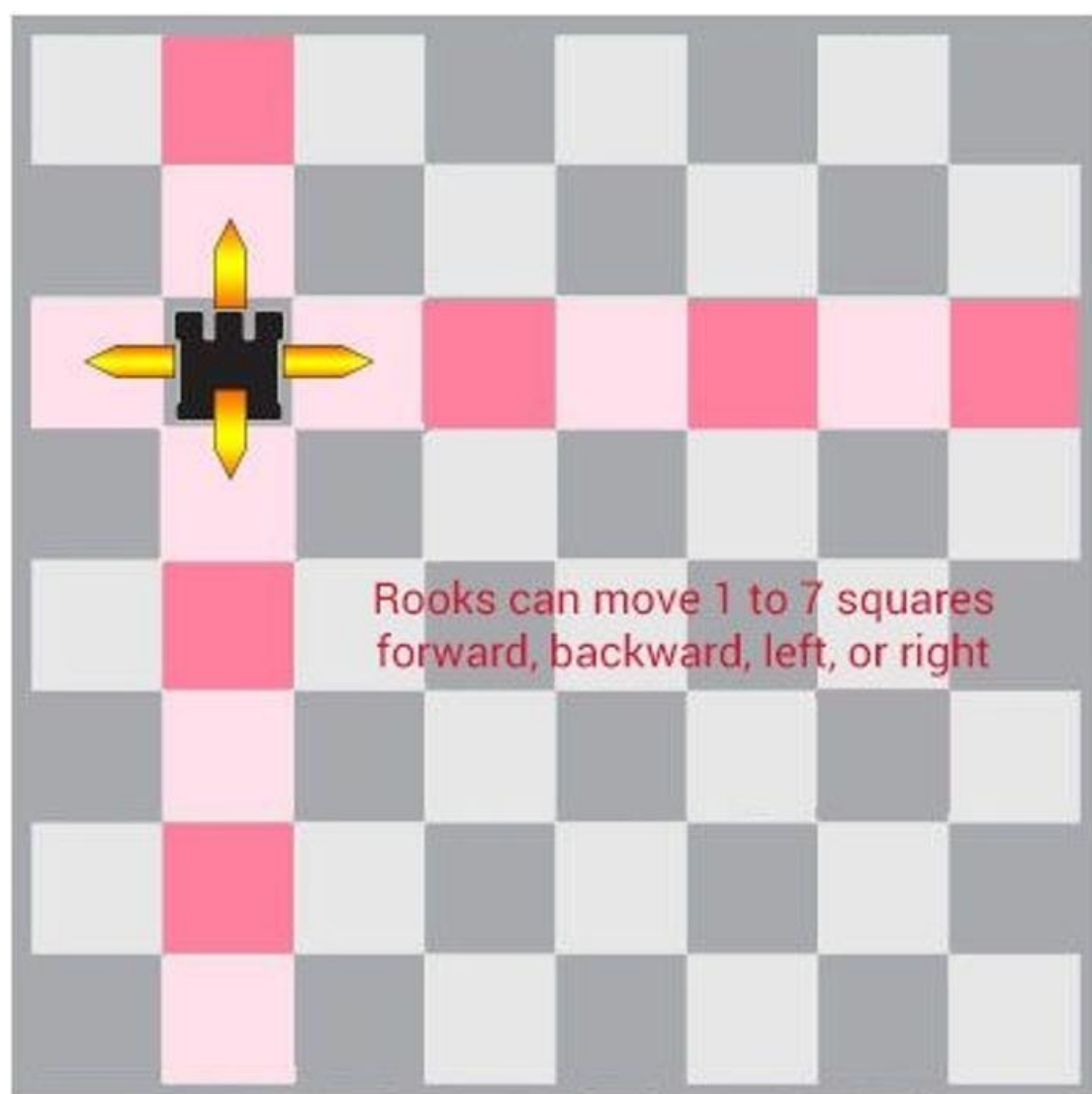
### ***Làm thế nào để đi quân Xe***

Quân Xe có cách di chuyển đơn giản nhất trên bàn cờ vua.

Quân Xe có thể di chuyển thẳng theo phương tiến, lùi, sang ngang bất kỳ lúc nào. Một lần di chuyển, quân Xe có thể thực hiện từ 1 đến 7 ô miễn là đường đi không bị cản bởi quân nào.

Sức mạnh của quân Xe vì vậy rất lớn trong trung cuộc và tàn cuộc, vì lúc đó số lượng quân của hai bên đã vơi bớt, ít cản đường xe hơn.

Trong cờ vua, quân Xe có thể di chuyển cùng lúc với quân Vua trong nước đi nhập thành (Castling).



## Quân Mã – Knight



## *The Knight*

Khi nói đến cờ vua, quân Mã thường được phân tích như là một quân có nhiều cơ hội sáng tạo nhất. Đây cũng là quân khó đoán nhất bàn cờ. Nhiều chiến thuật cờ vua cực kỳ độc đáo được tạo ra với quân Mã hạ gục đối phương thần sầu.

### ***Làm thế nào để đi quân Mã***

Quân Mã có thể di chuyển về phía trước, phía sau, trái hoặc phải hai ô vuông và sau đó phải di chuyển một hình vuông theo hướng vuông góc. Nói cách khác, quân Mã di chuyển theo hình chữ L

Quân Mã chỉ có thể di chuyển đến một trong tám vị trí trên bàn cờ vua. (Xem hình minh họa)

Quân Mã có thể đi đến bất kỳ vị trí không có quân cùng màu với nó.

Quân Mã có thể nhảy qua các quân khác.

Quân Mã thường được sử dụng trong vùng trung tâm bàn cờ vua vì nó là quân đầu tiên có thể tiến lên khu vực này.

Quân Mã có thể tấn công ăn quân khác mà không sợ bị ăn lại bởi quân khác (không phải quân Mã)

Quân Mã có cách di chuyển khác các quân khác, vì vậy quân Mã lấp lại yếu điểm của quân khác.





## Quân Tượng – Bishop



## The Bishop

Quân Tượng là một trong những quân mạnh trong bộ cờ Vua. Ở khai cuộc cờ vua, sức mạnh của Tượng có thể bị kiềm hãm vì quân Tốt chặn đường đi khá nhiều, nhưng một khi đường đi được giải phóng, quân Tượng sẽ tỏa sáng tốt. Quân Tượng luôn xuất sắc đỡ cho quân Vua nhưng nhiều cờ thủ vẫn thà thí Tượng để giữ Xe hơn.

### ***Làm thế nào để đi quân Tượng***

Quân Tượng di chuyển theo đường chéo theo bất kỳ hướng nào, miễn là không có quân cản trở.

Quân Tượng sẽ bị kẹt nếu có quân cản trên đường đi của nó, khác với quân Mã có thể nhảy qua quân khác.

Quân Tượng có thể ăn bất kỳ quân nào trên bước đi



A bishop can move in any direction diagonally so long as another piece isn't obstructing its path

## Quân Hậu – Queen



Quân Hậu được xem là quân cờ vua nguy hiểm và linh hoạt nhất trên bàn cờ vua, vì vậy quân hậu rất quan trọng để giữ trong cờ vua bên cạnh quân Vua. Mất quân Vua coi như thua cờ, nhưng mất quân Hậu thì cũng ... gần với thua cờ rồi trừ khi bạn là một tay cờ có nghề. Hầu hết mọi kỳ thủ đều sẵn lòng thí quân khác để cứu Hậu. Mỗi bên bắt đầu với một quân Hậu nhưng nếu di chuyển được quân Tốt đến cuối bàn cờ, thì nó sẽ biến thành quân hậu thứ hai. Đó là lý do tại sao một bộ cờ vua thường có 2 quân Hậu.

### ***Làm thế nào để đi quân Hậu***

Cách di chuyển của quân Hậu giống như cách đi của quân Xe và quân Tượng trong cờ vua.

Quân Hậu có thể di chuyển theo đường chéo và đường thẳng bất kỳ đâu.

Quân Hậu không thể nhảy qua quân khác.

Quân Hậu có thể ăn bất kỳ quân nào của đối thủ.

Quân Hậu là một quân phòng thủ ưa thích của nhiều kỳ thủ cờ vua vì khả năng di chuyển và ăn quân siêu phạm của nó. Quân Hậu là một nhân tố không thể thiếu trong tàn cuộc cờ vua. Tuy nhiên, tốt hơn là vẫn đẩy Hậu vào trung tâm bàn cờ để phòng thủ được tốt hơn phần bàn cờ của mình. Hậu là quân bảo vệ của quân Vua. Quân Mã của đối phương là mối đe dọa của quân Hậu, vì Hậu không có khả năng ăn Mã đang tấn công mình.



## Quân Vua – King



Quân cuối cùng là quân Vua, đây là quân cờ quan trọng nhất, bên nào có thể chiếu hết đường chạy của quân này thì đối phương sẽ thắng bất kể hơn thua về binh lính.

### ***Làm thế nào để đi quân Vua***

Quân Vua có thể di chuyển bất kỳ hướng nào 1 ô

Quân Vua không thể di chuyển đến quân cờ bên mình đang đứng.

Quân Vua không thể đi nước mà nó bị chiếu tướng

Quân vua có thể dùng trong thế nhập thành Castling là quân Vua có thể di chuyển 3 ô và đổi chỗ cho quân Xe.

Kỳ thủ hay có thể dùng quân Vua để nhử mồi đặt bẫy rồi ăn quân đối thủ, nhưng vẫn phải dùng vua sao cho an toàn.

Thông thường, để giữ an toàn cho quân Vua, kỳ thủ dồn Vua vào các góc để ít hướng tấn công nhắm đến được bằng Castling.

❖ Các tính năng của chương trình ứng dụng :

- Tính năng đăng nhập
- Tính năng quên mật khẩu (Đổi mật khẩu phòng trường hợp người dùng vô tình quên mật khẩu).
- Tính năng đăng kí
- Trong một ván đấu sẽ có các tính năng như sau :
  - Tương tác qua tin nhắn, gửi các icon emoji với đối phương.
  - Các tin nhắn và icon sẽ được hiển thị lên một tiện ích mở rộng Guna2.Panel.
  - Di chuyển các quân cờ dựa trên luật chơi cờ vua đã nêu ở phần hướng dẫn trên.
  - Thoát phòng chơi khi đã kết thúc game.
- Giao diện sảnh chờ ( Lobby interface ) :
  - Xem thông tin cá nhân
    - Chỉnh sửa thông tin cá nhân như ảnh Avatar, Username, Email.
  - Đăng xuất
  - Xem bảng xếp hạng các người chơi trong server
  - Xem danh sách bạn bè
  - Tạo phòng
  - Tham gia phòng ngẫu nhiên
  - Nhắn tin (text/icon) chat thế giới.
  - Xem lịch sử các trận đấu của người chơi.

## II. Cơ sở lý thuyết

1.

### a. Socket

#### ❖ Socket là gì?

- Socket là điểm cuối (end-point) trong liên kết truyền thông hai chiều (two-way communication) biểu diễn kết nối giữa Client – Server. Các lớp Socket được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.
- Socket là giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều, hay còn gọi là two-way communication để kết nối 2 process trò chuyện với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket.

#### ❖ Lập trình Socket là gì ?

- Lập trình socket (socket programming) là việc sử dụng các API (Application Programming Interface) hoặc thư viện socket có sẵn trong các ngôn ngữ lập trình như C/C++, Java, Python, C#, để xây dựng ứng dụng mạng có khả năng truyền thông qua socket. Điều này cho phép các ứng dụng giao tiếp, truyền dữ liệu và trao đổi thông tin qua mạng.

❖ **Tại sao người dùng lại cần đến Socket?**

- Ưu điểm lớn nhất của socket là hỗ trợ hầu hết các hệ điều hành bao gồm MS Windows, Linux,... Ngoài ra, socket cũng được sử dụng với nhiều ngôn ngữ lập trình, gồm C, C++, Java, Visual Basic, Visual C++,... nên nó có thể tương thích với hầu hết mọi đối tượng người dùng với những cấu hình máy khác nhau.
- Đặc biệt, người dùng cũng có thể chạy cùng một lúc nhiều socket liên tục, giúp nâng cao hiệu suất làm việc, cũng như tiết kiệm thêm nhiều thời gian và công sức hơn.

❖ **Socket hoạt động như thế nào?**

- Như đã đề cập trước đó, chức năng của socket là kết nối giữa client và server thông qua TCP/IP và UDP để truyền và nhận giữ liệu qua Internet. Giao diện lập trình ứng dụng mạng này chỉ có thể hoạt động khi đã có thông tin về thông số IP và số hiệu cổng của 2 ứng dụng cần trao đổi dữ liệu cho nhau.
- Hai ứng dụng cần truyền thông tin phải đáp ứng điều kiện sau thì socket mới có thể hoạt động:
  - Hai ứng dụng có thể nằm cùng trên một máy hoặc 2 máy khác nhau
  - Trong trường hợp 2 ứng dụng cùng nằm trên một máy, số hiệu cổng không được trùng nhau.

❖ **Phân loại Socket :**

**Socket** được chia làm 4 phân loại khác nhau :

1. Stream Socket ( hoạt động thông qua TCP )
2. Datagram Socket( hoạt động thông qua UDP )
3. Websocket
4. Unix socket

Tuy có 4 phân loại nhưng vì kiến thức và yêu cầu có hạn nên chúng ta sẽ chỉ tìm hiểu chính trên 2 phân loại TCP và UDP.

## **b. TCP**

❖ **Giao thức TCP/IP là gì?**



- TCP/IP (viết tắt của **Transmission Control Protocol (TCP)** và **Internet Protocol (IP)**) giao thức tiêu chuẩn trên Internet đảm bảo trao đổi thành công các gói dữ liệu giữa các thiết bị qua mạng. TCP là giao thức truyền tải cơ bản cho nhiều loại ứng dụng, bao gồm máy chủ web và trang web, ứng dụng email, FTP và các ứng dụng tương tự.. TCP/IP được đặt theo tên của 2 giao thức là giao thức điều khiển giao vận và giao thức liên mạng. Đây là 2 giao thức đầu tiên trên thế giới được định nghĩa. TCP/IP được phát triển vào năm 1978 bởi Bob Kahn và Vint Cerf.

❖ **Cách hoạt động của giao thức TCP?**

- TCP hoạt động theo tiến trình bắt tay 3 bước (3 way handshake). Tiến trình này hoạt động như sau:
  - Máy khách gửi cho máy chủ một gói SYN — một yêu cầu kết nối từ port nguồn của nó đến port đích đến của máy chủ.
  - Máy chủ phản hồi bằng gói SYN/ACK, xác nhận việc nhận được yêu cầu kết nối.
  - Máy khách nhận gói SYN/ACK và trả lời bằng gói ACK của chính nó.
- Sau khi kết nối được thiết lập, TCP hoạt động bằng cách chia nhỏ dữ liệu đã truyền thành các segment (phân đoạn), mỗi segment được đóng gói thành một gói dữ liệu và được gửi đến đích của nó.

### c. UDP

❖ **Giao thức UDP là gì?**

- UDP là một giải pháp được dùng thay cho giao thức TCP. Mặc dù đều hoạt động trên giao thức IP, nhưng UDP và TCP có những khác biệt rất quan trọng. Cụ thể:

❖ **Cách hoạt động của giao thức UDP?**

- Giao thức TCP hỗ trợ giao tiếp máy chủ tới máy chủ (host-to-host). Được coi là phương thức vận chuyển bảo mật đáng tin cậy nhưng có tốc độ truyền chậm hơn. TCP chia nhỏ các tập dữ liệu lớn thành các gói dữ liệu riêng lẻ, đảm bảo lưu lượng và kiểm soát lỗi, thực hiện gửi lại các tập tin bị mất và gửi theo đúng thứ tự.
- Giao thức UDP cho phép giao tiếp quy trình tới quy trình (process-to-process). UDP chỉ gửi gói tin được gọi là datagram, các gói tin này được truyền đi theo các đường dẫn khác nhau giữa người gửi và người nhận. Nghĩa là UDP có khả năng làm mất dữ liệu hoặc truyền dữ liệu không theo thứ tự, tuy nhiên lại sử dụng chi phí băng thông và độ trễ thấp hơn.

Tóm lại : Giao thức UDP hoạt động tương tự như TCP, nhưng nó bỏ qua quá trình kiểm tra lỗi. Khi một ứng dụng sử dụng giao thức UDP, các gói tin được gửi cho bên nhận và bên gửi không phải chờ để đảm bảo bên nhận đã nhận được gói tin, do đó nó

lại tiếp tục gửi gói tin tiếp theo. Nếu bên nhận bỏ lỡ một vài gói tin UDP, họ sẽ mất vì bên gửi không gửi lại chúng. Do đó thiết bị có thể giao tiếp nhanh hơn.

#### **d. Những công cụ, thư viện kèm theo:**

- **MongoDb:**

- **Khái niệm:** MongoDB là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) mã nguồn mở và phát triển bởi MongoDB Inc. Nó được thiết kế để lưu trữ và truy vấn dữ liệu theo mô hình tài liệu, trong đó dữ liệu được biểu diễn dưới dạng các tài liệu JSON (JavaScript Object Notation) linh hoạt.

- **Ưu điểm và nhược điểm của MongoDB:**

- **Ưu điểm của MongoDB:**

- **Cấu trúc dữ liệu linh hoạt:** MongoDB cho phép lưu trữ dữ liệu với cấu trúc không đồng nhất, có thể thay đổi linh hoạt theo thời gian. Điều này giúp dễ dàng thay đổi cấu trúc dữ liệu trong quá trình phát triển ứng dụng.
- **Khả năng mở rộng:** MongoDB hỗ trợ khả năng mở rộng ngang (horizontal scaling), cho phép chia nhỏ dữ liệu và phân phối trên nhiều máy chủ. Điều này giúp tăng hiệu suất và khả năng xử lý dữ liệu trong các ứng dụng có khối lượng lớn.
- **Tính linh hoạt và dễ sử dụng:** MongoDB cung cấp một ngôn ngữ truy vấn linh hoạt và dễ hiểu (MongoDB Query Language - MQL), giúp phát triển ứng dụng nhanh chóng.
- **Tích hợp tốt với các ngôn ngữ lập trình:** MongoDB hỗ trợ nhiều ngôn ngữ lập trình phổ biến như Python, JavaScript, Java, và .NET, giúp dễ dàng tích hợp vào các ứng dụng hiện có.

- **Nhược điểm của MongoDB:**

- **Không hỗ trợ các tính năng ACID đầy đủ:** MongoDB là hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL), không hỗ trợ các tính năng ACID (Atomicity, Consistency, Isolation, Durability) như các cơ sở dữ liệu SQL truyền thống. Làm giảm tính nhất quán và độ tin cậy của dữ liệu trong một số trường hợp đặc biệt.
- **Chi phí lưu trữ tăng lên với cấu trúc dữ liệu phức tạp:** Trong trường hợp dữ liệu có cấu trúc phức tạp hoặc liên kết mạnh mẽ, MongoDB có thể tốn nhiều không gian lưu trữ hơn so với cơ sở dữ liệu SQL truyền thống.

- Khả năng truy vấn phức tạp có hạn: Mặc dù MongoDB hỗ trợ các truy vấn linh hoạt, nhưng nó có thể gặp khó khăn trong việc thực hiện truy vấn phức tạp hơn.

- **Thư viện Guna Framework:**

- **Khái niệm:** GUNA Framework là một framework phát triển giao diện người dùng (UI) đa nền tảng và linh hoạt. Nó cung cấp công cụ và thành phần UI để tạo ra các ứng dụng đẹp và dễ sử dụng trên nhiều nền tảng như máy tính để bàn, di động và web.

- **Ưu điểm của Guna Framework:**

- **Hiệu suất cao:** Guna Framework được thiết kế để đạt hiệu suất cao với việc tối ưu hóa và sử dụng các thuật toán cắt cạnh, giúp xử lý dữ liệu và truy vấn nhanh chóng.
- **Đa nền tảng:** Guna Framework hỗ trợ phát triển ứng dụng trên nhiều nền tảng, bao gồm Web, iOS, Android và Node.js.
- **Dễ sử dụng:** Guna Framework có cú pháp dễ hiểu và học nhanh, giúp tiết kiệm thời gian và công sức trong quá trình phát triển ứng dụng.
- **Hỗ trợ cộng đồng:** Guna Framework có cộng đồng lớn và tài liệu phong phú, giúp bạn nhận được hỗ trợ và tìm hiểu từ người dùng khác.

- **Nhược điểm của Guna Framework:**

- **Hạn chế với ứng dụng lớn:** Guna Framework không phù hợp cho các ứng dụng lớn và phức tạp, vì có hạn chế về khả năng mở rộng và quản lý dữ liệu.
- **Tích hợp bên ngoài hạn chế:** Guna Framework có ít plugin và tích hợp bên ngoài hơn so với một số framework khác, làm khó khăn trong việc tích hợp công cụ và dịch vụ bên thứ ba.
- **Thời gian phát triển và cập nhật:** Sử dụng Guna Framework có thể đòi hỏi thời gian và công sức để tìm hiểu và áp dụng các tính năng mới nhất và cập nhật của framework.
- **Hạn chế về tài liệu:** So với một số framework phổ biến khác, Guna Framework có thể thiếu tài liệu chi tiết và hướng dẫn sử dụng cho các tính năng phức tạp và trường hợp sử dụng đặc biệt.

- **Thư viện Json:**

- **Khái niệm:** Thư viện JSON (JavaScript Object Notation) là một công cụ để xử lý dữ liệu trong định dạng JSON. JSON là một định dạng dữ liệu dựa trên văn bản phổ biến được sử dụng để truyền tải dữ liệu giữa các ứng dụng web.

- **Ưu điểm và nhược điểm của thư viện JSON:**

- **Ưu điểm của thư viện JSON:**

- **Đơn giản và dễ sử dụng:** Thư viện JSON cung cấp các phương thức và hàm dễ hiểu và dễ sử dụng, giúp xử lý dữ liệu JSON một cách đơn giản và thuận tiện.
- **Khả năng tương thích:** JSON là một định dạng dữ liệu phổ biến và được hỗ trợ rộng rãi trong hầu hết các ngôn ngữ lập trình và nền tảng.

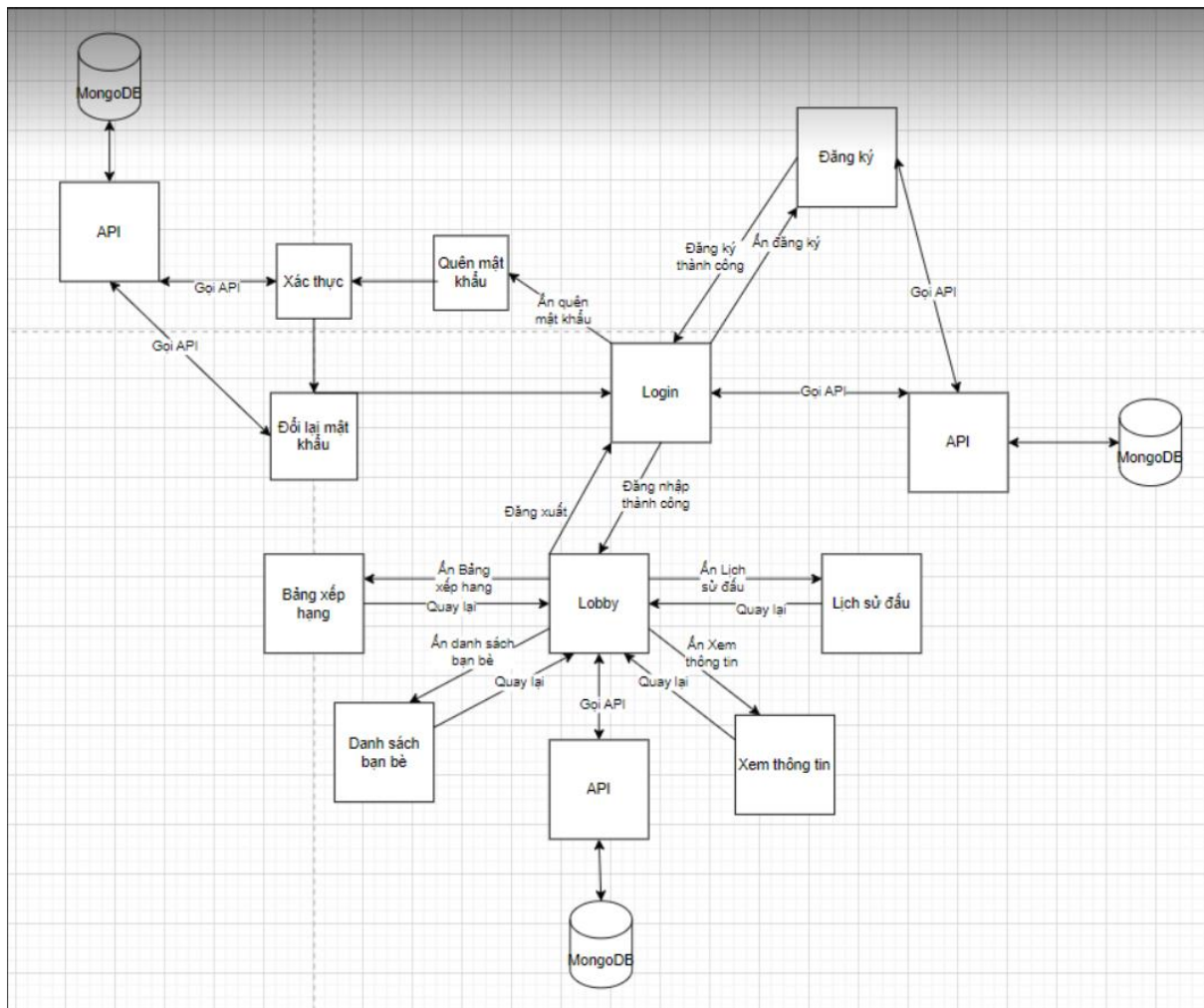
- Kích thước nhỏ: JSON có cấu trúc đơn giản và tối thiểu, dẫn đến kích thước dữ liệu nhỏ hơn so với các định dạng dữ liệu phức tạp hơn. Điều này giúp giảm tải băng thông và tăng hiệu suất truyền dữ liệu.
- Đọc và ghi dữ liệu dễ dàng: Thư viện JSON cung cấp các phương thức để dễ dàng đọc và ghi dữ liệu JSON từ và vào các nguồn dữ liệu như tệp tin hoặc API.

➤ **Nhược điểm của thư viện JSON:**

- Hạn chế trong xử lý dữ liệu phức tạp: JSON thường không hỗ trợ cấu trúc dữ liệu phức tạp và quan hệ, làm cho việc xử lý các truy vấn phức tạp trở nên khó khăn và tốn thời gian.
- Khả năng mở rộng hạn chế: JSON không cung cấp các tính năng mở rộng tự nhiên như các hệ quản trị cơ sở dữ liệu, ví dụ như các khóa ngoại,... Hạn chế trong việc xử lý dữ liệu lớn hoặc ứng dụng yêu cầu tính nhất quán cao.
- Thiếu kiểm soát dữ liệu: JSON không có các quy tắc nghiêm ngặt về kiểu dữ liệu, đảm bảo tính nhất quán và đúng đắn của dữ liệu. Điều này có thể dẫn đến các lỗi và vấn đề liên quan đến kiểu dữ liệu và kiểm soát dữ liệu.

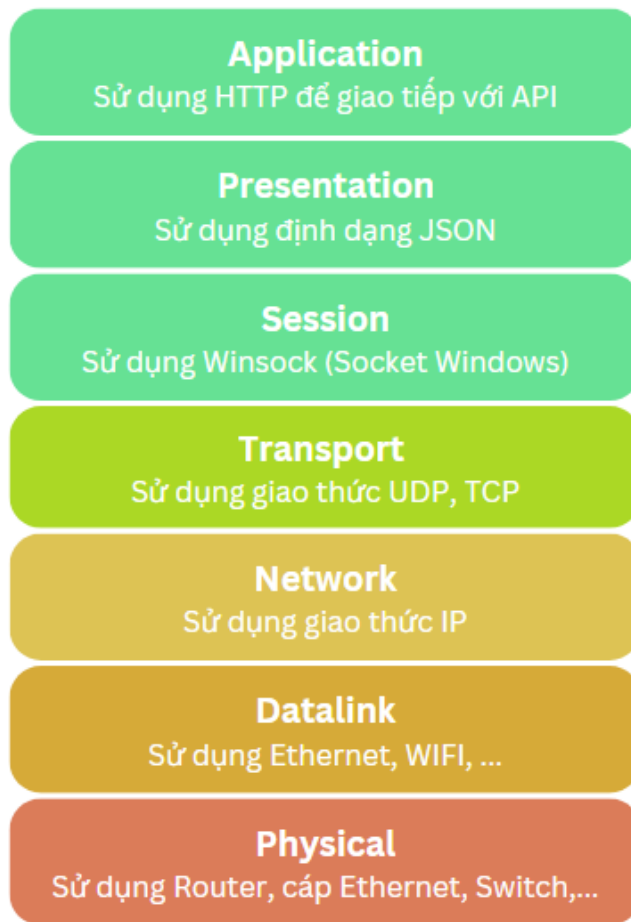
## **Chương 2. Phân tích thiết kế hệ thống**

### **I. Sơ đồ kiến trúc tổng quan hệ thống**



## II. Network stack

Mô hình OSI (Open Systems Interconnection) là một khung làm việc được sử dụng để mô tả và phân chia các chức năng của các giao thức mạng thành từng tầng riêng biệt. Trong ứng dụng ChessMates sử dụng Winform C#, chúng ta sẽ áp dụng mô hình này để xây dựng hệ thống mạng cho ứng dụng.



**Tầng Physical (Sử dụng Router, cáp Ethernet, Switch, ...):** Tầng Physical có trách nhiệm quản lý các thiết bị vật lý như router, cáp Ethernet,.. Các thiết bị này được sử dụng để kết nối mạng với nhau và định tuyến gói tin giữa chúng.

**Tầng Datalink (Sử dụng Ethernet, Wifi, ...):** Tầng Datalink đảm nhiệm việc truyền dữ liệu qua các phương tiện truyền thông vật lý như Ethernet, Wireless,... Đây đều là một trong những giao thức phổ biến được sử dụng để truyền dữ liệu giữa các thiết bị vật lý.

**Tầng Network: Sử dụng IP:** Tầng Network sử dụng giao thức IP (Internet Protocol) để định địa chỉ và định tuyến gói tin trong mạng. Trong ứng dụng này, IP được sử dụng để xác định địa chỉ mạng và điểm đích cho việc truyền dữ liệu.

**Tầng Transport (Sử dụng TCP và UDP):** Tầng Transport sử dụng giao thức TCP và UDP để thiết lập kết nối giữa các client và client hoặc giữa client và server với nhau.

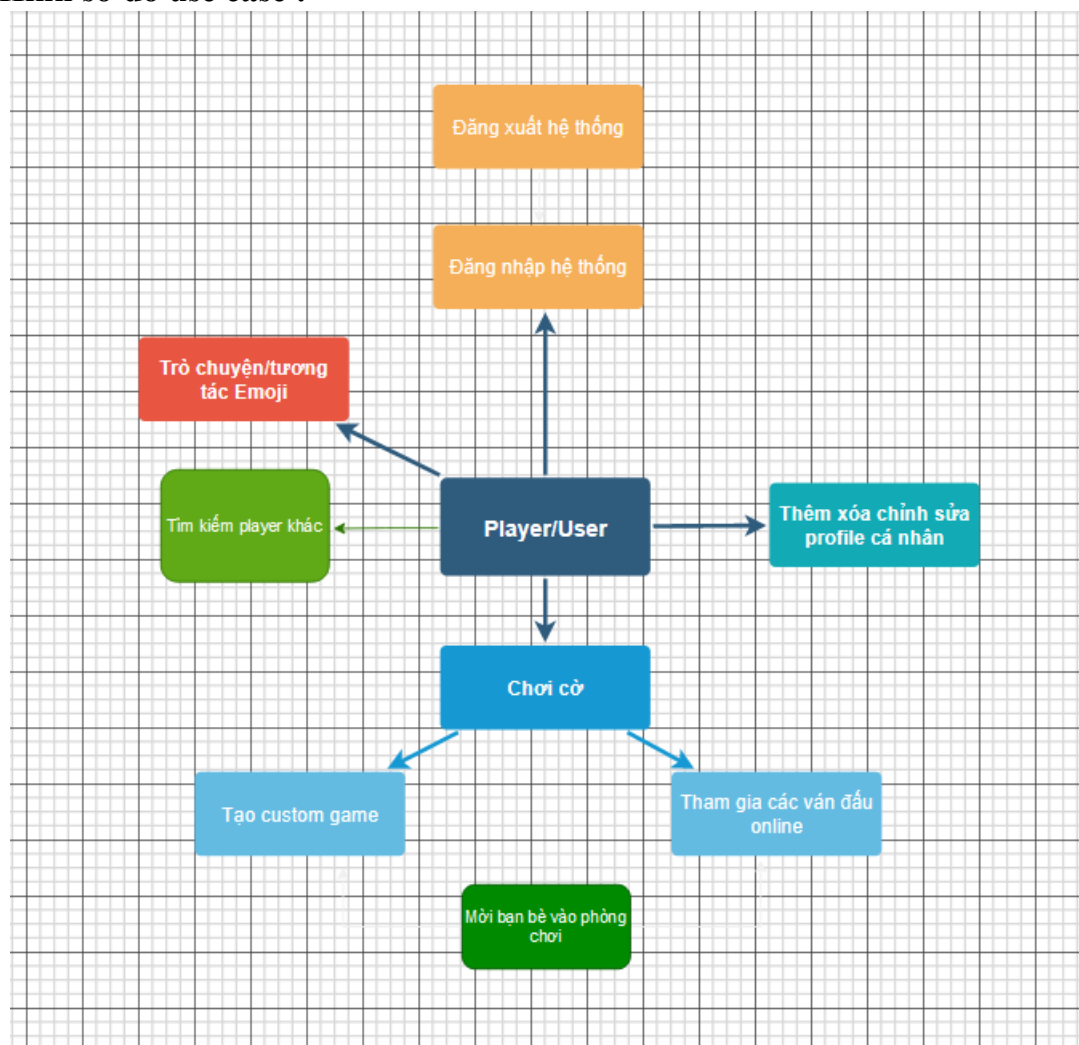
**Tầng Session (Sử dụng Winsock):** Tầng Session sử dụng Winsock để tạo và duy trì phiên truyền thông giữa các ứng dụng trên các thiết bị. Trong ứng dụng này, Winsock được sử dụng để quản lý việc kết nối và truyền dữ liệu giữa các client và server với nhau.

**Tầng Presentation (Sử dụng định dạng JSON):** Tầng Presentation đảm nhiệm việc trình bày dữ liệu để ứng dụng có thể hiểu được. Trong ứng dụng này, chúng ta sử dụng định dạng JSON (JavaScript Object Notation) để đại diện và truyền dữ liệu giữa các ứng dụng.

**Tầng Application (Sử dụng HTTP):** Tầng Application sử dụng HTTP để kết nối sử dụng API. Trong ứng dụng này, chúng ta sử dụng HTTP để truyền dữ liệu qua lại giữa client và server.

### III. Sơ đồ use case

❖ Hình sơ đồ use case :



❖ **ĐẶC TẢ USE CASE CHƠI CỜ VUA**

#### **1/ Use case Đăng nhập hệ thống**

- Actor : player/user

- Chức năng : Người chơi có nhu cầu đăng nhập để xác minh danh tính và tham gia vào hệ thống chơi cờ
- Các bước thực hiện :

+ ) Actor yêu cầu đăng nhập

+ ) Hệ thống hiển thị form yêu cầu nhập username và password

+ ) Actor nhập hoàn thiện form

+ ) Hệ thống redirect Actor vào giao diện menu của game

## **2/ Use case Đăng xuất hệ thống**

- Actor : Đối tượng đã hoàn thành đăng nhập thành công vào hệ thống
- Chức năng : Thoát khỏi hệ thống
- Các bước thực hiện :

+ ) Actor chọn thoát/đăng xuất

+ ) Hệ thống hiển thị messagebox xác nhận cho người chơi

+ ) Actor click vào nút xác nhận

+ ) Hệ thống redirect người chơi đến giao diện form Login game

## **3/ Use case chơi cờ**

- Actor : Player/User
- Chức năng : Ấn nút để bắt đầu lựa chọn mode chơi cờ theo ý người chơi
- Các bước thực hiện :

+ ) Actor ấn nút chơi

+ ) Hệ thống hiển thị các lựa chọn/mode chế độ chơi



+ ) Actor chọn và sẽ được chuyển tiếp đến mode chơi như ý

#### **4/ Use case Tạo custom game**

- Actor : Các đối tượng đã ấn nút chơi cờ
- Chức năng : Cung cấp chức năng tạo ván đấu theo ý user
- Các bước thực hiện :

+ ) Actor lựa chọn, chỉnh sửa các điều kiện, các luật lệ

+ ) Actor xác nhận tạo game

+ ) Hệ thống tạo và chuyển hướng Actor vào game vừa tạo.

#### **5/ Use case Tham gia game online**

- Actor : player/User
- Chức năng : Chơi, tham gia ván đấu với những người chơi khác qua mạng WLAN
- Các bước thực hiện :

+ ) Hệ thống xác nhận Actor có kết nối internet hay chưa! Nếu chưa có kết nối internet sẽ hiển thị MessageBox báo “Vui lòng kết nối mạng để chơi online!.”

+ ) Actor bắt đầu thực hiện tìm phòng chơi sau khi hệ thống đã xác nhận

+ ) Actor xác nhận vào phòng chơi đã chọn

+ ) Hệ thống chuyển hướng Actor vào phòng đã chọn

#### **6/ Use case Mời bạn bè**

- Actor : Đối tượng đã tham gia vào một ván đấu
- Chức năng : Có thể mời người bên ngoài vào phòng
- Các bước thực hiện :

- + ) Actor ấn chọn mời bạn bè
- + ) Hệ thống hiển thị box mời bạn bè có trong Friend List của người chơi
- + ) Actor chọn user muốn mời vào phòng
- + ) Hệ thống gửi lời mời đến đối tượng được chọn

## **7/ Use case tìm kiếm người chơi khác**

- Actor : Player/User
  - Chức năng : Tìm kiếm thông tin profile của người chơi qua tên hoặc userID của họ
  - Các bước thực hiện :
- + ) Actor chọn tìm kiếm
  - + ) Actor nhập thông tin đối tượng cần tìm vào
  - + ) Hệ thống hiển thị thông tin

## **8/ Use Case trò chuyện, tương tác Emoji**

- Actor : Player/User
  - Chức năng : Trò chuyện, gửi Emoji với bạn bè/với kênh chat thể giới/với đối thủ trong ván đấu
  - Các bước thực hiện :
- + ) Actor nhập nội dung vào chat box
  - + ) Hệ thống xem xét nội dung có chứa từ cấm hay không
  - + ) Hệ thống xác nhận nội dung và tiến hành gửi.

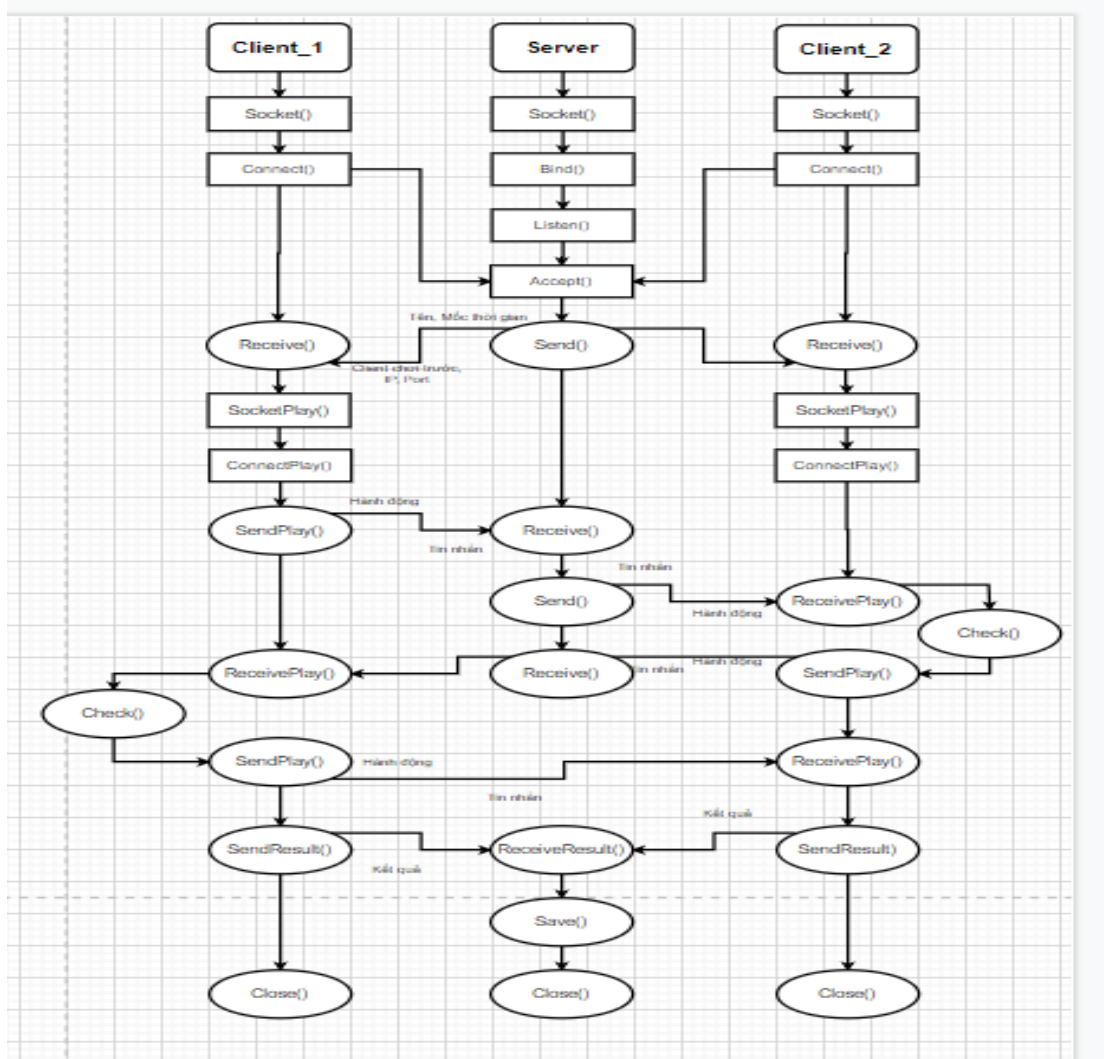
## **9/ Use case thêm, xóa, sửa profile**

- Actor : player/user

- Chức năng : Thêm xóa sửa thông tin cá nhân
- Các bước thực hiện :
  - +) Actor chọn vào mục hiển thị thông tin cá nhân
  - +) Hệ thống hiển thị thông tin người chơi
  - +) Actor chọn option chỉnh sửa profile
  - +) Hệ thống cho phép chỉnh sửa thông tin
  - +) Actor chỉnh sửa/xóa/thêm tùy ý
  - +) Actor xác nhận thông tin đã chỉnh sửa hoàn tất
  - +) Hệ thống lưu lại thay đổi.

#### **IV. Luồng của ứng dụng**

- ❖ Mô hình luồng chat vẽ trên Draw.io :



### ❖ Giải thích mô hình :

#### ➤ Máy Server :

- Trong mô hình này, máy chủ sẽ tạo và Bind() một địa chỉ IP và port cụ thể đến.
- Socket sau đó sẽ lắng nghe kết nối đến từ client dùng phương thức listen().
- Khi có kết nối đến máy chủ, máy chủ chấp nhận kết nối bằng phương thức Accept() và tạo một Socket() quản lý client đó.
- Thực hiện kết nối thành công. Lúc này máy chủ và máy khách giao tiếp thông qua phương thức Send() và Receive() dùng để gửi và nhận thông tin.

#### ➤ Máy client :

- Máy khách tạo một socket.
- Kết nối socket đến địa chỉ IP và port của máy chủ dùng phương thức Connect().
- Máy chủ lúc này nếu chấp nhận kết nối từ máy khách thì hai bên sẽ có thể trao đổi thông tin với nhau. Đồng thời khi 2 clients muốn trao đổi thông tin với nhau, máy chủ lúc này sẽ làm trung

gian thực hiện nhận thông tin từ client này và gửi tin nhắn đó đến client ở đích.

## Chương 3. Hiện thực hệ thống

### I. Tính năng đăng nhập, đăng ký, quên mật khẩu và xác thực cho tính năng quên mật khẩu.



The image shows a user interface for a login and registration system. It features a light orange background with a chess knight icon in a circle at the top. Below the icon, there are two input fields: 'Username' with a placeholder 'Nhập tài khoản' and 'Password' with a placeholder 'Nhập mật khẩu'. To the left of the password field is a lock icon. Below the password field is a label 'Thông báo lỗi'. At the bottom, there are two buttons: 'Đăng nhập' (Login) and 'Đăng ký' (Register). Below the buttons is a link 'Quên mật khẩu' (Forgot password).

Button Đăng nhập: Chức năng cho phép người dùng đăng nhập vào hệ thống bằng tài khoản gồm: username và password do người dùng đăng ký khi khởi tạo tài khoản.

Label thông báo lỗi giúp người dùng biết được tài khoản đã bị nhập sai và cần phải nhập lại, khi nhập sai đến số lần cụ thể, người dùng sẽ phải chờ một khoảng thời gian để có thể đăng nhập lại.

Button Đăng ký chuyển hướng người dùng đến form đăng ký để khởi tạo tài khoản.

Label Quên mật khẩu hỗ trợ người dùng lấy lại mật khẩu trong trường hợp người dùng không nhớ được mật khẩu của mình.

# ĐĂNG KÝ

**Username**



**Email**



**Mật khẩu**



**Xác nhận mật khẩu**



**Quay lại**

**Đăng ký**

Người dùng nhập username, email, mật khẩu và xác nhận mật khẩu vào các Textbox trong form đăng ký.

Nếu người dùng nhập vào email không hợp lệ, chương trình sẽ hiển thị thông báo bên dưới textbox nhập email là “Email” không hợp lệ.

Mật khẩu hợp lệ phải có độ dài từ 6 đến 30 ký tự, phải chứa số, chữ thường và chữ hoa.

Sau khi ấn đăng ký hệ thống sẽ tạo tài khoản cho người dùng sử dụng, đăng nhập bằng tài khoản đó để vào trò chơi.

A login form with an orange background. It features a 'Username' label with a person icon and a text input field containing the placeholder 'Nhập tên tài khoản'. Below it is an 'Email' label with an envelope icon and a text input field containing the placeholder 'Nhập email'. At the bottom, there are two buttons: a dark brown 'Xác nhận' button and a red 'Quay lại' link with a dashed underline.

Khi ấn vào quên mật khẩu ở form Đăng nhập, người dùng sẽ được chuyển hướng đến form Quên mật khẩu. Tại đây người dùng cần cung cấp username và email của mình để thực hiện việc lấy lại mật khẩu.

Khi nhấn nút Xác nhận, người dùng sẽ được điều hướng đến form xác thực.

Nhấn label Quay lại để quay về form đăng nhập.

A verification form with an orange background. It has a label 'Nhập mã xác thực' above a text input field with the placeholder 'Nhập mã'. Below the input field is a link 'Gửi lại mã' with a dashed underline. At the bottom is a dark brown 'Xác nhận' button.

Sau khi ấn Xác nhận ở Form Quên mật khẩu, người dùng sẽ được thông báo rằng “Vui lòng kiểm tra hộp thư để lấy mã” và chuyển hướng đến form xác thực để tiến hành xác thực xem có đúng username và email hay chưa.

Nhấn vào label Gửi lại mã để gửi lại mã xác thực trong trường hợp không nhận được mail chứa mã xác thực.

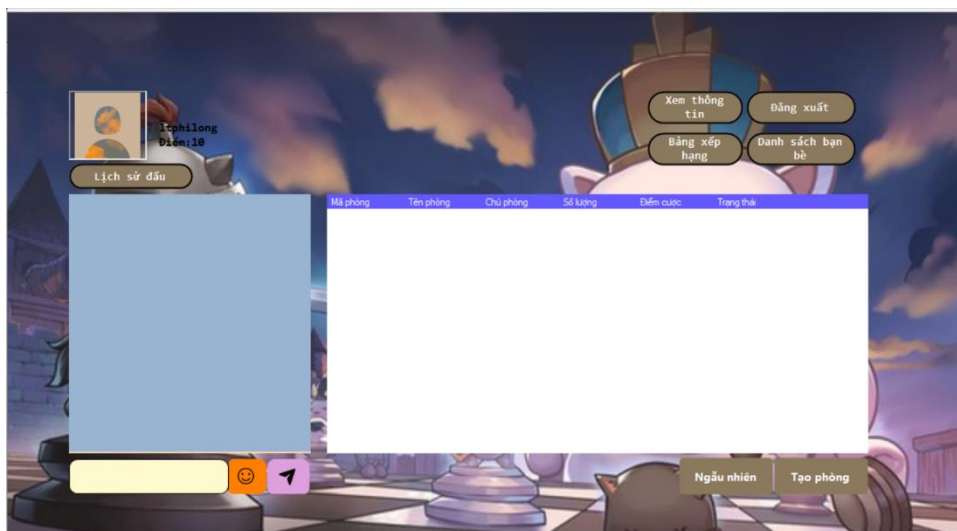
Nhấn vào nút Xác nhận để hệ thống kiểm tra xem mã xác thực có đúng hay không, nếu đúng thì chương trình sẽ thực hiện bước kế tiếp, nếu sai chương trình sẽ yêu cầu nhập lại.



The image shows a user interface for changing a password. It has a light orange background. At the top, the text "Mật khẩu mới" (New password) is displayed. Below it is a lock icon and a text input field with the placeholder "Nhập mật khẩu mới". Underneath that is the text "Xác nhận mật khẩu" (Confirm password). Below this is another lock icon with a checkmark and a text input field with the placeholder "Xác nhận mật khẩu". At the bottom center is a dark orange button with the text "Xác nhận" (Confirm).

Tại đây người chơi sẽ được việc cập nhật lại mật khẩu. Sau khi nhấn nút xác nhận hệ thống tài khoản sẽ thay đổi mật khẩu cũ thành mật khẩu mới.

## II. Sảnh chờ và các tính năng hiển thị thông tin.



Sau khi đăng nhập, người dùng sẽ được chuyển hướng đến sảnh chờ.



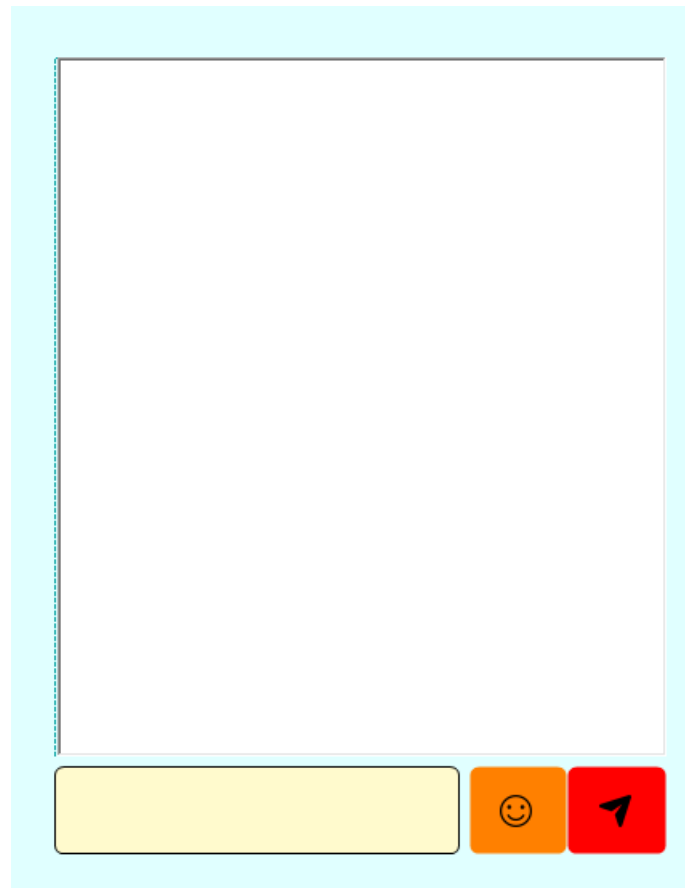
Tại đây sẽ có các button Xem lịch sử đấu, Xem thông tin, Đăng xuất, Bảng xếp hạng, Danh sách bạn bè, tạo phòng Ngẫu nhiên, Tạo phòng và khung chat với tất cả client, Datagridview để hiển thị danh sách phòng chơi.

### III. Tính năng chat

- Chia làm 3 phần chính : Chat với bạn bè (1-1) , chat ở sảnh chờ và chat với đối phương trong phòng chơi.

#### ❖ Chat phòng chơi :

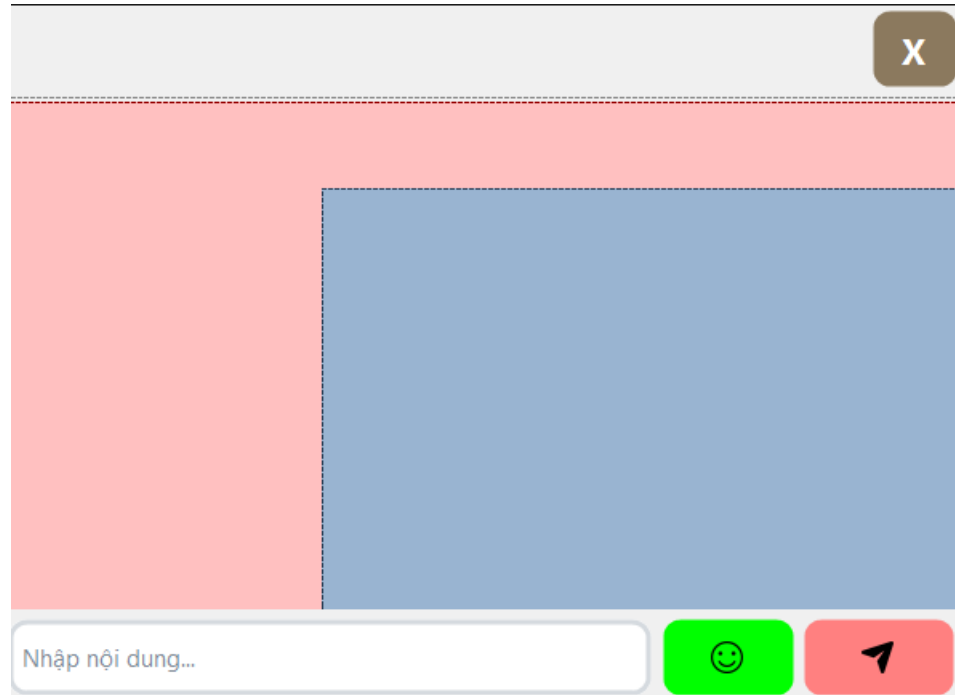
- Tổng quan cách hoạt động :
  - Sau khi 2 người chơi tham vào một ván cờ, hệ thống sẽ kết nối 2 users này lại với nhau dựa trên IPv4 Address lấy được nhờ sử dụng lớp NetworkInterface.
  - Thực hiện gửi/nhận tin nhắn, icon dựa trên giao thức UDP.
  - Thiết kế :



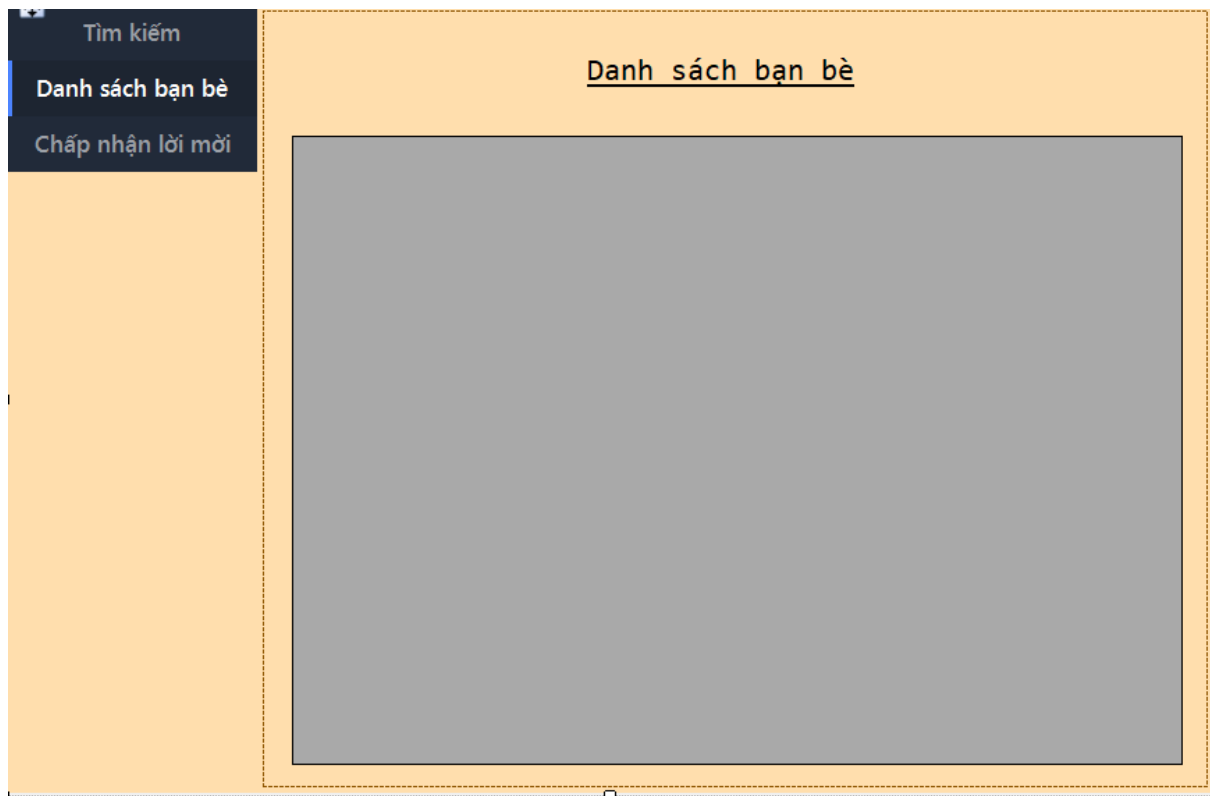
- Button gửi : Nhiệm vụ gửi tin nhắn
- Button icon : Nhiệm vụ hiển thị panel cho người dùng chọn.
- Listchat : hiển thị nội dung tin nhắn giữa 2 người chơi.
- Textbox : Nhập tin nhắn

### ❖ Chat riêng :

- Tổng quan cách hoạt động :
  - Chọn 1 người chơi là bạn bè của mình được hiển thị trong danh sách bạn bè. Hệ thống sau đó sẽ kết nối 2 người chơi qua IP. Hai người chơi sau đó thực hiện chat dựa trên giao thức TCP.
- Thiết kế :



- Textbox : Nhập tin nhắn
- Button gửi : Nhiệm vụ gửi tin nhắn
- Button icon : Nhiệm vụ hiển thị panel cho người dùng chọn.
- Winform panel ( màu xanh) : Dùng để hiển thị icon khi các user tương tác với nhau.
- Guna2.Panel ( màu hồng ) : Hiển thị khung chat giữa 2 người chơi.



- Guna2.TabControl : Hiển thị hộp thoại các tab all user bao gồm tính năng tìm kiếm, friendlist, Accept invite
- DataGridView : Hiển thị danh sách bạn bè

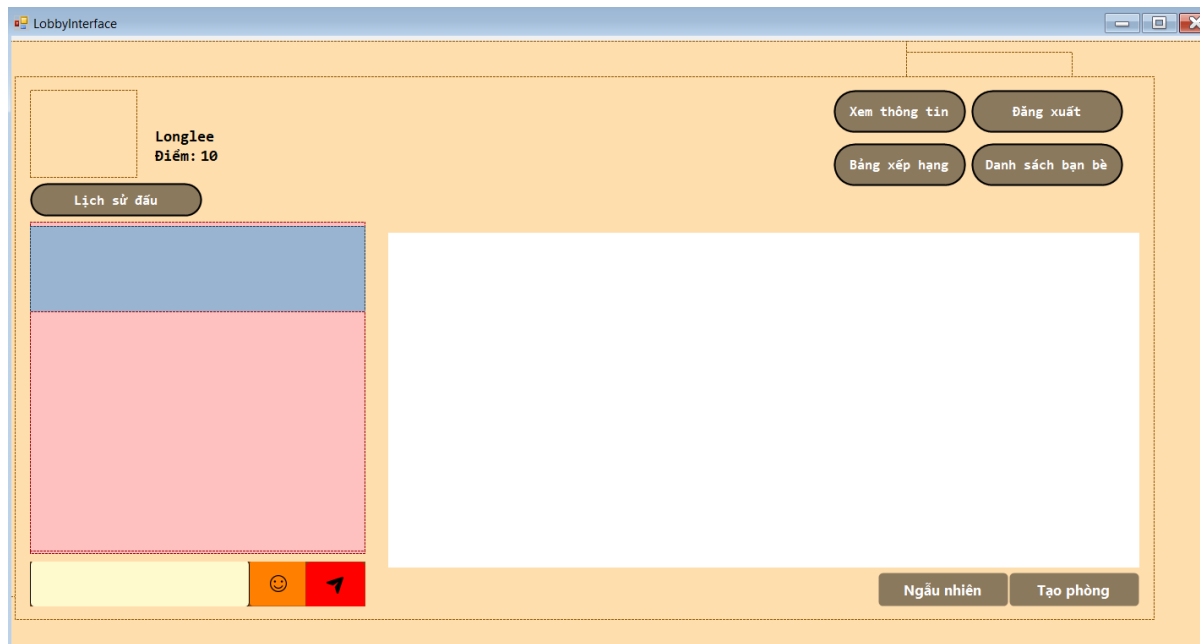
#### ❖ Chat ngoài sảnh chờ :

Tổng quan cách hoạt động :

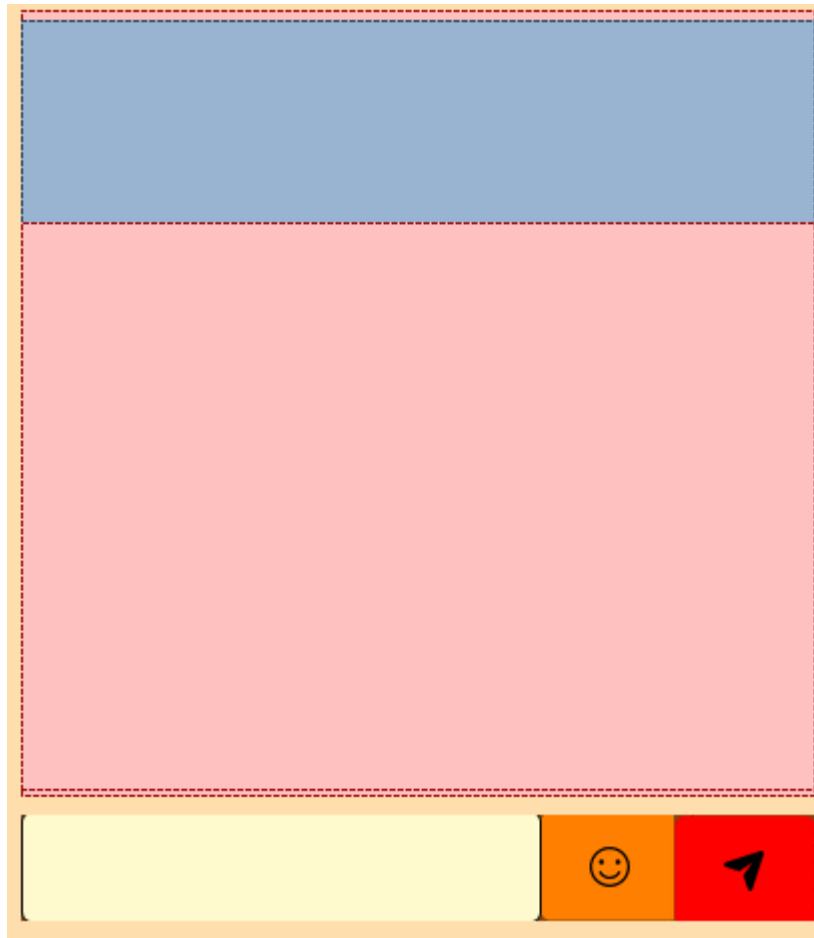
- Mở Server.
- Trong phương thức khởi tạo constructor của giao diện sảnh ( Lobby Interface ), các clients khởi tạo phương thức TcpClient() rồi connect kết nối đến IPAddress của máy Server. Đồng thời tạo một luồng để lắng nghe message từ các người chơi khác. Cho luồng bắt đầu listen lắng nghe với phương thức Start().
- Sau khi hoàn tất kết nối, các clients sẽ có thể tương tác (gửi, nhận) tin nhắn và icon với nhau thông qua khung chat.

#### Giao diện của sảnh :

Hình design lobby Interface :



- Button “Xem thông tin” cho phép xem thông tin người dùng
- Button “Đăng xuất” cho phép người dùng chuyển hướng đến form Đăng nhập và đăng nhập bằng 1 tài khoản khác
- Button “Bảng xếp hạng” cho phép người dùng xem thứ hạng của mình và người khác dựa trên số điểm có được
- Button “Danh sách bạn bè” cho phép người dùng kết bạn và tìm kiếm bạn bè
- Button “Ngẫu nhiên” cho phép người dùng tham gia phòng đấu một cách ngẫu nhiên



- Button “SendMessage” : Có nhiệm vụ gửi tin nhắn đi sau khi đã mã hóa.
- Button “sendIcon” : Đảm nhiệm hiển thị panel để chọn icon và gửi icon đó.
- Textbox : Nhập tin nhắn muốn gửi.
- Guna2Panel “MultichatFrame” ( màu xanh ) : Dùng để hiển thị nội dung tin nhắn cũng như các icon khi các user tương tác với nhau.
- WindowsForm.Panel ( màu hồng ) : Khi Click vào button gửi icon thì panel bảng này sẽ hiện lên các icon cho người dùng chọn.

## Chương 4. Kết luận, hướng phát triển

### 4.1/ Những thứ đã thực hiện được:

- Thiết kế được các form như đăng ký, đăng nhập, nhập mã xác thực, quên mật khẩu, sảnh chờ, thông tin người chơi, trận đấu.
- Giao diện bắt mắt, thân thiện với người dùng.
- Hiểu biết nhiều hơn về ngôn ngữ lập trình c#, hiểu rõ hơn về Thread và các thư viện hỗ trợ.
- Biết cách xây dựng cơ sở dữ liệu một cách hợp lý.

#### 4.2/ Về hạn chế:

- Về mặt mã nguồn thì chưa thật sự tối ưu, code còn quá lộn xộn và gây chịu.
- Các tính năng đôi khi vẫn có lỗi xảy ra trong quá trình hoạt động.
- Về hiển thị thông báo thì còn quá sơ sài vì chưa áp dụng animation vào.
- Đôi khi chương trình vẫn bị giựt lag do việc tổ chức gọi api chưa thực sự hoàn hảo.
- Đây là lần đầu nhóm cùng nhau phát triển một dự án nên đôi lúc sẽ không hiểu ý nhau.

#### 4.3/ Hướng phát triển:

- Làm cho chương trình hoạt động một cách mượt mà hơn, giảm thiểu tình trạng lỗi ít nhất có thể.
- Tìm tòi và bổ sung thêm nhiều tính năng mới mẻ hơn.
- Tổ chức và quản lý source code một cách tối ưu hơn.
- Tăng tính bảo mật cho dữ liệu.

**Bảng phân công công việc**

STT	MSSV	Tên	Nội dung công việc phụ trách	Mức độ hoàn thành	Tự đánh giá (theo thang điểm 10)
1	21522302	Lê Trần Phi Long	Xây dựng API và thuật toán game cờ vua	100%	10
2	21522143	Hứa Hồ Gia Huy	Chat gửi tin nhắn dùng giao thức TCP & UDP. Thiết kế giao diện đồ họa, hiển thị danh sách bạn bè, tất cả người chơi.	100%	10
3	21521959	Dương Viết Đức	Login, Signin, xác nhận tài khoản, gửi mã, thay đổi mật khẩu	100%	10
4	21521868	Trương Chí Bảo	Thiết kế giao diện, hiển thị lịch sử đấu, thông tin người chơi, thứ hạng, tham gia phòng chơi, hiển thị	100%	10

			danh sách phòng đấu.		
--	--	--	-------------------------	--	--

### **Tài liệu tham khảo :**

1. [https://vi.wikipedia.org/wiki/C%E1%BB%9D\\_vua](https://vi.wikipedia.org/wiki/C%E1%BB%9D_vua)
2. <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-7.0>
3. <https://www.codeproject.com/Articles/36112/Chess-Program-in-C>

### **Slide thuyết trình:**

[https://www.canva.com/design/DAFk-K\\_8efY/YsE3zmIWXPPjh62W-SUB4g/edit?fbclid=IwAR38wYFkGa0fUMgkhE6hzC36elWTZeoHLY\\_lWqxAnXD5e1pyrxl90k05Zg](https://www.canva.com/design/DAFk-K_8efY/YsE3zmIWXPPjh62W-SUB4g/edit?fbclid=IwAR38wYFkGa0fUMgkhE6hzC36elWTZeoHLY_lWqxAnXD5e1pyrxl90k05Zg)

### **Video demo:**

[https://drive.google.com/file/d/1NeGXEj32ledTg-nXDlCNSXSEfpIoyz\\_I/view?fbclid=IwAR2UsfVZhVFXuUHoXp45QrrrZZDcKC\\_o-fcjFdXTk80i\\_9SHj4idbi-vb4Y](https://drive.google.com/file/d/1NeGXEj32ledTg-nXDlCNSXSEfpIoyz_I/view?fbclid=IwAR2UsfVZhVFXuUHoXp45QrrrZZDcKC_o-fcjFdXTk80i_9SHj4idbi-vb4Y)