

**TRƯỜNG ĐẠI HỌC HOA SEN**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🕮🙢

**BÁO CÁO CUỐI KỲ**

ĐỀ TÀI:

XÂY DỰNG GAME CARO

Môn học: Phát Triển Phần Mềm Ứng Dụng

GV: Trần Thị Trương Thi

SV Thực hiện:

* Nguyễn Văn Tân
* Nguyễn Phương Khoa

Lớp: TIN340DV01 – 0100-0101

HO CHI MINH CITY, 2021

MỤC LỤC

[DANH MỤC HÌNH 3](#_Toc92439609)

[DANH MỤC USECASE DIAGRAM 4](#_Toc92439610)

[DANH MỤC THUẬT TOÁN 6](#_Toc92439611)

[CHƯƠNG 1: DẪN NHẬP 7](#_Toc92439612)

[1.1. Giới thiệu đề tài 7](#_Toc92439613)

[1.2. Mục đích đề tài 7](#_Toc92439614)

[1.3. Giới hạn đề tài 7](#_Toc92439615)

[CHƯƠNG 2: GIỚI THIỆU CÔNG NGHỆ 8](#_Toc92439616)

[2.1. Ngôn ngữ C# 8](#_Toc92439617)

[2.2. Winform 8](#_Toc92439618)

[2.3. Telerik UI For Winform 8](#_Toc92439619)

[2.4. Bunifu UI For Winform 8](#_Toc92439620)

[CHƯƠNG 3: MÔ TẢ YÊU CẦU PHẦN MỀM 10](#_Toc92439621)

[3.2. Firebase 10](#_Toc92439622)

[3.2. Sử dụng ngôn ngữ C# 10](#_Toc92439623)

[3.3. UI Framework 10](#_Toc92439624)

[CHƯƠNG 4: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 11](#_Toc92439625)

[4.1. Usecase Hệ Thống 11](#_Toc92439626)

[*4.1.1. Usecae User* 11](#_Toc92439627)

[4.2. Usecase các chức năng 12](#_Toc92439628)

[*4.2.1. Usecase chế độ chơi với người* 12](#_Toc92439629)

[*4.2.2. Chế độ dánh với máy* 12](#_Toc92439630)

[4.3. Class Diagram 13](#_Toc92439631)

[4.4. Sequence Diagram 14](#_Toc92439632)

[CHƯƠNG 3: THIẾT KẾ GIAO DIỆN HỆ THỐNG 19](#_Toc92439633)

[3.1. Màn hình cho User 19](#_Toc92439634)

[*3.1.1. Màn hình chờ* 19](#_Toc92439635)

[*3.1.2. Màn hình đăng ký* 19](#_Toc92439636)

[*3.1.3. Màn hình đăng nhập* 20](#_Toc92439637)

[*3.1.4. Màn hình trang chủ* 20](#_Toc92439638)

[*3.1.5. Màn hình thông tin người chơi* 21](#_Toc92439639)

[*3.1.6. Màn hình trang chủ game caro* 21](#_Toc92439640)

[*3.1.7. Màn hình trang Send Email* 22](#_Toc92439641)

[*3.1.8. Màn Player vs Player* 22](#_Toc92439642)

[*3.1.9. Màn hình Player vs Pc* 23](#_Toc92439643)

[CHƯƠNG 4: MÔ TẢ VÀ THIẾT KẾ MỘT SỐ THUẬT TOÁN (VẼ FLOWCHART) 24](#_Toc92439644)

[4.1. Login 24](#_Toc92439645)

[4.2 Connect mạng LAN 25](#_Toc92439646)

[4.3. Thuật toán đánh với máy 26](#_Toc92439647)

[CHƯƠNG 5: HƯỚNG PHÁT TRIỂN VÀ KẾT QUẢ ĐỀ TÀI, HẠN CHẾ 32](#_Toc92439648)

[9.1. Kết quả đạt được 32](#_Toc92439649)

[9.2. Phát triển ứng dụng 32](#_Toc92439650)

[9.3. Hạn chế 32](#_Toc92439651)

[TÀI LIỆU THAM KHẢO 33](#_Toc92439652)

# DANH MỤC HÌNH

[Giao diện 1: Màn hình chờ 6](#_Toc90122109)

[Giao diện 2: Màn hình đăng ký 6](#_Toc90122110)

[Giao diện 3: Màn hình đăng nhập 7](#_Toc90122111)

[Giao diện 4: Màn hình trang chủ 7](#_Toc90122112)

[Giao diện 5: Detail Profile 8](#_Toc90122113)

[Giao diện 6: Màn hình Game Caro 8](#_Toc90122114)

[Giao diện 7: Trang hướng dẫn 9](#_Toc90122115)

[Giao diện 8: Màn hình chơi game 9](#_Toc90122116)

[Giao diện 9: Màn hình Add User 10](#_Toc90122117)

[Giao diện 10: Màn hình View User 10](#_Toc90122118)

# DANH MỤC USECASE DIAGRAM

[UsecaseDiagram 1: User 12](#_Toc92384966)

[UsecaseDiagram 2: Admin 12](#_Toc92384967)

[UsecaseDiagram 3: Chế độ chơi với người 13](#_Toc92384968)

[UsecaseDiagram 4: Play vs PC 13](#_Toc92384969)

# 

**DANH MỤC SEQUEN DIAGRAM**

[Sequence Diagram 1: Login 11](#_Toc90570406)

[Sequence Diagram 2: Register 11](#_Toc90570407)

[Sequence Diagram 3:Dashboard 12](#_Toc90570408)

[Sequence Diagram 4: Detail Profile 12](#_Toc90570409)

[Sequence Diagram 5:List Game 13](#_Toc90570410)

[Sequence Diagram 6: Enter GameCaro 13](#_Toc90570411)

[Sequence Diagram 7: How to play 14](#_Toc90570412)

[Sequence Diagram 8: Connect Lan 14](#_Toc90570413)

[Sequence Diagram 9: New Game 15](#_Toc90570414)

[Sequence Diagram 10: Close 15](#_Toc90570415)

# DANH MỤC THUẬT TOÁN

[Thuật toán 1: Login 23](#_Toc92438398)

[Thuật toán 2: Connect Lan 24](#_Toc92438399)

[Thuật toán 3: Các ô [i,j] được duyệt (màu vàng) xung quanh ô [x,y] 25](#_Toc92438400)

# CHƯƠNG 1: DẪN NHẬP

## *1.1. Giới thiệu đề tài*

Game caro là gì? Là một loại game đối kháng gồm có hai người chơi trên một màn hình có kẻ các ô vuông đều nhau và người chơi đánh các ký tự O---X (tùy quy định tùng người).Trò chơi gồm hai người, lần lượt mỗi bên đánh dấu O-X. Nếu mỗi bên đạt được năm dấu chung hàng (có thể là hàng ngang, hàng dọc, hàng chéo) người chơi đó sẽ thắng.

## *1.2. Mục đích đề tài*

Để làm quen với ngôn ngữ C# nhóm đã chọn đề tài “Xây dựng ứng game caro”. Việc thực hiện đề tài này giúp các thành viên trong nhóm có thể vận dụng các kiến thức trong quá trình học, cũng như làm quen với việc thực hành xây dưng ứng dụng WinForm.

Ứng dụng còn có mục đích giải trí cũng như rèn luyện tư duy đối với mọi người.

## *1.3. Giới hạn đề tài*

Với phạm vi là bài tập lớn và thời gian thực hiện có hạn, cùng với những kiến thức đã học nên hệ thống của nhóm chỉ dừng lại ở một số chức năng cơ bản:

* Đăng nhập, đăng ký, phân quyền người dùng
* Chức năng quản lý User đối với quyền Aadmin (Thêm, xóa, sửa)
* Chức năng đánh với người qua mạng Lan
* Đánh với máy
* Xem thông tin chi tiết người chơi

# CHƯƠNG 2: GIỚI THIỆU CÔNG NGHỆ

## *2.1. Ngôn ngữ C#*

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.

Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code thành native code.

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

## *2.2. Winform*

Winform hay còn gọi là Windows form là giải pháp chạy trên nền Windows. Winform là công nghệ của Microsoft, cho phép lập trình các ứng dụng Windows trên PC.

## *2.3. Telerik UI For Winform*

Telerik UI for WinForms bao gồm hơn 120 điều khiển UI mà bạn có thể sử dụng để dễ dàng xây dựng các ứng dụng Line of Business độc đáo và trực quan tuyệt đẹp.

Với kiến trúc được thiết kế tốt và hỗ trợ CAB, Telerik UI for WinForms hoàn toàn phù hợp để phát triển Enterprise desktop. Chỉ với một vài cú nhấp chuột, bạn có thể bắt đầu xây dựng một ứng dụng lấy cảm hứng từ Windows.

Telerik UI for WinForms cũng có một công cụ hỗ trợ quá trình chuyển đổi các ứng dụng bằng cách sử dụng các điều khiển Microsoft Windows Forms tiêu chuẩn cho các đối tác Telerik UI for WinForms cho phép bạn dễ dàng hiện đại hóa các ứng dụng máy tính để bàn Windows của mình

## *2.4. Bunifu UI For Winform*

Bunifu UI WinForms là công cụ định hướng DLL để giúp bạn xây dựng các giao diện ứng dụng máy tính để bàn tuyệt vời. Nó đảm bảo trải nghiệm người dùng tuyệt vời trong ứng dụng của bạn và giảm thời gian phát triển.

Bunifu UI WinForms cung cấp và dễ dàng sử dụng chức năng kéo và thả trong môi trường phát triển .NET của bạn.

Với các điều khiển đã được tùy chỉnh, các điều khiển Bunifu cho phép bạn dành ít thời gian hơn để suy nghĩ về việc triển khai giao diện, giúp bạn tiết kiệm hàng giờ phát triển.

Các điều khiển Bunifu UI WinForms sử dụng tư duy thiết kế tối giản hiện đại. Nó cũng được lấy cảm hứng từ các nguyên tắc thiết kế phẳng cho trọng lượng nhẹ, giao diện tối ưu và đẹp mắt.

# CHƯƠNG 3: MÔ TẢ YÊU CẦU PHẦN MỀM

## *3.2. Firebase*

Sử dụng firebase dể lưu trữ thông tin tài khoản người dùng

## *3.2. Sử dụng ngôn ngữ C#*

Sử dụng ngôn ngữ C# để xây dựng ứng dụng Window.

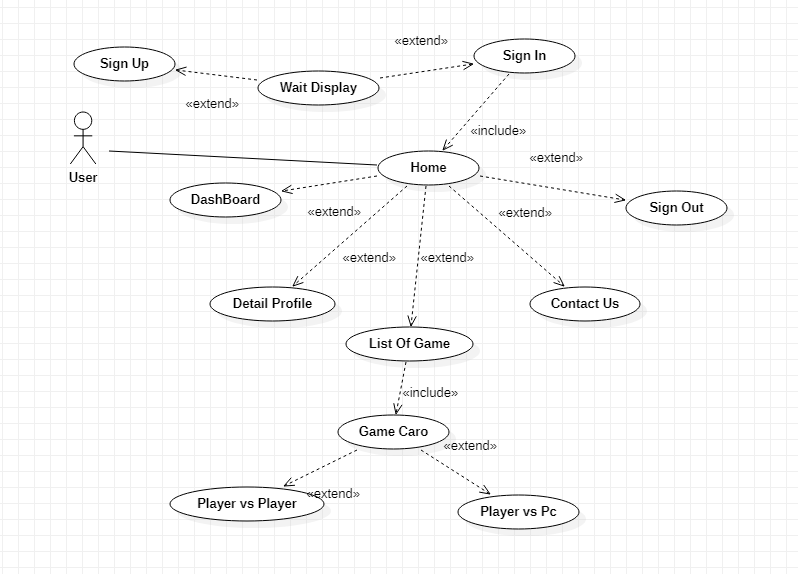
## *3.3. UI Framework*

* *Telerik UI Framework*
* Bunifu UI Framework

# CHƯƠNG 4: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## *4.1. Usecase Hệ Thống*

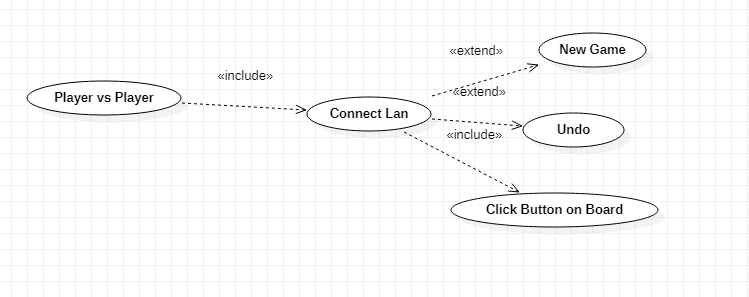
### ***4.1.1. Usecae User***



UsecaseDiagram 1: User

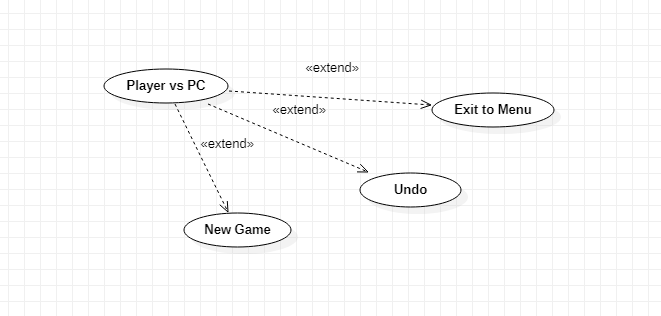
## *4.2. Usecase các chức năng*

### ***4.2.1. Usecase chế độ chơi với người***



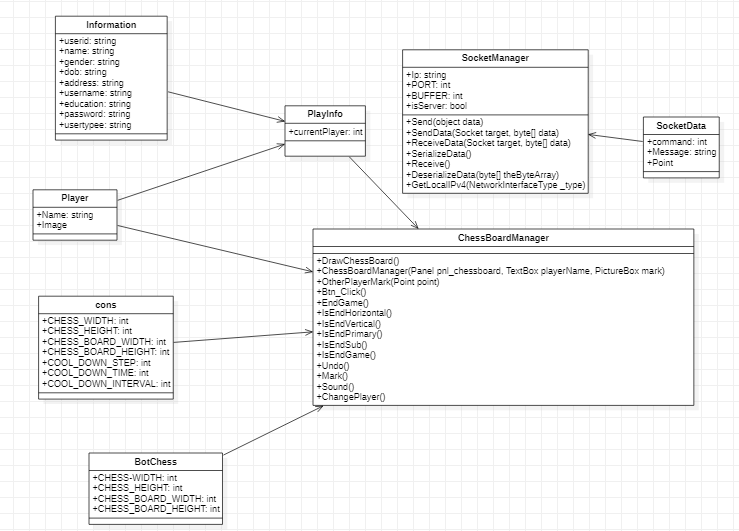
UsecaseDiagram 3: Chế độ chơi với người

### ***4.2.2. Chế độ dánh với máy***

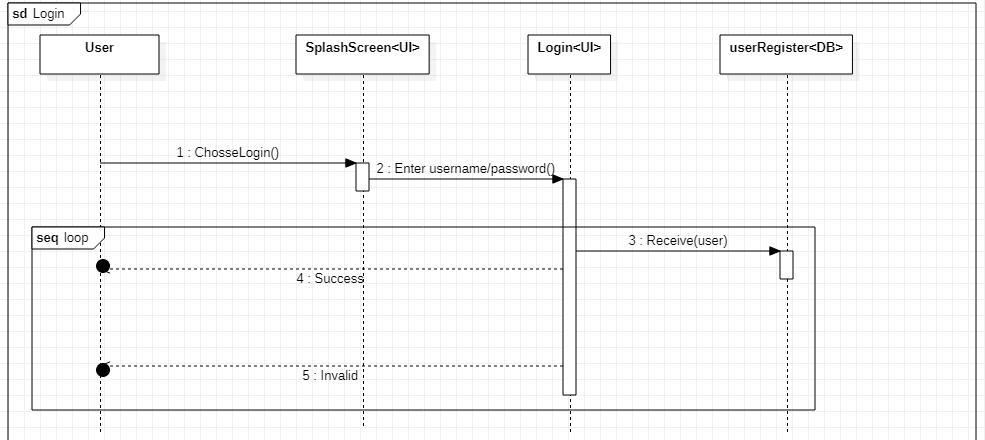


UsecaseDiagram 4: Play vs PC

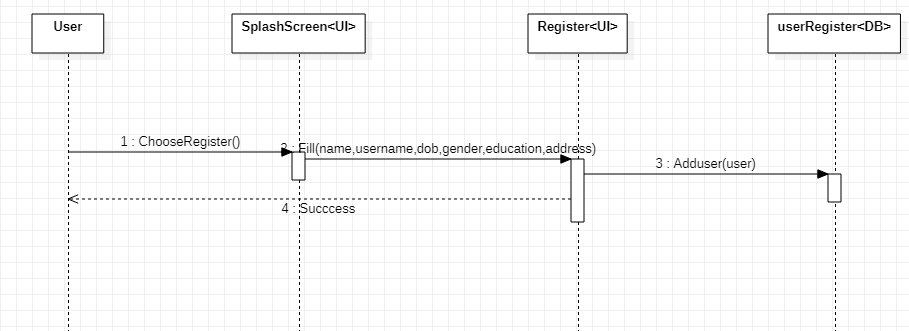
## *4.3. Class Diagram*

**

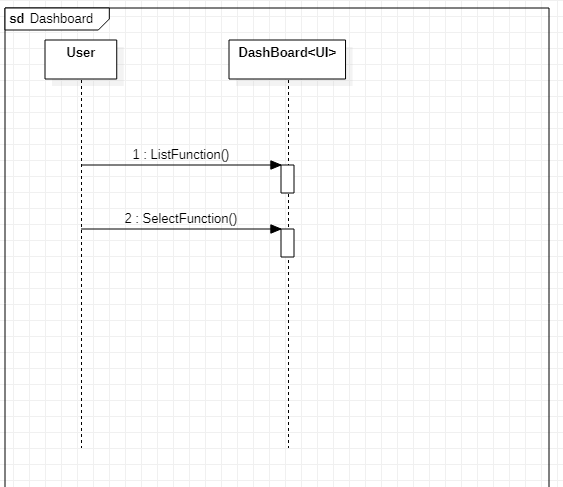
## *4.4. Sequence Diagram*

**

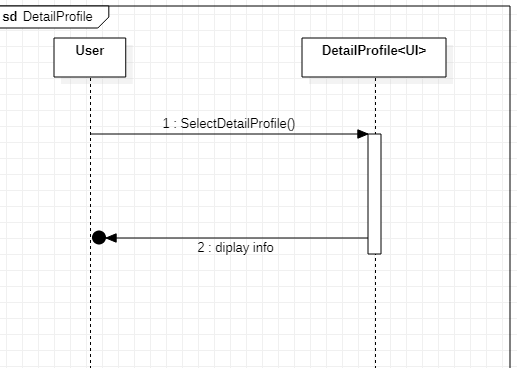
Sequence Diagram 1: Login

**

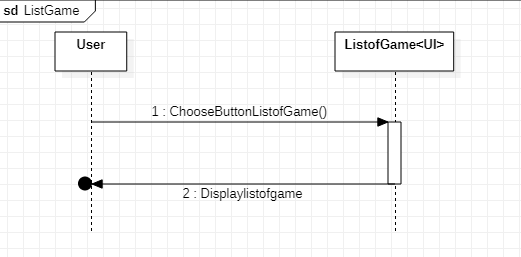
Sequence Diagram 2: Register

**

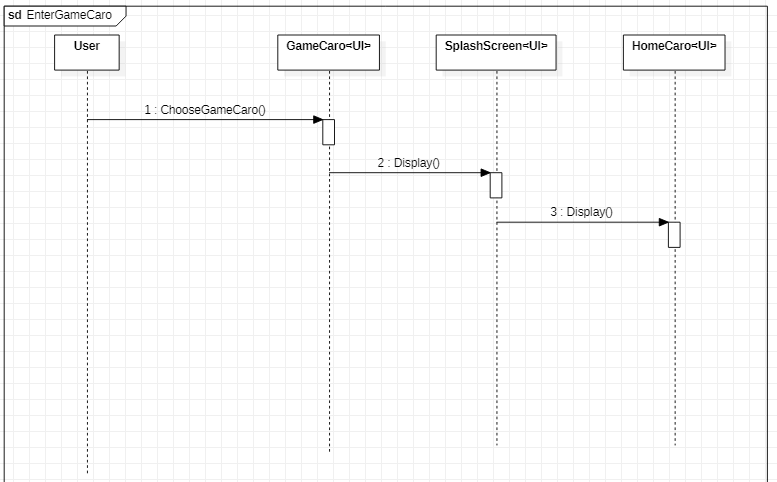
Sequence Diagram 3:Dashboard

**

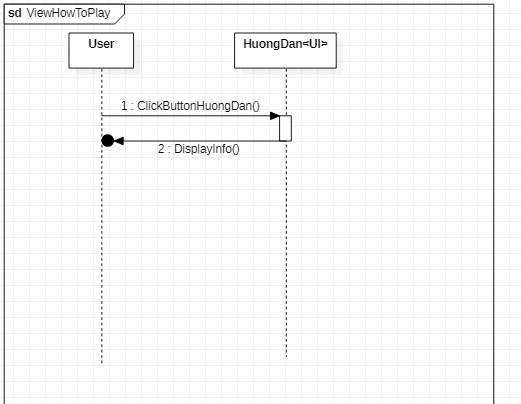
Sequence Diagram 4: Detail Profile

**

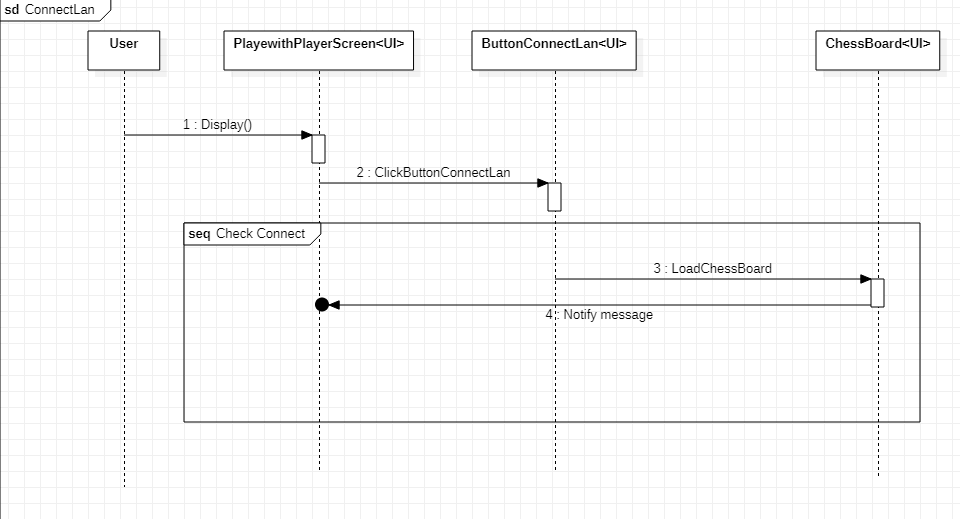
Sequence Diagram 5:List Game

**

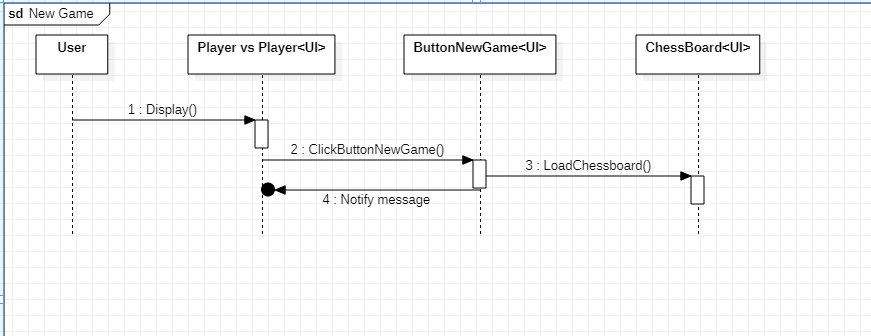
Sequence Diagram 6: Enter GameCaro

**

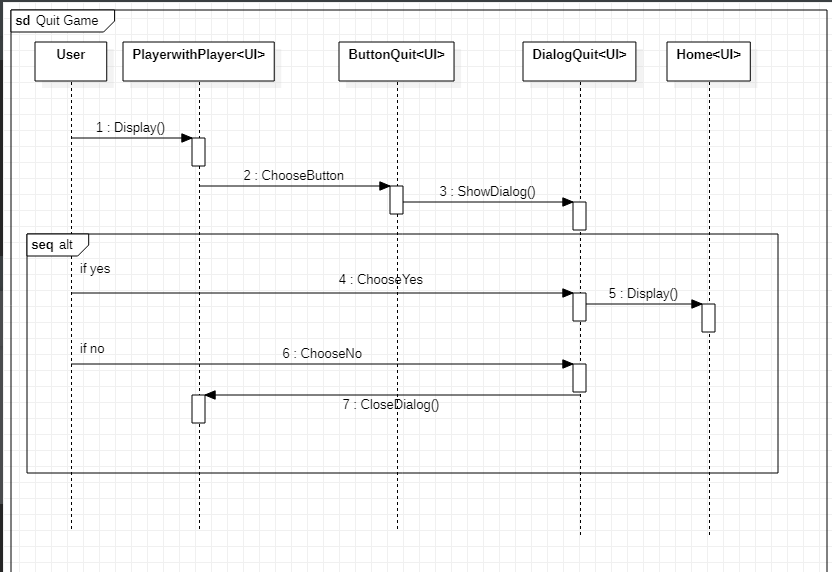
Sequence Diagram 7: How to play



Sequence Diagram 8: Connect Lan



Sequence Diagram 9: New Game



Sequence Diagram 10: Close

**CHƯƠNG 3: THIẾT KẾ GIAO DIỆN HỆ THỐNG**

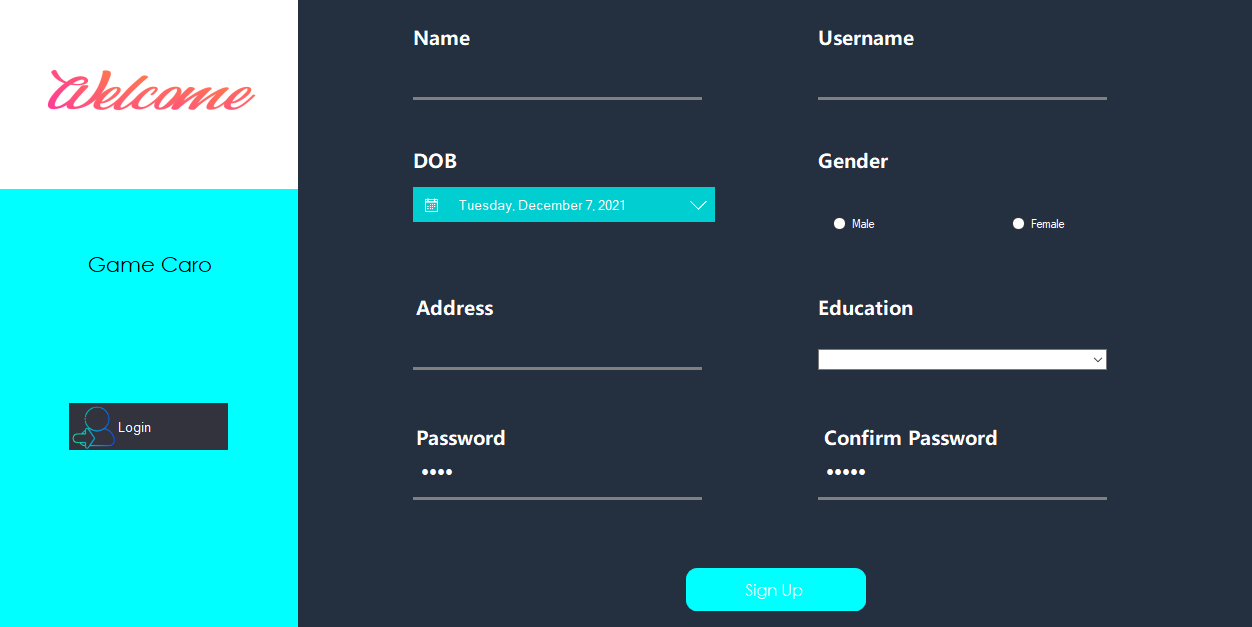
## 3.1. Màn hình cho User

### ***3.1.1. Màn hình chờ***

****

Giao diện 1: Màn hình chờ

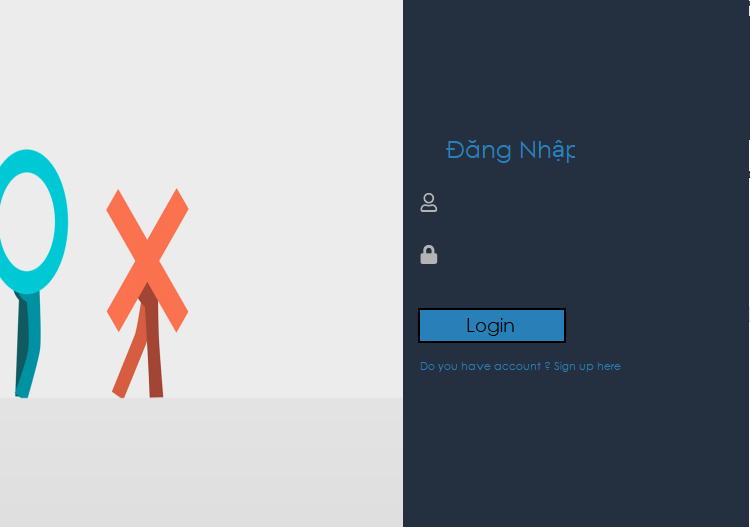
### ***3.1.2. Màn hình đăng ký***



Giao diện 2: Màn hình đăng ký

**Chức năng:** nếu người chơi chưa có tài khoản thì phải truy cập trang này để tạo cho mình một tài khoản. Sau khi tạo thành công tài khoản, người chơi có thể sử dụng tài khoản đấy để đăng nhập vào trò chơi.

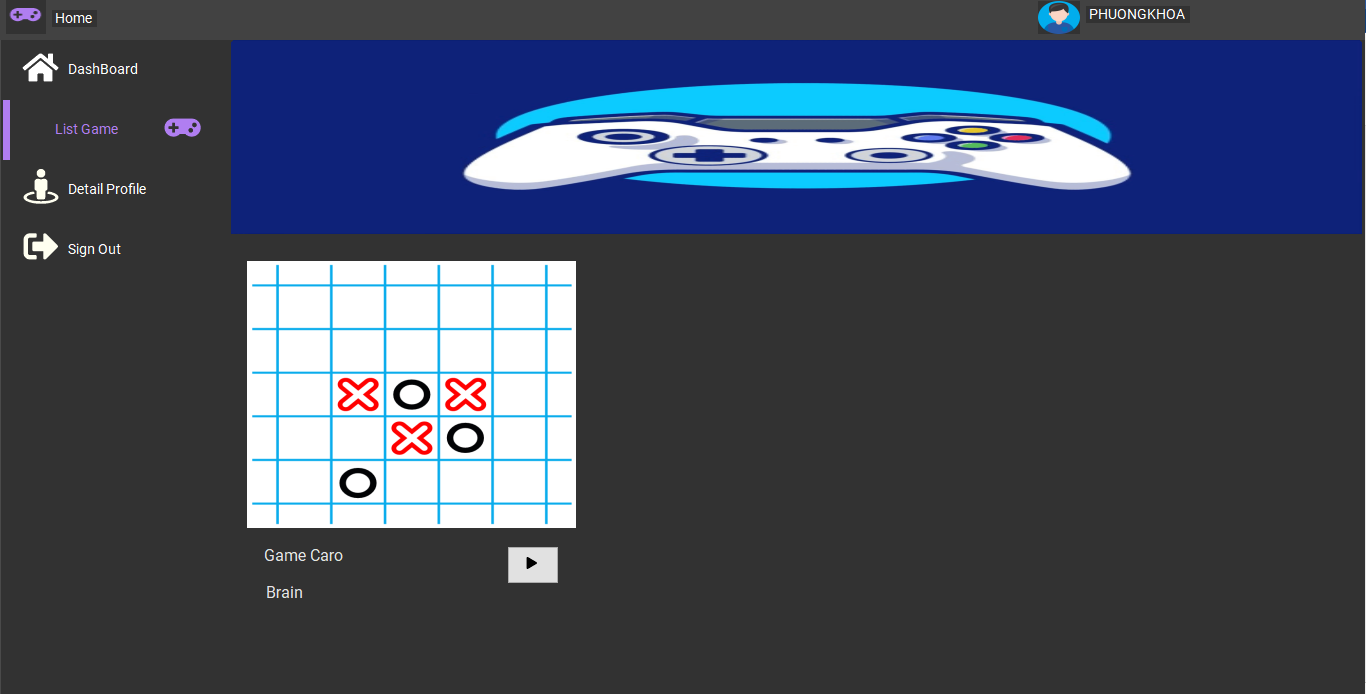
### ***3.1.3. Màn hình đăng nhập***

**

Giao diện 3: Màn hình đăng nhập

**Chức năng:** Để có thể vào được game bắt buộc người chơi phải điền đúng các thông tin đã đăng ký từ trước (tài khoản, mật khẩu). Nếu không đúng các thông tin sẽ không được vào game.

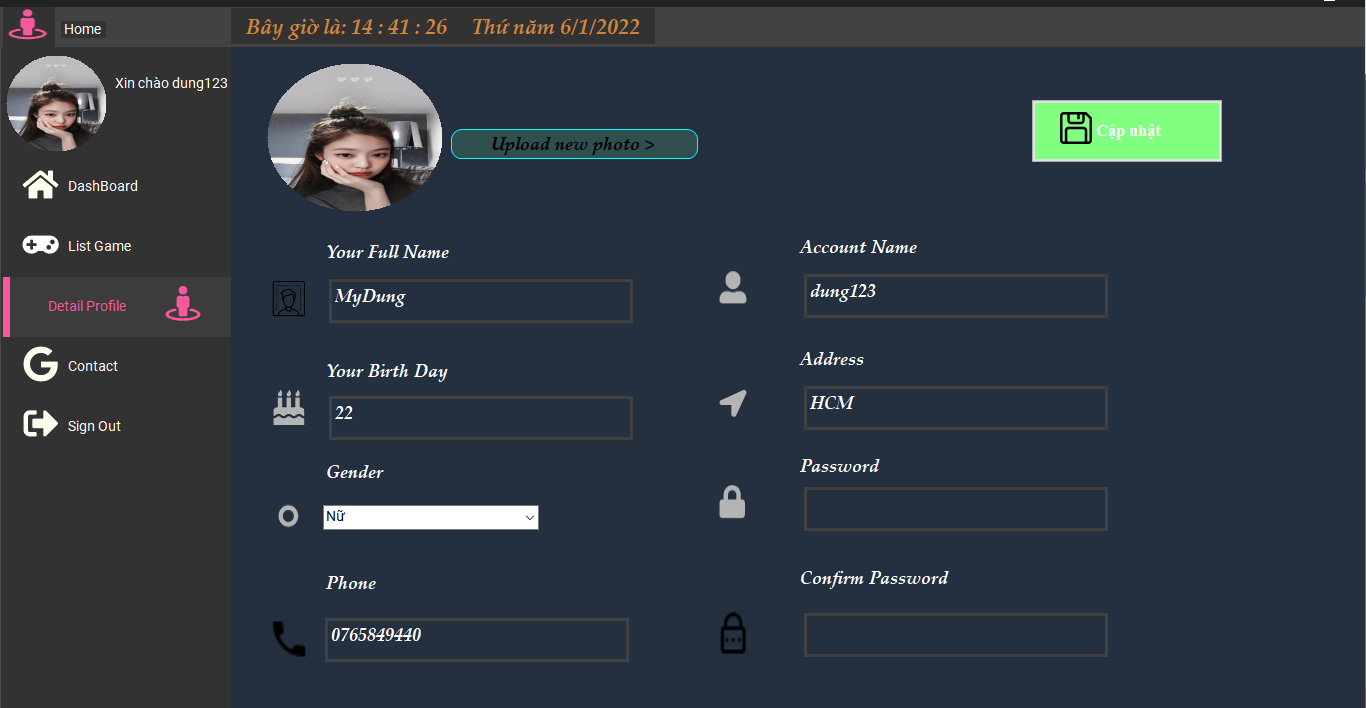
### ***3.1.4. Màn hình trang chủ***



**Giao diện 4: Màn hình trang chủ**

**Chức năng:** Ở màn hình này sẽ hiển thị danh sách game. Bạn có thế click vào biểu tượng play để có thể vào game.

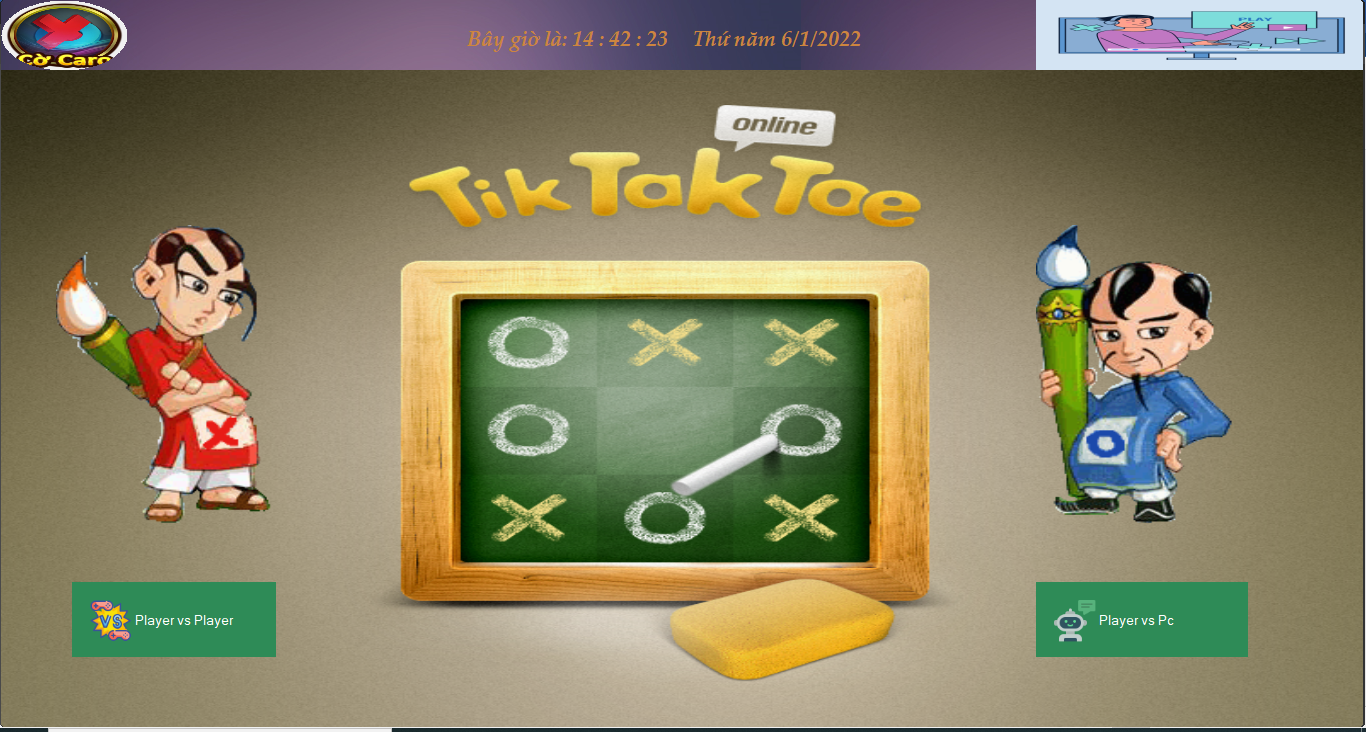
### ***3.1.5. Màn hình thông tin người chơi***



Giao diện 5: Detail Profile

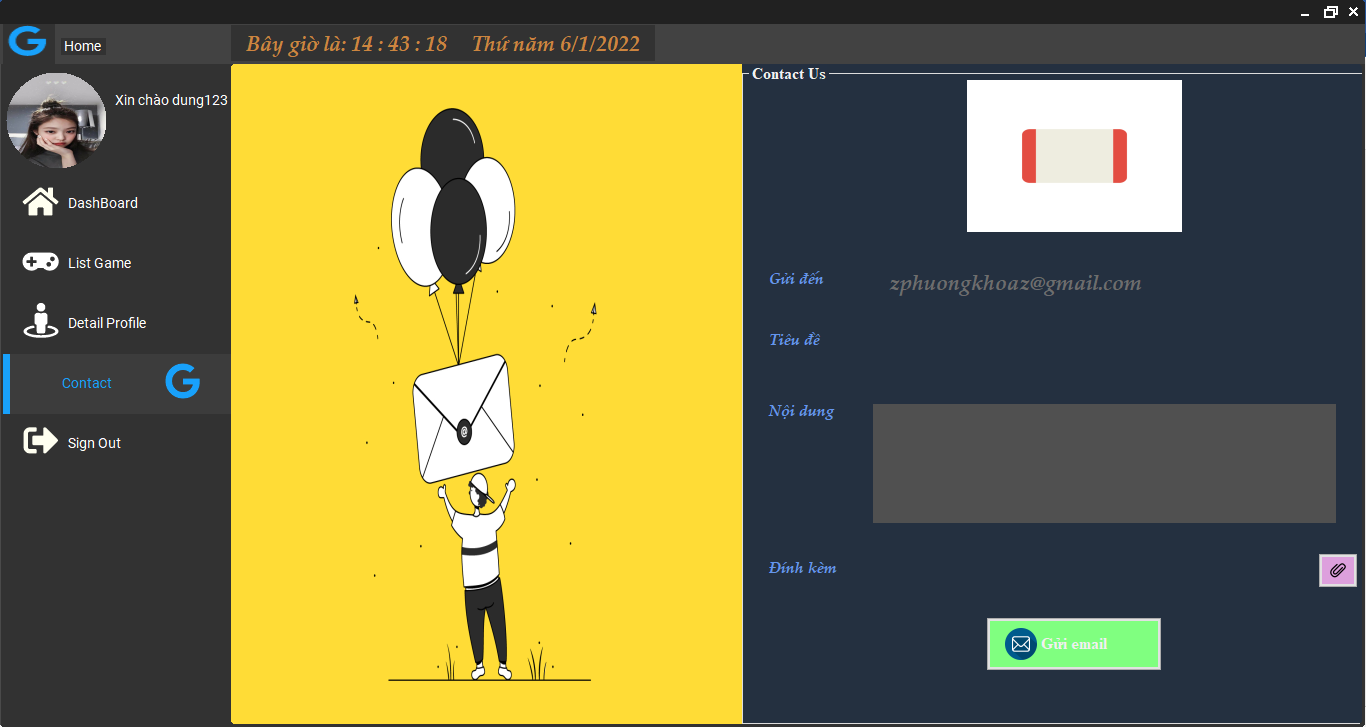
**Chức năng:** Ở màn hình này bạn có thể xem được thông tin chi tiết của bạn, đồng thời có thể cập nhật thông tin và ảnh nền.

### ***3.1.6. Màn hình trang chủ game caro***



Giao diện 6: Màn hình Game Caro

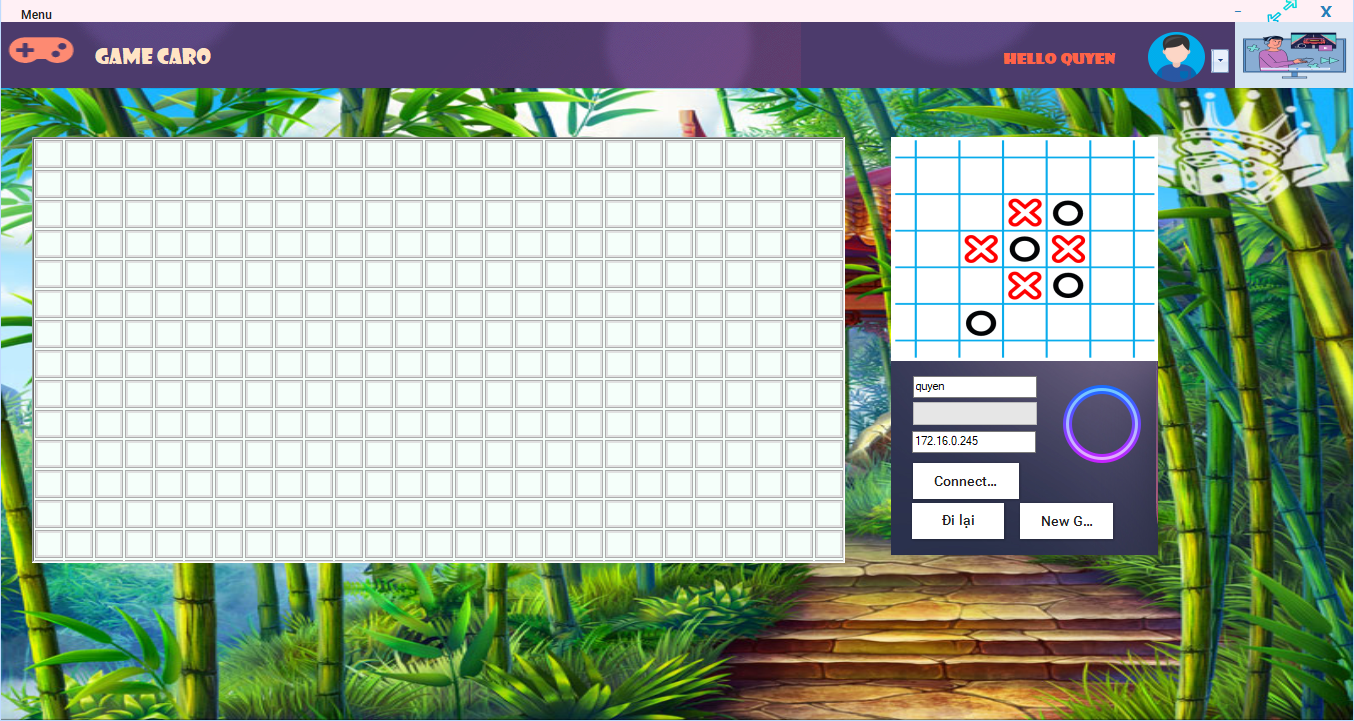
### ***3.1.7. Màn hình trang Send Email***



Giao diện 7: Trang Send Emai

**Chức năng**: Ở màn hình này người dùng có thể gửi mail về bất kỳ sự cố nào người dùng gặp phại cho quản trị viên.

### ***3.1.8. Màn Player vs Player***



Giao diện 8: Màn hình chơi game

**Chức năng:** Ở màn hình này gôm các chức năng sau:

* Nút Connect Lan: dùng để Connect đến mạng Lan
* Nút Đi lại: người chơi có thể đi lại nước cờ.
* Nút New Game: Người chơi bắt đầu game mới.
* Bàn cờ: nơi người chơi đánh những nước cờ bằng cách click vào từng ô trên bàn cờ.

### ***3.1.9. Màn hình Player vs Pc***



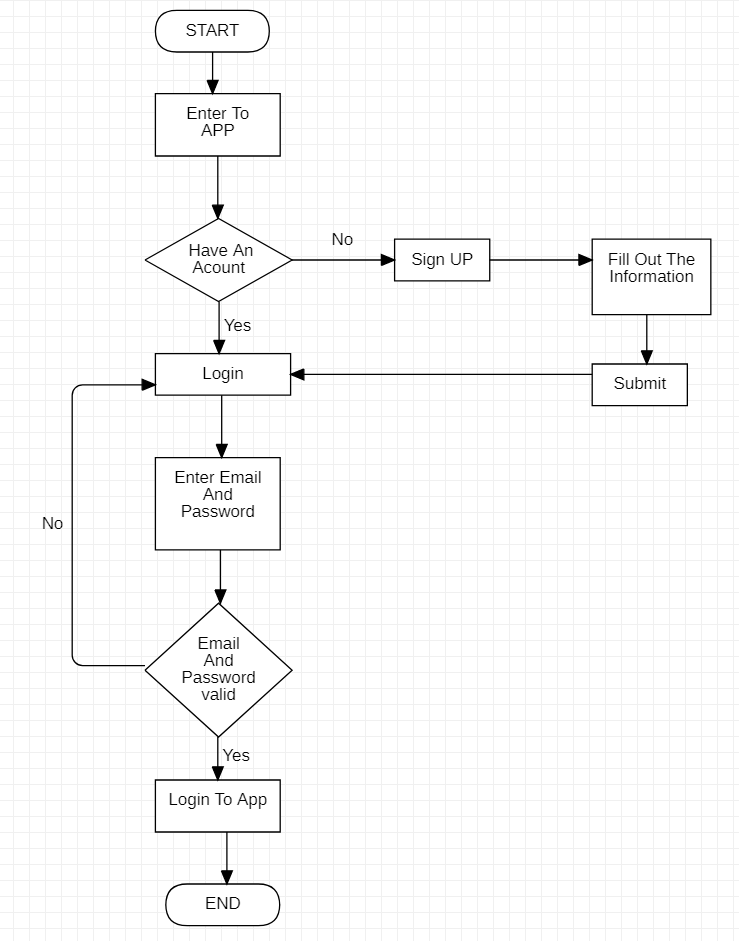
Giao diện 9: Player vs PC

**Chức năng:** Ở màn hình này gôm các chức năng sau:

* Nút Đi lại: người chơi có thể đi lại nước cờ.
* Nút New Game: Người chơi bắt đầu game mới.
* Bàn cờ: nơi người chơi đánh những nước cờ bằng cách click vào từng ô trên bàn cờ.

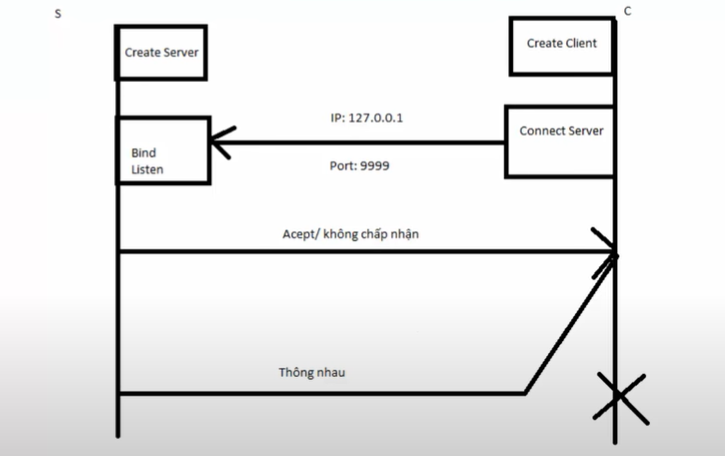
# CHƯƠNG 4: MÔ TẢ VÀ THIẾT KẾ MỘT SỐ THUẬT TOÁN (VẼ FLOWCHART)

## *4.1. Login*



Thuật toán : Login

## 4.2 Connect mạng LAN



Thuật toán : Connect Lan

* Đầu tiên sẽ khởi tạo server -> Đợi Client
* Tiếp theo Client sẽ khởi tạo -> Connect tới Server
* Client Connect Server thông qua IP, Port
* Sau đó Server trả về kết quả accept/ko chấp nhận
* Nếu không chấp nhận thì nó cắt
* Còn nếu chấp nhận thì nó tiếp tục tạo đường thông với nhau.

## 4.3. Thuật toán đánh với máy

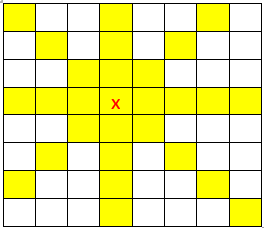
* **Thuật toán kiểm tra kết thúc**

Tên thủ tục và tham số: Check(int x, int y)

Các biến:

* Int i,j: dùng để đếm chạy vòng lặp kiểm tra
* Int column, row, mdiagonal, ediagonal: dùng để đếm chuỗi bao nhiêu quân liên tiếp có sẵn trên bàn cờ theo cột, hàng, đường chéo chính, đường chéo phụ.

Nguyên lý hoạt động: Chạy vòng while để duyệt các ô xung quanh ô [x,y] vừa mới được đặt quân cờ, nếu có xuất hiện ô [i,j] giống quân cờ với ô [x,y] thì tăng các biến đếm column, row, mdiagonal, ediagonal tương ứng với ô[i,j] nằm ở vị trí tương đối nào so với ô [x,y] ( trên cùng hàng, cùng cột, cùng đường chéo chính, cùng đường chéo phụ)



Thuật toán : Các ô [i,j] được duyệt (màu vàng) xung quanh ô [x,y]

private void Check(int x, int y)

{

int i = x - 1, j = y;

int column = 1, row = 1, mdiagonal = 1, ediagonal = 1;

while (vtMap[x, y] == vtMap[i, j] && i >= 0)

{

column++;

i--;

}

i = x + 1;

while (vtMap[x, y] == vtMap[i, j] && i <= rows)

{

column++;

i++;

}

i = x; j = y - 1;

while (vtMap[x, y] == vtMap[i, j] && j >= 0)

{

row++;

j--;

}

j = y + 1;

while (vtMap[x, y] == vtMap[i, j] && j <= columns)

{

row++;

j++;

}

i = x - 1; j = y - 1;

while (vtMap[x, y] == vtMap[i, j] && i >= 0 && j >= 0)

{

mdiagonal++;

i--;

j--;

}

i = x + 1; j = y + 1;

while (vtMap[x, y] == vtMap[i, j] && i <= rows && j <= columns)

{

mdiagonal++;

i++;

j++;

}

i = x - 1; j = y + 1;

while (vtMap[x, y] == vtMap[i, j] && i >= 0 && j <= columns)

{

ediagonal++;

i--;

j++;

}

i = x + 1; j = y - 1;

while (vtMap[x, y] == vtMap[i, j] && i <= rows && j >= 0)

{

ediagonal++;

i++;

j--;

}

if (row >= 5 || column >= 5 || mdiagonal >= 5 || ediagonal >= 5)

{

Gameover();

if (vsComputer)

{

if (player == 1)

MessageBox.Show("You win!!");

else

MessageBox.Show("You lost!!");

}

}

* **Thuật toán tìm nước đi tối ưu**

Khởi tạo:

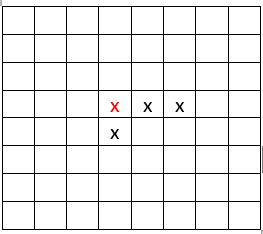
* Mảng 1 chiều:

int Attack[7] ={ 0; 9; 54; 162; 1458; 13112; 118008}

int Defense[7]={0; 3; 27; 99; 729; 6561; 59049}

các phần tử của 2 mảng này là các con số ngẫu nhiên nhưng theo quy luật tăng dần, dùng để quy ước điểm số. Giả sử máy đặt một ô cờ [x,y] (màu đỏ) thì điểm số tại ô này được tính là tạo thành chuỗi hàng 3 quân liên tiếp và chuỗi cột 2 quân liên tiếp. Số điểm là:

= ATTACK[3] + ATTACK[2]



Minh họa cơ chế tính điểm

Tương tự với điểm phòng thủ.

**Các hàm:**

ComputerChesses(int x, int y): tính điểm TẤN CÔNG khi giả sử máy đánh tại ô [x,y]

EnemyChesses(int x, int y): Tính điểm PHÒNG THỦ khi giả sử máy đánh tại ô [x,y]

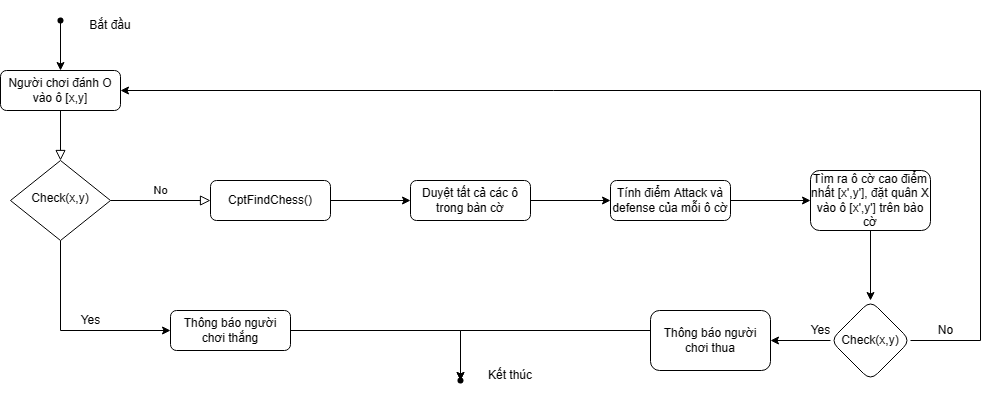
Calculate(int attack, int defense) Tính tổng điểm tấn công và phòng thủ khi giả sử máy đánh tại ô [x,y]

**Các thủ tục:**

CptFindChess(): duyệt bàn cờ và tìm ô [x,y] có số điểm cao nhất, nếu tìm ra được thì gọi hàm PutChess()

PutChess(int x, int y): đặt quân cờ X tối ưu vào ô [x,y] trên bàn cờ

***Nguyên lý hoạt động***



**Hàm ComputerChesses(int x, int y)**

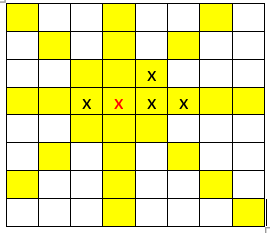
Các biến sử dụng:

Int i, j: biến đếm dùng để duyệt bàn cờ.

Int column, row, mdiagonal, ediagonal:khởi tạo bằng 0, dùng để đếm nếu đặt quân X của máy vào ô [x,y] thì sẽ tạo thành chuỗi bao nhiêu quân liên tiếp tương ứng vị trí trên cột, hàng, đường chéo chính, đường chéo phụ.

Int sc, sc\_, sr, sr\_, smdiagonal, smdiagonal\_, sediagonal, sediagonal\_: khởi tạo bằng 0, cặp biến bao gồm sc và sc\_ dùng để đánh dấu trên cột (hướng lên và hướng xuống) không bị chặn bởi quân O của người chơi. Sử dụng các biến này để ưu tiên đánh các quân X tạo thành chuỗi 3 quân liên tiếp mà không bị chặn ở 2 đầu. Tương tự với các biến sr, sr\_, smdiagonal, smdiagonal\_, sediagonal, sediagonal\_.

Hàm ComputerChesses đầu tiên sẽ duyệt trên cột, hàng, đường chéo chính, đường chéo phụ quanh ô [x,y] (màu đỏ) và đếm xem đã có sẵn bao nhiêu quân X (màu đen) trên các vị trị đó rồi.



Ví dụ: column = 0, row = 3, ediagonal = 2, mdiagonal =0

if (column == 4) column = 5;

if (row == 4) row = 5;

if (mdiagonal == 4) mdiagonal = 5;

if (ediagonal == 4) ediagonal = 5;

if (column == 3 && sc == 1 && sc\_ == 1) column = 4;

if (row == 3 && sr == 1 && sr\_ == 1) row = 4;

if (mdiagonal == 3 && sm == 1 && sm\_ == 1) mdiagonal = 4;

if (ediagonal == 3 && se == 1 && se\_ == 1) ediagonal = 4;

if (column == 2 && row == 2 && sc == 1 && sc\_ == 1 && sr == 1 && sr\_ == 1) column = 3;

if (column == 2 && mdiagonal == 2 && sc == 1 && sc\_ == 1 && sm == 1 && sm\_ == 1) column = 3;

if (column == 2 && ediagonal == 2 && sc == 1 && sc\_ == 1 && se == 1 && se\_ == 1) column = 3;

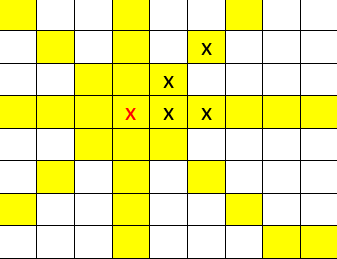
if (row == 2 && mdiagonal == 2 && sm == 1 && sm\_ == 1 && sr == 1 && sr\_ == 1) column = 3;

if (row == 2 && ediagonal == 2 && se == 1 && se\_ == 1 && sr == 1 && sr\_ == 1) column = 3;

if (ediagonal == 2 && mdiagonal == 2 && sm == 1 && sm\_ == 1 && se == 1 && se\_ == 1) column = 3;

Khi đã có số lượng quân X có sẵn, máy tính sẽ chia trường hợp ưu tiên:

* Nếu đã có sẵn 4 ô X liên tiếp,tăng biến đếm chuỗi thành 5, đánh X tạo thành ô thứ 5 và chiến thắng
* Nếu đã có sẵn 3 ô X liên tiếp và không bị chặn 2 đầu, tăng biến đếm chuỗi thành 4, đánh X tạo thành ô thứ 4
* Nếu đã có sẵn 2 ô X liên tiếp trên bất kỳ 2 vị trí của cột, hàng, chéo chính, chéo phụ, tăng biến đếm chuỗi thành 3, đánh X tạo thành ô thứ 3 nước đôi



Ví dụ nước đôi, ediagonal = 2, mdiagonal=2, đánh X để tạo thành chuỗi 3

* Các trường hợp còn lại bao gồm chuỗi 2, chuỗi 3 bị chặn thì các biến đếm column, row, ediagonal, mdiagonal sẽ giữ nguyên sau khi duyệt.

Các trường hợp trên ngược lại khi tính điểm phòng thủ. Khởi tạo mảng Defense[] có các phần tử nhỏ hơn mảng Attack[] nên máy tính sẽ ưu tiên tấn công trước.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

# CHƯƠNG 5: HƯỚNG PHÁT TRIỂN VÀ KẾT QUẢ ĐỀ TÀI, HẠN CHẾ

## 9.1. Kết quả đạt được

* Áp dụng được các kiến thức đã học vào hệ thống.
* Xây dựng được ứng dụng Game Caro phù hợp với mọi đối tượng người chơi
* Thiết kế giao diện đẹp mắt.

## *9.2. Phát triển ứng dụng*

Với mục đích xây dựng hệ thống , giúp người dùng có những trải nghiệm tốt nhất khi sử dụng, nhóm đã có những hướng phát triển mới giúp hế thống có thể vận hành tốt hơn với nhiều chức năng hơn.

Khắc phục một số nhược điểm còn tồn tại trong quá trình sử dụng của người dùng. Nâng cấp thêm một số tính năng giúp người dùng có thể thuận tiện hơn trong việc sử dụng. Ngoài ra nhóm sẽ phát triển thêm tính năng như tính điểm người chơi, kết bạn, xếp hạng.

## *9.3. Hạn chế*

Các chức năng trong ứng dụng vẫn còn ít, một số chức năng khác chưa thât sự tối ưu.

# TÀI LIỆU THAM KHẢO

# https://www.howkteam.vn/learn