**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN**

**MÔN: LẬP TRÌNH MẠNG CĂN BẢN**

**ĐỀ TÀI: “ỨNG DỤNG CHƠI CỜ CARO”**

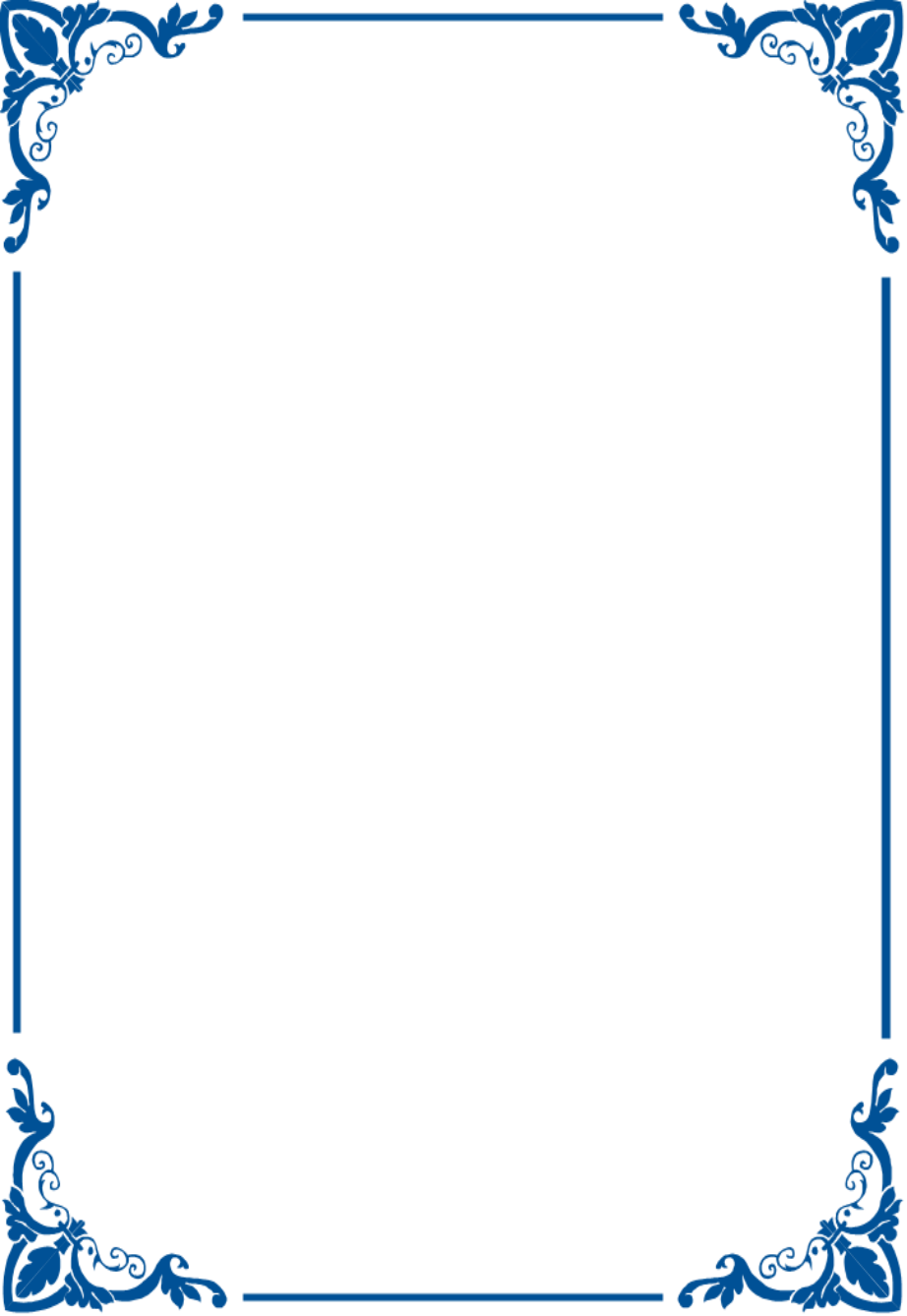
**Giảng viên hướng dẫn:** Trần Hồng Nghi

**Sinh viên thực hiện:** 22521639 - Lê Thị Tú Uyên

22521537 - Mai Kim Trinh

22520436 - Lê Nguyễn Minh Hiếu

22520312 - Đào Nhật Duy



**MỤC LỤC**

[LỜI CẢM ƠN 3](#_heading=h.30j0zll)

[DANH MỤC HÌNH ẢNH 4](#_heading=h.1fob9te)

[Chương 1: TỔNG QUAN ĐỀ TÀI 6](#_heading=h.3znysh7)

[I. Giới thiệu đề tài 6](#_heading=h.2et92p0)

[II. Mục tiêu đề tài 7](#_heading=h.3dy6vkm)

[III. Công nghệ sử dụng 7](#_heading=h.1t3h5sf)

[Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 8](#_heading=h.4d34og8)

[I. Sơ đồ phân rã chức năng 8](#_heading=h.2s8eyo1)

[II. Cấu trúc Database 8](#_heading=h.3rdcrjn)

[Chương 3: HIỆN THỰC ĐỀ TÀI 10](#_heading=h.lnxbz9)

[I. Kết nối database Firebase 10](#_heading=h.35nkun2)

[1. Class Table 10](#_heading=h.1ksv4uv)

[2. Class DataBase 13](#_heading=h.4i7ojhp)

[IsUsernameExists(string uName): 13](#_heading=h.1ci93xb)

[AddNewUser(string uName, string pass, string mail): 13](#_heading=h.qsh70q)

[CheckLogin(string uName, string uPass): 14](#_heading=h.1pxezwc)

[GetUser(): 14](#_heading=h.2p2csry)

[ChangePass(string newPass): 15](#_heading=h.3o7alnk)

[UserOnline(); UserOffline(): 15](#_heading=h.ihv636)

[ChangeAvatar(int index): 16](#_heading=h.1hmsyys)

[UpdateUserInfo(): 16](#_heading=h.2grqrue)

[SendFriendRequest(int idRcv): 16](#_heading=h.3fwokq0)

[AcceptFriendRequest(int idSend): 17](#_heading=h.4f1mdlm)

[Hàm lấy ra danh sách: 17](#_heading=h.19c6y18)

[SendMessTo(int uReceive, string content): 18](#_heading=h.28h4qwu)

[ListMessWith(int idUser): 19](#_heading=h.37m2jsg)

[AddMatch(int idUser): 19](#_heading=h.46r0co2)

[SetScore\_Winrate(int idUser): 20](#_heading=h.111kx3o)

[HistoryMatch(): 20](#_heading=h.206ipza)

[II. Socket 20](#_heading=h.2zbgiuw)

[1. Class SocketData: 20](#_heading=h.1egqt2p)

[2. Class SocketManager: 22](#_heading=h.sqyw64)

[3. Truyền dữ liệu giữa các socket: 25](#_heading=h.1664s55)

[III. Gửi nhận mail 27](#_heading=h.25b2l0r)

[Chương 4: THỰC NGHIỆM ĐỀ TÀI 28](#_heading=h.1jlao46)

[1. Giao diện đăng nhập (Form Login) 28](#_heading=h.43ky6rz)

[2. Giao diện đăng ký (Form Sign Up) 29](#_heading=h.xvir7l)

[3. Giao diện quên mật khẩu (Form Forgot Password) 30](#_heading=h.1x0gk37)

[4. Giao diện chính (Form Main) 31](#_heading=h.2w5ecyt)

[5. Giao diện chơi game (Form Home) 31](#_heading=h.3vac5uf)

[6. Giao diện nhắn tin (Form Message) 33](#_heading=h.1opuj5n)

[7. Giao diện cá nhân (Form Profile) 34](#_heading=h.2nusc19)

[8. Giao diện cài đặt (Form Setting) 35](#_heading=h.3mzq4wv)

[9. Giao diện Hướng dẫn (Form Guide) 37](#_heading=h.1gf8i83)

[10. Giao diện Báo cáo (Form Report) 38](#_heading=h.2fk6b3p)

[Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 38](#_heading=h.3ep43zb)

[TÀI LIỆU THAM KHẢO 39](#_heading=h.1tuee74)

[BẢNG PHÂN CHIA CÔNG VIỆC 40](#_heading=h.4du1wux)

# 

# LỜI CẢM ƠN

Trước hết, chúng em muốn bày tỏ lòng biết ơn sâu sắc đến toàn thể giảng viên và nhân viên của Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM đã tạo điều kiện cho chúng em tham gia môn học **Lập trình mạng căn bản**, từ đó giúp chúng em có được những kiến thức cơ bản cần thiết để thực hiện đề tài này.

Đặc biệt, chúng em xin gửi lời cảm ơn và lòng biết ơn sâu sắc tới cô **Trần Hồng Nghi** - giảng viên lý thuyết môn **Lập trình mạng căn bản.** Qua những kiến thức được học từ môn **Lập trình mạng căn bản**, những lời giảng tận tình chu đáo và những lời góp ý chân thành mà cô đưa ra, chúng em đã và đang cố gắng hoàn thiện dần thông qua quá trình học tập.

Trải qua một học kỳ dài, chúng em đã nhận thấy cô **Trần Hồng Nghi** là một người cô nhiệt huyết và chu đáo với sinh viên. Phong cách giảng dạy độc đáo của cô không chỉ giúp chúng em dễ dàng tiếp cận kiến thức mà còn áp dụng vào cuộc sống hàng ngày một cách hiệu quả. Môn học này không chỉ đem lại niềm vui và sự hấp dẫn mà còn là nền tảng quan trọng giúp chúng em tiến xa hơn trong học tập và sự nghiệp sau này.

Sau quãng thời gian dài thực hiện đề tài, chúng em đã cùng nhau tìm hiểu, nghiên cứu và thảo luận để hoàn thiện công việc một cách cẩn thận nhất. Dù với ít kinh nghiệm và kỹ năng, cùng với việc làm việc nhóm lần đầu tiên, chắc chắn không tránh khỏi những sai sót. Chúng em rất mong nhận được sự góp ý từ cô để có thể hoàn thiện hơn qua đề tài này.

**Nhóm 21** – lớp **NT106.022.ANTT**, xin chúc cô sức khỏe dồi dào, thành công và hạnh phúc trong cuộc sống.

Chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, tháng 4 năm 2024

Nhóm sinh viên thực hiện

***Nhóm 21***

# DANH MỤC HÌNH ẢNH

[Hình 1: Cờ caro……………………………………………………………………. 6](#_heading=h.tyjcwt)

[Hình 2: Sơ đồ phân rã chức năng………………………………………………….. 8](#_heading=h.17dp8vu)

[Hình 3: Cấu trúc Database…………………………………………………………. 9](#_heading=h.26in1rg)

[Hình 4: Class User………………………………………………………………... 10](#_heading=h.44sinio)

[Hình 5: Class Message……………………………………………………………11](#_heading=h.2jxsxqh)

[Hình 6:Class Match………………………………………………………………. 11](#_heading=h.z337ya)

[Hình 8: Clas Friend………………………………………………………………. 12](#_heading=h.3j2qqm3)

[Hình 9: Class AdddInvite………………………………………………………… 13](#_heading=h.1y810tw)

[Hình 10:Khởi tạo kết nối FireBase……………………………………………….. 13](#_heading=h.2xcytpi)

[Hình 11: Hàm IsUserNameExists(string uName)................................................... 13](#_heading=h.2bn6wsx)

[Hình 12: Hàm AddNewUser(string uName, string pass, string mail)..................... 14](#_heading=h.3as4poj)

[Hình 13: Hàm CheckLogin(string uName, string uPass)........................................ 14](#_heading=h.49x2ik5)

[Hình 14: Hàm GetUser()......................................................................................... 15](#_heading=h.147n2zr)

[Hình 15: Hàm ChangePass(string newPass)........................................................... 15](#_heading=h.23ckvvd)

[Hình 16: Hàm UserOnline(); UserOffline()............................................................ 16](#_heading=h.32hioqz)

[Hình 17: Hàm ChangeAvatar(int index).................................................................. 16](#_heading=h.41mghml)

[Hình 18: Hàm UpdateUserInfo()............................................................................. 16](#_heading=h.vx1227)

[Hình 19: Hàm SendFriendRequest(int idRcv)........................................................ 17](#_heading=h.1v1yuxt)

[Hình 20: Hàm AcceptFriendRequest(int idSend).................................................... 17](#_heading=h.2u6wntf)

[Hình 21: Hàm ListRequests().................................................................................. 18](#_heading=h.3tbugp1)

[Hình 22: Hàm SendMessTo(int uReceive, string content) ………………………. 19](#_heading=h.nmf14n)

[Hình 23: Hàm ListMessWith(int idUser)................................................................ 19](#_heading=h.1mrcu09)

[Hình 24: Hàm AddMatch(int idUser)..................................................................... 20](#_heading=h.2lwamvv)

[Hình 25: Hàm SetScore\_Winrate(int idUser)......................................................... 20](#_heading=h.3l18frh)

[Hình 26: Hàm HistoryMatch()................................................................................ 20](#_heading=h.4k668n3)

[Hình 27: Class SockeData………………………………………………………... 21](#_heading=h.3ygebqi)

[Hình 28: Enum SocketCommand………………………………………………… 22](#_heading=h.2dlolyb)

[Hình 29: Hàm Connect Server…………………………………………………… 23](#_heading=h.3cqmetx)

[Hình 30: Hàm Create Server……………………………………………………... 24](#_heading=h.1rvwp1q)

[Hình 31: Hàm Create Server 2…………………………………………………… 24](#_heading=h.4bvk7pj)

[Hình 32: Hàm gửi nhận dữ liệu…………………………………………………... 25](#_heading=h.2r0uhxc)

[Hình 33: Hàm ProcessData()................................................................................... 26](#_heading=h.3q5sasy)

[Hình 34: Gửi mail………………………………………………………………… 28](#_heading=h.kgcv8k)

[Hình 35: Gửi mail (tt).............................................................................................. 28](#_heading=h.34g0dwd)

[Hình 36. Form Login……………………………………………………………... 29](#_heading=h.2iq8gzs)

[Hình 37. Form Sign Up…………………………………………………………... 29](#_heading=h.3hv69ve)

[Hình 38. Form Forgot Password…………………………………………………. 30](#_heading=h.4h042r0)

[Hình 39. Form Main……………………………………………………………… 31](#_heading=h.1baon6m)

[Hình 40.Chế độ Practice…………………………………………………………. 32](#_heading=h.2afmg28)

[Hình 41.Chế độ PvBot…………………………………………………………… 32](#_heading=h.pkwqa1)

[Hình 42.Chế độ PvP……………………………………………………………… 33](#_heading=h.39kk8xu)

[Hình 43. Form Message………………………………………………………….. 33](#_heading=h.48pi1tg)

[Hình 44. Form Profile…………………………………………………………….. 34](#_heading=h.1302m92)

[Hình 45. Form General…………………………………………………………… 35](#_heading=h.2250f4o)

[Hình 46. Form Account………………………………………………………….. 36](#_heading=h.haapch)

[Hình 47. Form Privacy…………………………………………………………… 36](#_heading=h.319y80a)

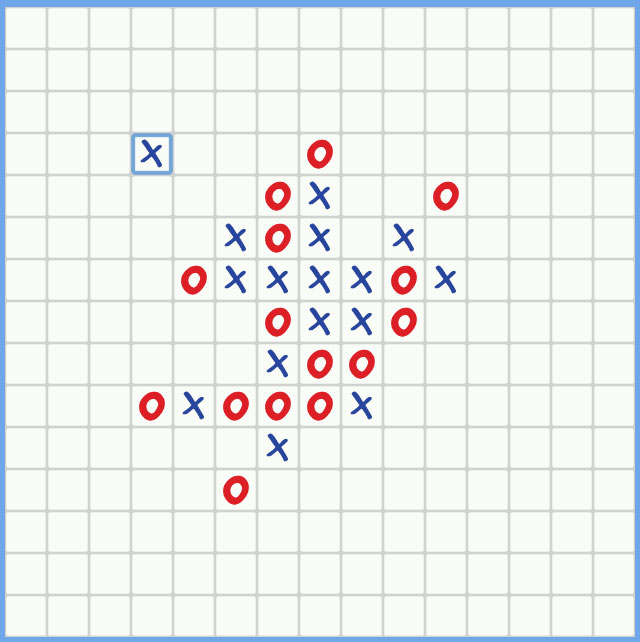
[Hình 48. Form Guide…………………………………………………………….. 37](#_heading=h.40ew0vw)

[Hình 49. Form Report……………………………………………………………. 38](#_heading=h.upglbi)

# Chương 1: TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài

* Cờ Caro, một trong những trò chơi cờ trí tuệ phổ biến, đã thu hút rất nhiều người chơi từ các độ tuổi khác nhau. Với sự phát triển của công nghệ, việc chơi cờ Caro không chỉ giới hạn trong không gian vật lý mà còn có thể trải nghiệm thông qua các ứng dụng mạng xã hội và phần mềm.
* Cờ caro, hay Gomoku, có lịch sử lâu đời bắt nguồn từ Trung Quốc, nơi nó được chơi trên bàn cờ Go. Trò chơi lan đến Nhật Bản, được gọi là "Gomoku," và trở nên phổ biến rộng rãi. Người Nhật phát triển biến thể Renju để tăng tính cân bằng. Cờ caro cũng lan sang Hàn Quốc với tên "Omok" và sau đó đến các nước phương Tây vào cuối thế kỷ 19, đầu thế kỷ 20.



##### Hình 1: Cờ caro.

* Thông qua đồ án này, ứng dụng CARO ONLINE không chỉ đơn thuần là một nền tảng để chơi cờ Caro mà còn tích hợp nhiều tính năng thông minh như chế độ chơi đơn người với máy, chế độ chơi trực tuyến giữa người dùng, và các tùy chỉnh cá nhân hóa khác nhau để tạo ra trải nghiệm chơi tốt nhất. Báo cáo này sẽ trình bày quy trình phát triển ứng dụng, kiến thức cơ sở về thuật toán và công nghệ sử dụng, cùng với những thử nghiệm và đánh giá hiệu suất của ứng dụng. Hy vọng sẽ mang lại những kiến thức hữu ích và trải nghiệm thú vị cho người đọc về cả mảng lập trình ứng dụng và trò chơi logic.

## Mục tiêu đề tài

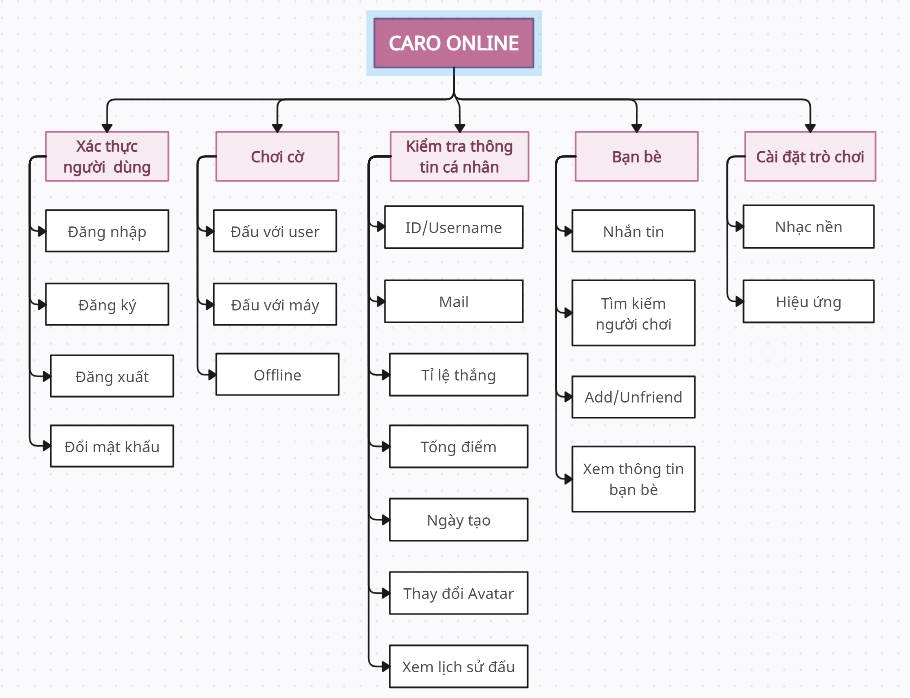
* Xây dựng được trò chơi có giao diện, mô hình bắt mắt và ấn tượng với người dùng
* Trò chơi thân thuộc và dễ dàng sử dụng đối với người chơi
* Hiểu và vận dụng được cách sử dụng Windows Form, C#

## Công nghệ sử dụng

* Sử dụng cơ sở dữ liệu Firebase
* Sử dụng ngôn ngữ C Sharp (C#)
* Môi trường thực hiện: ứng dụng Visual Studio
* Các công cụ có sẵn trong Toolbox (Visual Studio)

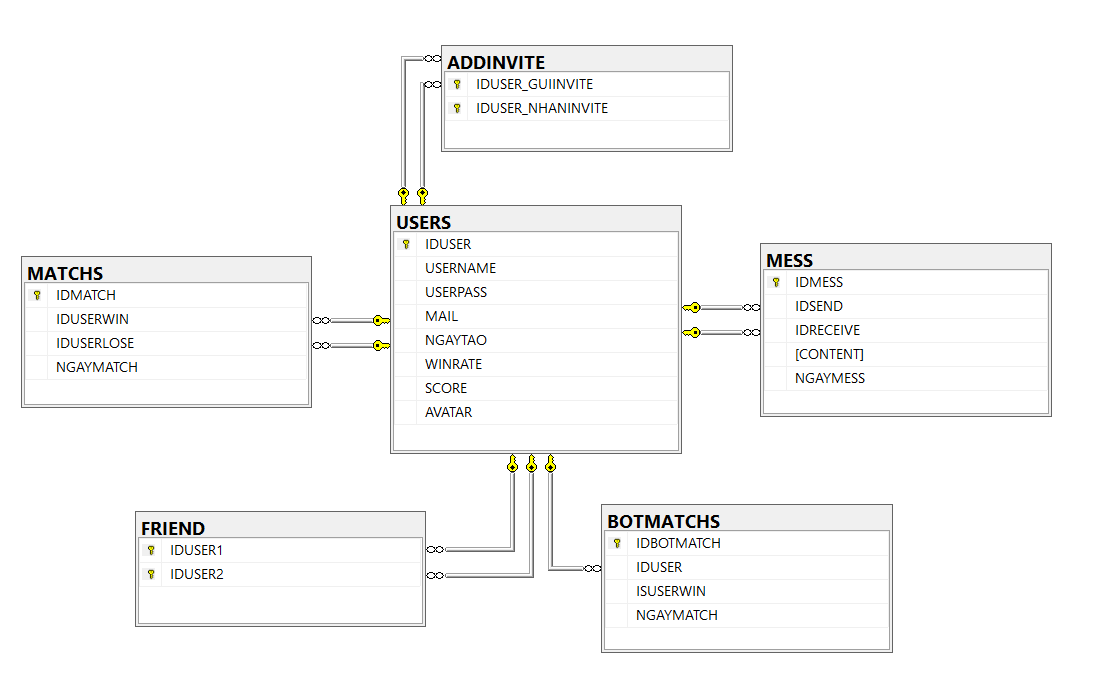
# Chương 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Sơ đồ phân rã chức năng



##### Hình 2: Sơ đồ phân rã chức năng.

## Cấu trúc Database

****

##### Hình 3: Cấu trúc Database.

**USERS (IDUSER, USERNAME, USERPASS, MAIL, NGAYTAO, WINRATE, SCORE, AVATAR)**

* IDUSER: ID của 1 User
* USERNAME: Tài khoản đăng nhập của User
* USERPASS: Mật khẩu đăng nhập
* MAIL: Gmail của User
* NGAYTAO: Ngày tạo tài khoản
* WINRATE: Tỉ lệ thắng của User
* SCORE: Tổng điểm (Số trận thắng)
* AVATAR: Lưu chỉ số của avatar của User

**FRIEND (IDUSER1, IDUSER2 )**

* IDUSER1: ID User
* IDUSER2: ID User

**MATCHS (IDMATCH, IDUSERWIN, IDUSERLOSE, NGAYMATCH)**

* IDMATCH: ID trận đấu
* IDUSERWIN: ID User (thắng)
* IDUSERLOSE: ID User (thua)
* NGAYMATCH: Ngày đấu

**BOTMATCHS (IDBOTMATCH, IDUSER, ISUSERWIN, NGAYMATCH)**

* IDBOTMATCH: ID trận đấu
* IDUSER: ID User
* IUSERWIN: Kiểm tra người chơi thắng hay thua
* NGAYMATCH: Ngày đấu

**MESS (IDMESS, IDSEND, IDRECEIVE, CONTENT, NGAYMESS)**

* IDMESS: ID tin nhắn
* IDSEND: ID User gửi tin
* IDRECEIVE: ID User nhận tin
* CONTENT: Nội dung tin nhắn
* NGAYMESS: Ngày nhắn tin

**ADDINVITE (IDUSER\_GUIINVITE, IDUSER\_NHANINVITE)**

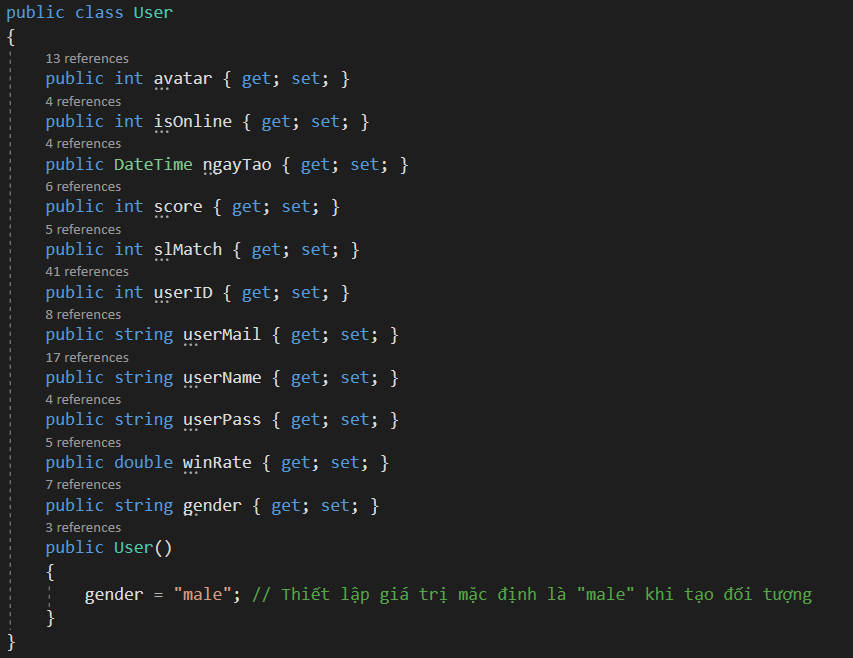
* IDUSER\_GUIINVITE: ID User gửi lời mời
* IDUSER\_NHANINVITE: ID User nhận lời mời

# Chương 3: HIỆN THỰC ĐỀ TÀI

## Kết nối database Firebase

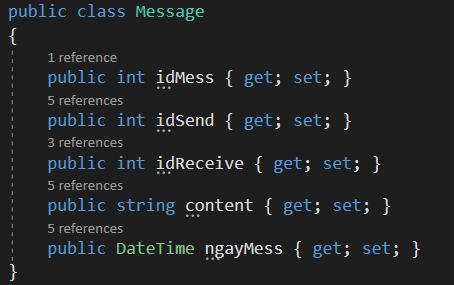
### Class Table

* Lưu trữ kiểu dữ các đối tượng của mỗi bảng:
* User:



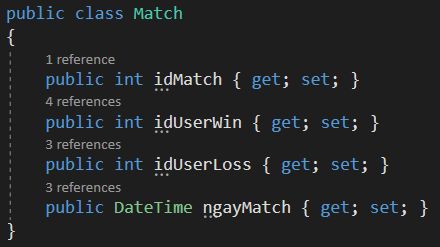
##### Hình 4: Class User.

* Message:



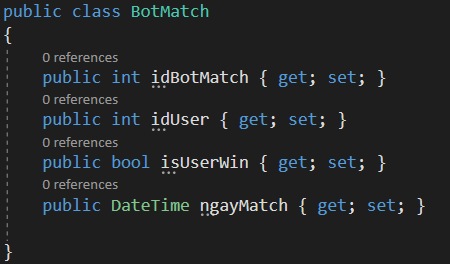
##### Hình 5: Class Message.

* Match:



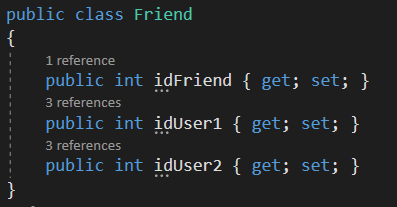
##### Hình 6:Class Match.

* BotMatch:



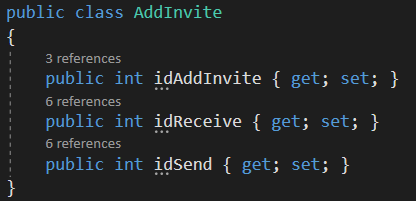
*Hình 7: Class BotMatch.*

* Friend:



##### Hình 8: Clas Friend.

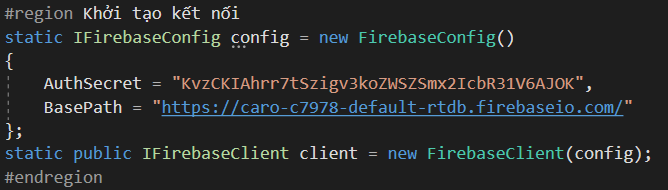
* AddInvite:



##### Hình 9: Class AdddInvite.

### Class DataBase

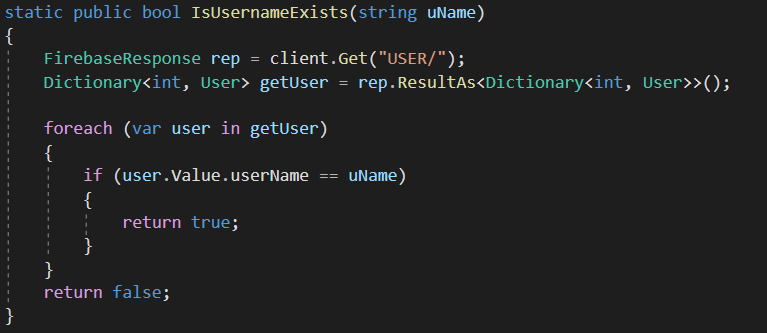
* Lưu trữ các hàm truy xuất, cập nhật Firebase
* Khởi tạo kết nối:



##### Hình 10:Khởi tạo kết nối FireBase.

#### IsUsernameExists(string uName):

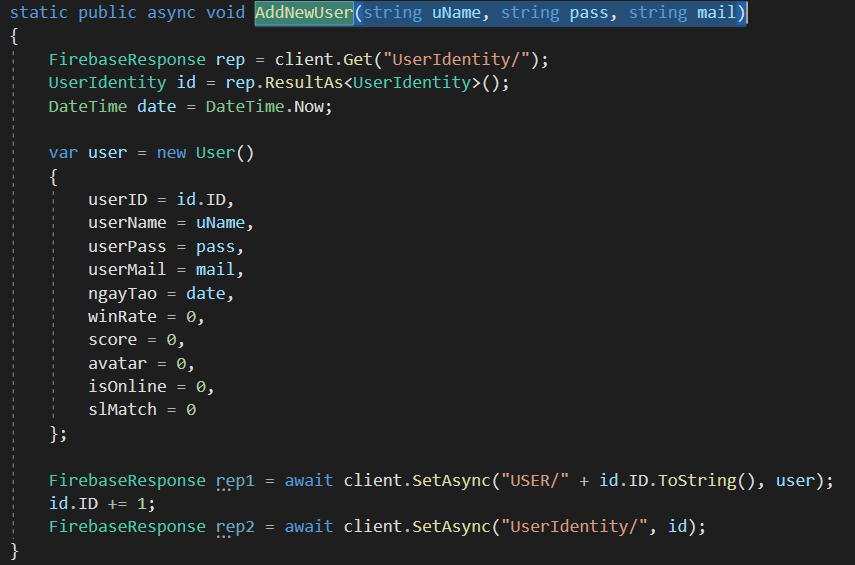
* Khi thực hiện đăng ký, kiểm tra xem username đăng ký đã tồn tại hay chưa



##### Hình 11: Hàm IsUserNameExists(string uName).

#### AddNewUser(string uName, string pass, string mail):

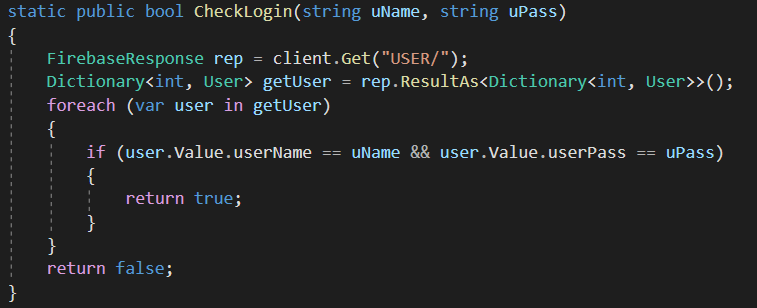
* Add 1 user mới lên Firebase khi đăng ký tài khoản thành công

****

##### Hình 12: Hàm AddNewUser(string uName, string pass, string mail).

#### CheckLogin(string uName, string uPass):

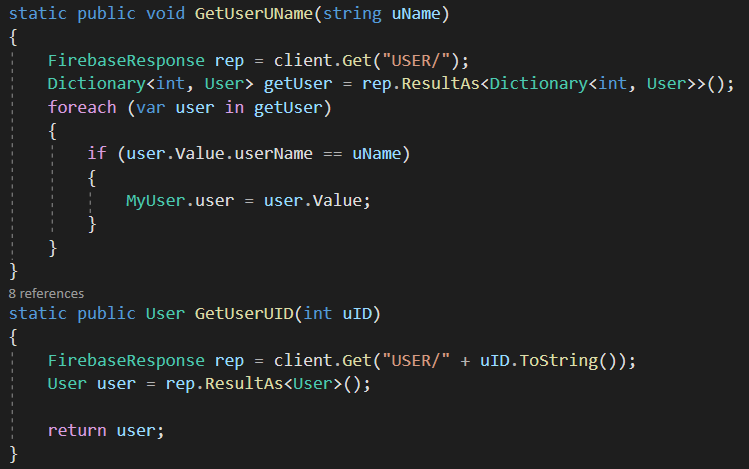
* Kiểm tra username, password mỗi khi đăng nhập



##### Hình 13: Hàm CheckLogin(string uName, string uPass).

#### GetUser():

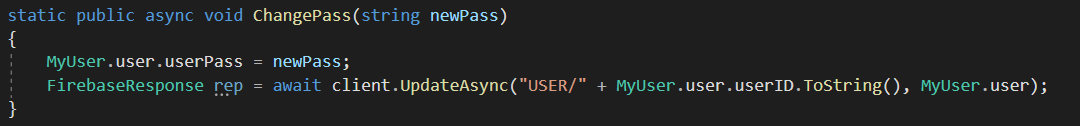
* Trả về 1 đối tượng User từ userName hoặc userId

****

##### Hình 14: Hàm GetUser().

#### ChangePass(string newPass):

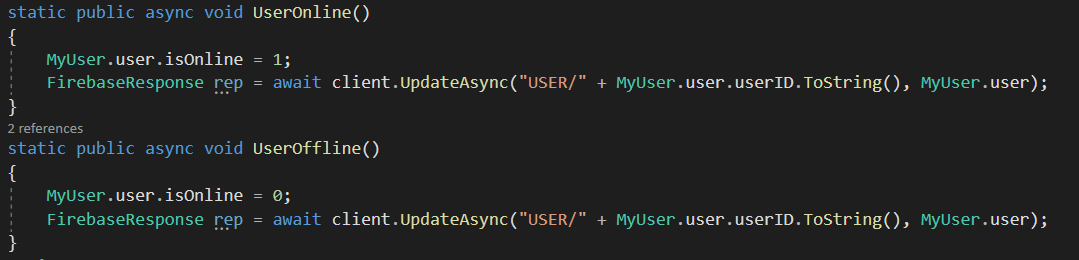
* Thay đổi mật khẩu trong Form Privacy



##### Hình 15: Hàm ChangePass(string newPass).

#### UserOnline(); UserOffline():

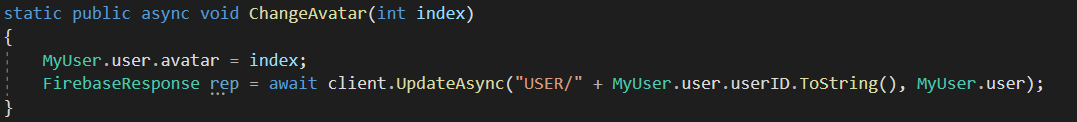
* Kiểm tra 1 user đang online hay offline
* Sau khi thực hiện đăng nhập sẽ gọi hàm UserOnline()
* Sau khi đăng xuất hoặc thoát ứng dụng sẽ thực hiện hàm UserOffline()



##### Hình 16: Hàm UserOnline(); UserOffline().

#### ChangeAvatar(int index):

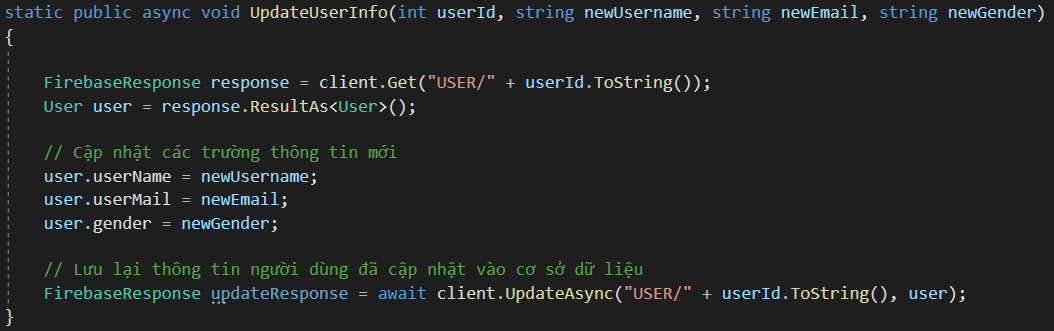
* Thay đổi ảnh đại diện của user
* Mỗi ảnh được đánh dấu bằng số (index)
* Truy xuất đến ảnh bằng dòng code: **Image.FromFile($"Resources\\{index}.png");**



##### Hình 17: Hàm ChangeAvatar(int index).

#### UpdateUserInfo():

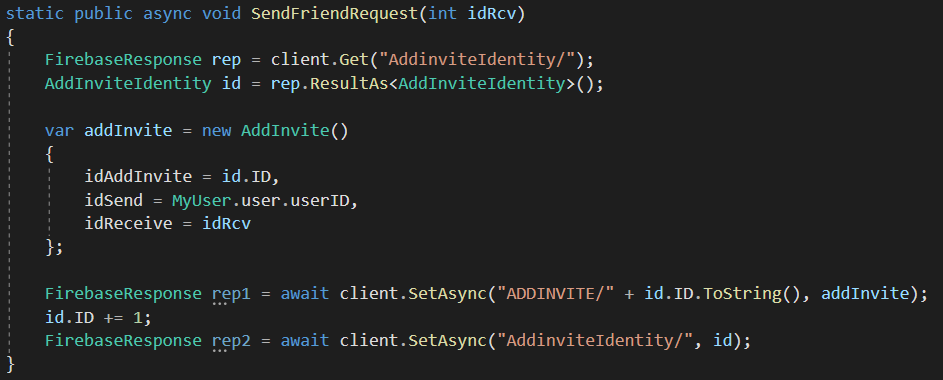
* Cập nhật lại thông tin của user lên database



##### Hình 18: Hàm UpdateUserInfo().

#### SendFriendRequest(int idRcv):

* Tạo lời mời kết bạn tới user có **id = idRcv**



##### Hình 19: Hàm SendFriendRequest(int idRcv).

#### AcceptFriendRequest(int idSend):

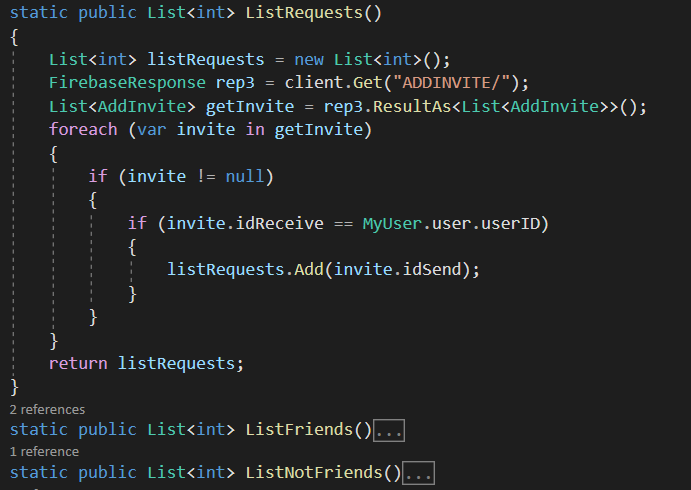
* Chấp nhận lời mời kết bạn của user có id là **idSend**
* Sau đó sẽ xóa đi lời mời kết bạn



##### Hình 20: Hàm AcceptFriendRequest(int idSend).

#### Hàm lấy ra danh sách:

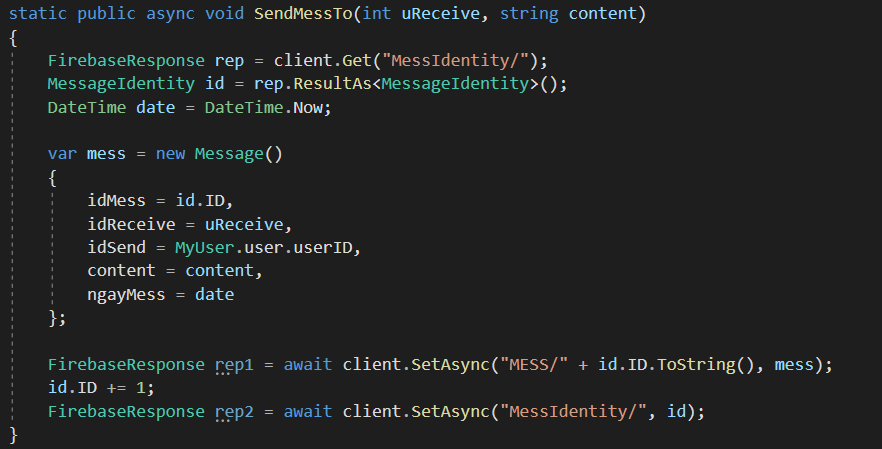
* Trả về 1 List<int> danh sách các lời mời kết bạn, danh sách bạn bè và danh sách các người chơi khác
* Duyệt từng phần tử trong List<int> với mỗi phần tử là userId rồi thêm vào danh sách trong Giao diện nhắn tin (Form Message)



##### Hình 21: Hàm ListRequests().

#### SendMessTo(int uReceive, string content):

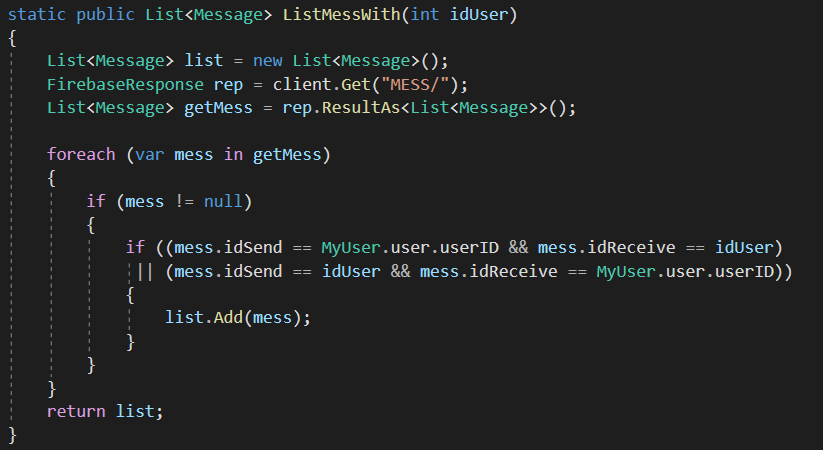
* Lưu thông tin tin nhắn lên database



##### Hình 22: Hàm SendMessTo(int uReceive, string content) .

#### ListMessWith(int idUser):

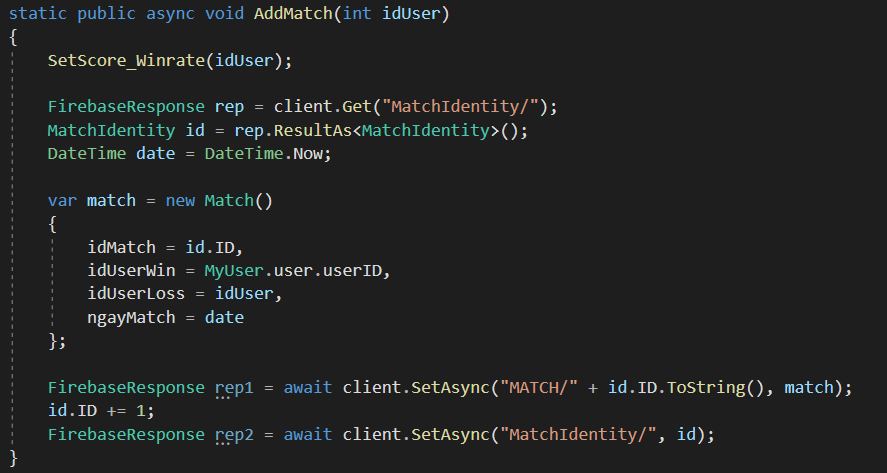
* Duyệt danh sách tin nhắn của người chơi với người chơi khác



##### Hình 23: Hàm ListMessWith(int idUser).

#### AddMatch(int idUser):

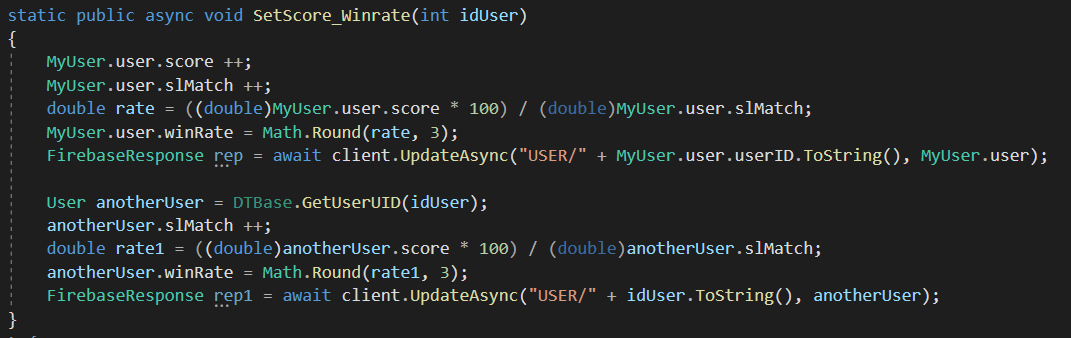
* Thêm 1 Match lên database mỗi khi kết thúc ván đấu



##### Hình 24: Hàm AddMatch(int idUser).

#### SetScore\_Winrate(int idUser):

* Cập nhật điểm(score), số trận đấu(slMatch) và tỉ lệ thắng(winRate) của từng người chơi lên database



##### Hình 25: Hàm SetScore\_Winrate(int idUser).

#### HistoryMatch():

* Duyệt danh sách lịch sử đấu của người chơi rồi thêm vào Giao diện cá nhân(Form Profile)

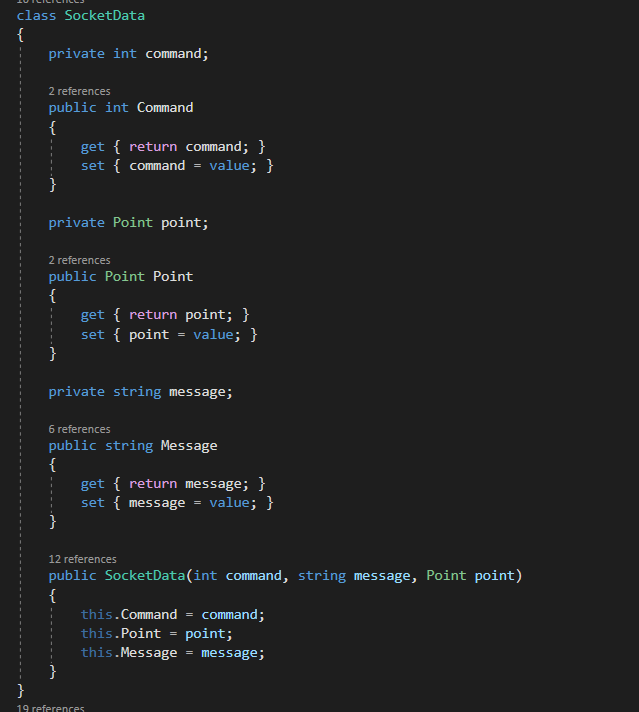


##### Hình 26: Hàm HistoryMatch().

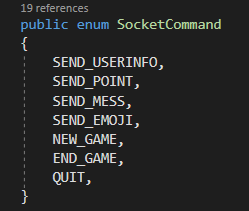
## Socket

### Class SocketData:

* Kiểu dữ liệu sẽ truyền đi giữa các socket



##### Hình 27: Class SockeData.

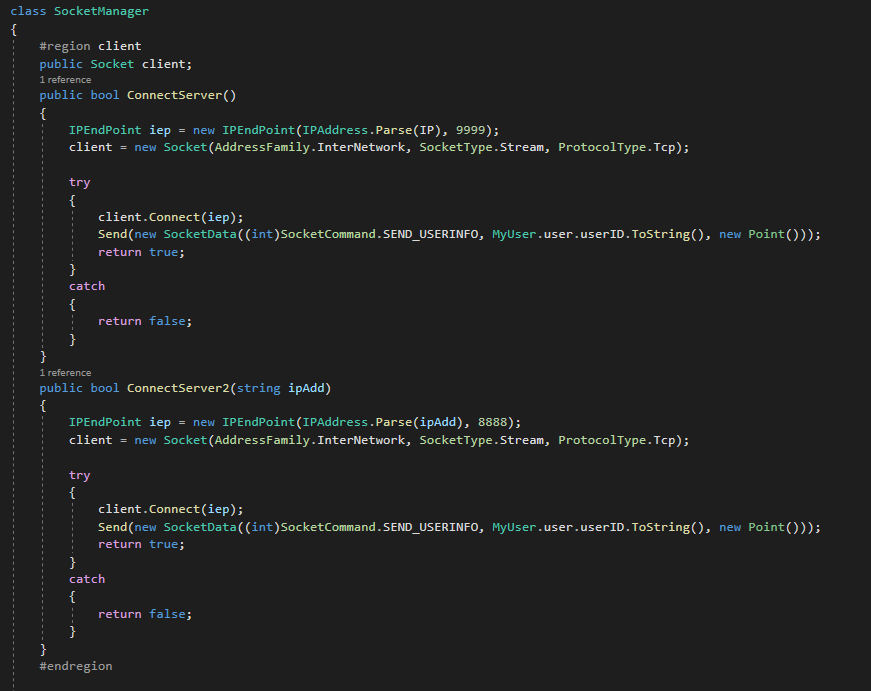


##### Hình 28: Enum SocketCommand.

* Command : là một enum lưu các kiểu truyền dữ liệu:
* SEND\_USERINFO : truyền thông tin về người chơi khi bắt đầu khởi tạo kết nối
* SEND\_POINT : truyền 1 vị trí quân cờ
* SEND\_MESS : Truyền tin nhắn
* SEND\_EMOJI : Truyền 1 icon
* NEW\_GAME : Truyền thông tin rằngmuốn bắt đầu một game mới
* END\_GAME : truyền thông tin kết thúc game (thắng thì kết thúc game)
* QUIT : truyền thông tin rằng người chơi đã thoát
* Point : sử dụng để truyền vị trí quân cờ khi đánh, nếu ko truyền quân cờ thì truyền 1 new Point()
* Message: sử dụng khi truyền 1 tin nhắn hoặc emoji

### Class SocketManager:

* Hàm khởi tạo kết nối, truyền nhận dữ liệu giữa các socket



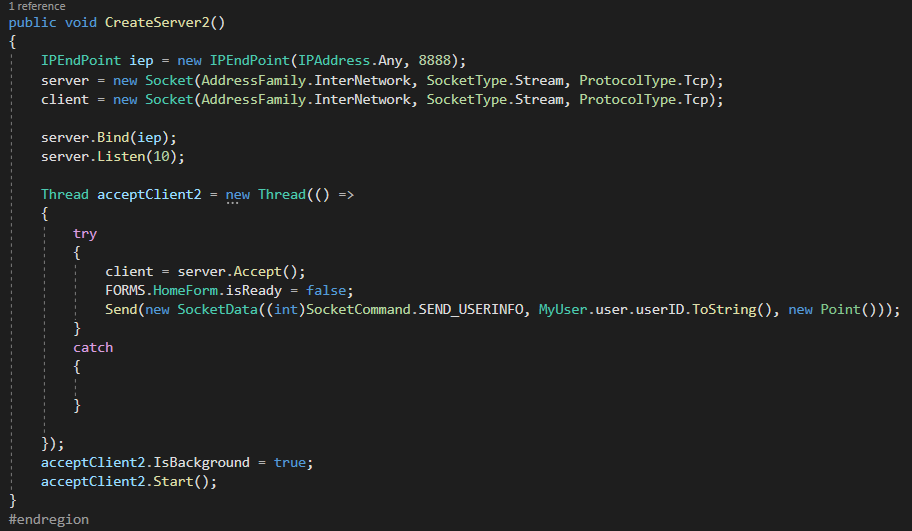
##### Hình 29: Hàm Connect Server.

* Hàm ConnectServer() : kết nối đến server trong chế độ tìm người chơi,sử dụng port 9999
* Hàm ConnectServer2(ipAdd) : kết nối đến server khi có bạn bè mời chơi, ipAdd là ip của người gửi lời mời,port 8888



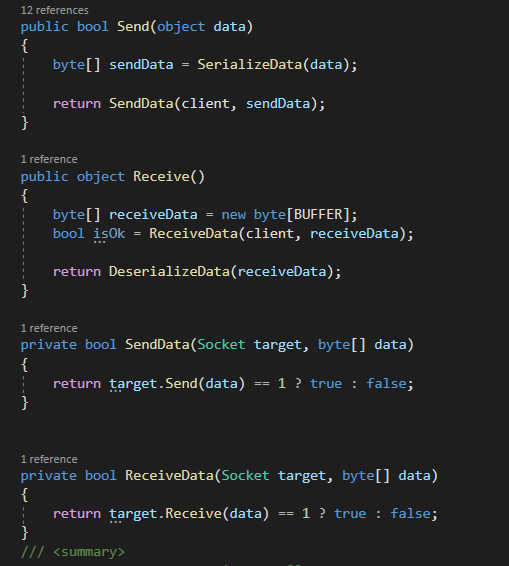
##### Hình 30: Hàm Create Server.

* Hàm CreateServer(): tạo server trong chế độ tìm người chơi, sử dụng port 9999



##### Hình 31: Hàm Create Server 2.

* Hàm CreateServer2(): tạo phòng để mời bạn bè, sử dụng port 8888

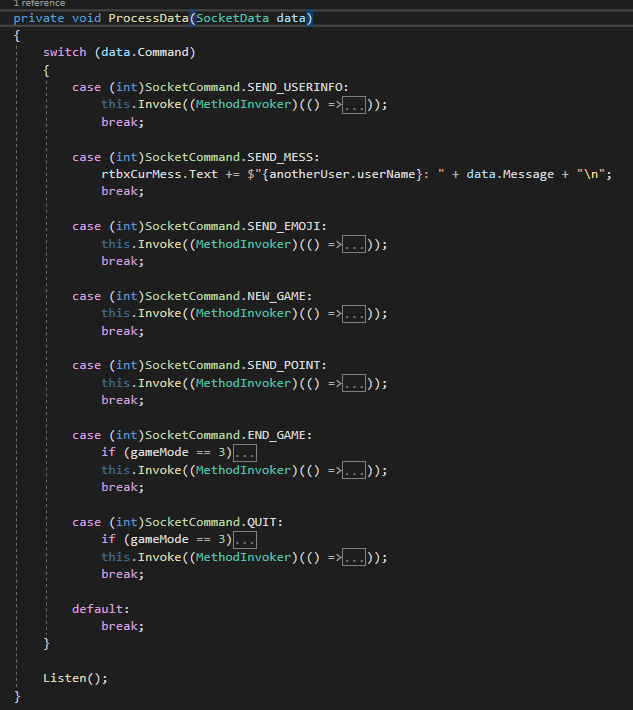


##### Hình 32: Hàm gửi nhận dữ liệu.

* Các hàm gửi nhận dữ liệu

### Truyền dữ liệu giữa các socket:

* Khi nhận được một SocketData, sẽ có hàm để xử lý theo từng kiểu dữ liệu nhận được

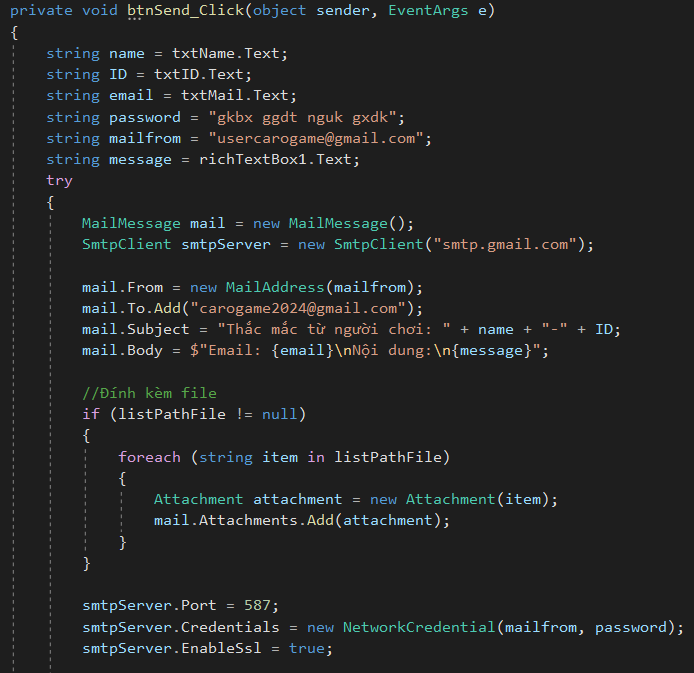


##### Hình 33: Hàm ProcessData().

* SEND\_USERINFO : Sẽ hiện avatar của user đó, lưu tag là id của user đó vào picturebox để khi bấm vào có thể xem thông tin.
* SEND\_MESS: Add nội dung của tin nhắn đó vào richtexbox.
* Send emoji: Hiện emoji lên trong vài giây (khi nhận sẽ hiện lên rồi kích hoạt timmer để ẩn emoji đi).
* NEW\_GAME: Khi kết thúc game sẽ hiện 1 button để hỏi người chơi có muốn chơi tiếp hay không, khi click sẽ gửi 1 new game, bên kia nhận 1 new game sẽ hiện lên thông báo ,nếu bấm yes thì chơi tiếp, no thì sẽ thoát ra.
* SEND\_POINT: Khi đánh 1 ô , sẽ đặt quân cờ tại ô đó, đồng thời gửi 1 send point cho người chơi kia, khi nhận 1 send point thì sẽ đặt quân cờ của người chơi còn lại tại point đó.
* END\_GAME: Khi người chơi thoát mà chưa bắt đầu ván mới thì sẽ gửi 1 end game, nhận 1 end game thì sẽ thông báo kết thúc game và đóng kết nối.
* QUIT: Khi thoát ra khỏi phòng trong khi đang chơi sẽ gửi 1 quit, nhận 1 quit sẽ thông báo rằng người chơi kia đã thoát , người chơi sẽ chiến thắng ván đó do người kia thoát giữa và đóng kết nối.

## Gửi nhận mail

Sử dụng SMTP để gửi email report. Tài khoản [usercarogame@gmail.com](mailto:usercarogame@gmail.com) sẽ được dùng để gửi phản hồi của người chơi đến gmail [carogame@gmail.com](mailto:carogame@gmail.com) cùng với các thông tin của người chơi.



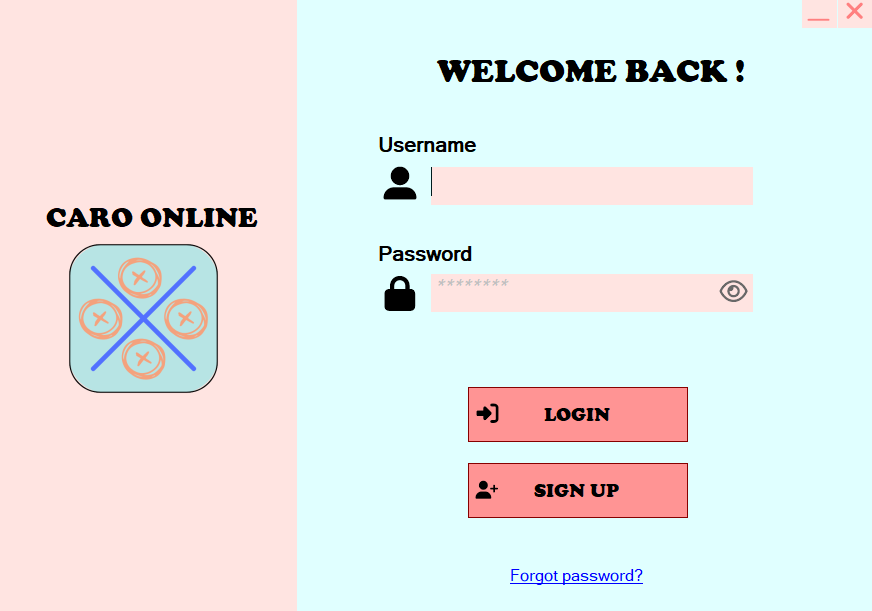
##### Hình 34: Gửi mail.



##### Hình 35: Gửi mail (tt).

# Chương 4: THỰC NGHIỆM ĐỀ TÀI

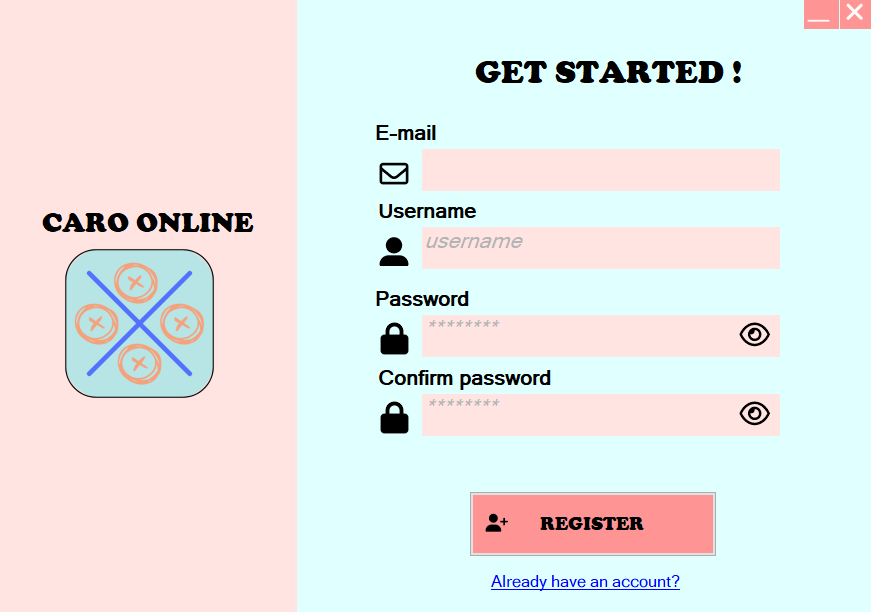
## Giao diện đăng nhập (Form Login)

****

##### Hình 36. Form Login

* tbxUsername: Nhập vào Username
* tbxPassword: Nhập vào password
* btnLogin: Kiểm tra Username/Password
* Nếu đúng: Chuyển sang Form Main
* Sai: Yêu cầu nhập lại
* btnSignup: Chuyển sang Form Sign Up
* btnForgotPass: Chuyển sang Forrm Forgot Password

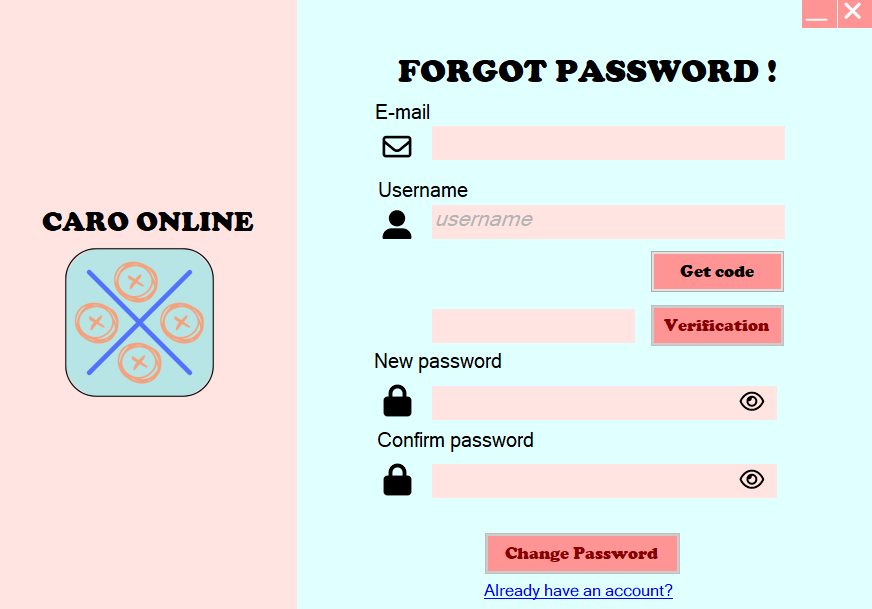
## Giao diện đăng ký (Form Sign Up)

****

##### Hình 37. Form Sign Up.

* tbxMail: Nhập Gmail của người dùng
* tbxUsername: Nhập Username
* tbxPass: Nhập password
* tbxConfirmPass: Nhập lại password vừa rồi
* tbxRegister: Kiểm tra đăng ký
* Nếu Username đăng ký đã tồn tại: Nhập lại username
* Nếu chưa: Tạo 1 tài khoản cho User, chuyển sang Form Login
* btnHaveAnAccount: Chuyển sang Form Login

## Giao diện quên mật khẩu (Form Forgot Password)

****

##### Hình 38. Form Forgot Password

* tbxMail: Nhập Gmail của tài khoản đã đăng ký
* tbxUsername: Nhập Username
* btnGetcode: Gửi code qua Gmail đã nhập trong tbxMail
* btnVerification: Nhập code hệ thống đã gửi về Gmail
* Kiểm tra nếu đúng: Sẽ cho đổi lại mật khẩu
* Nếu sai: Yêu cầu nhập lại code hoặc nhận lại code mới
* btnChangePass: Cập nhật lại mật khẩu của User, sau đó chuyển sang Form Login
* btnHaveAnAccount: chuyển sang Form Login

## Giao diện chính (Form Main)

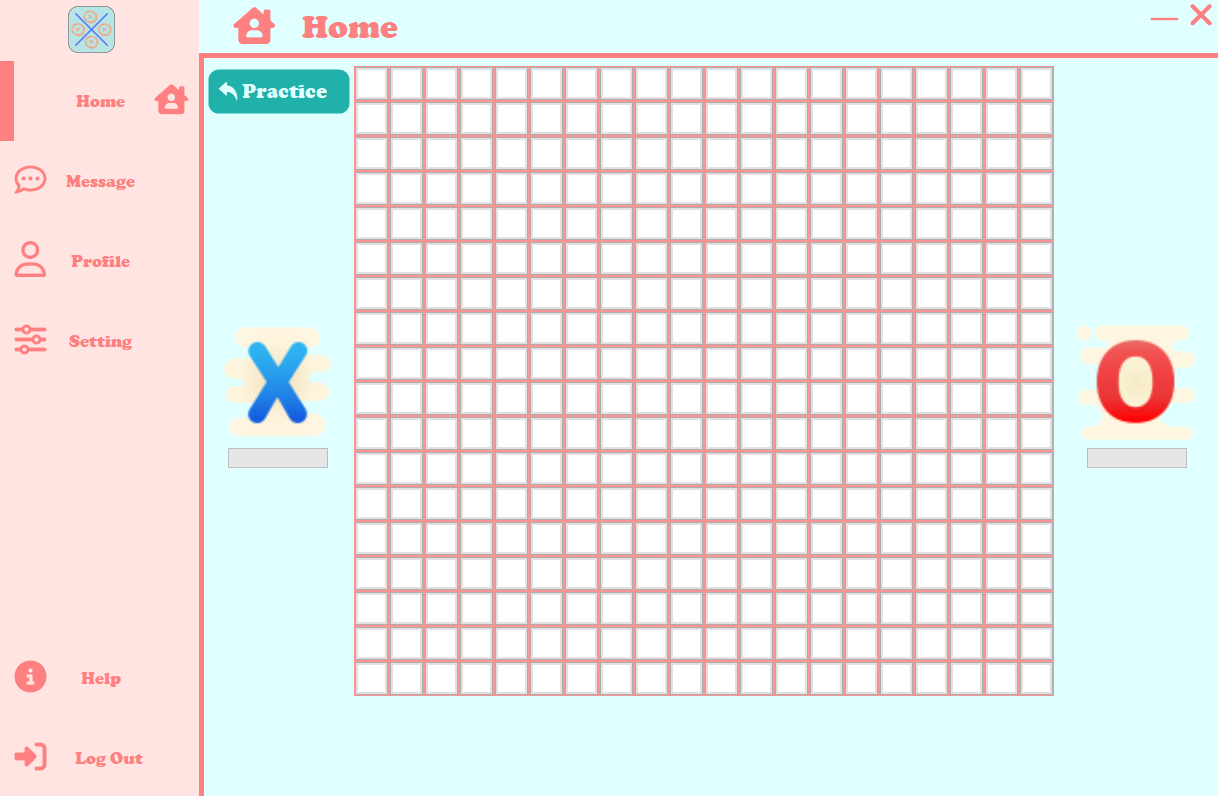


##### Hình 39. Form Main

* btnHome: Hiện Form Home
* btnMessage: Hiện Form Message
* btnProfile: Hiện Form Profile
* btnSetting: Hiện Form Setting
* btnHelp: Hiện Form Help
* btnLogOut: Chuyển sang Form Login

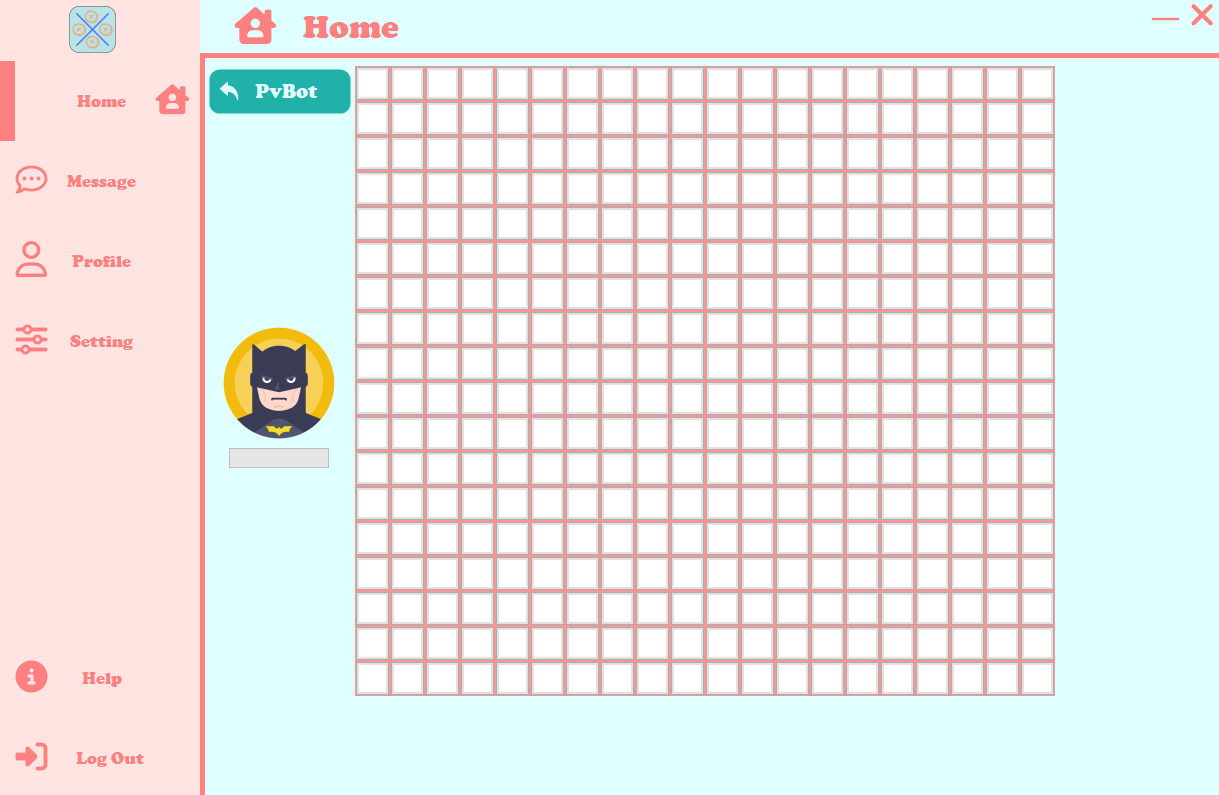
## Giao diện chơi game (Form Home)

* btnPractice: Tự luyện tập



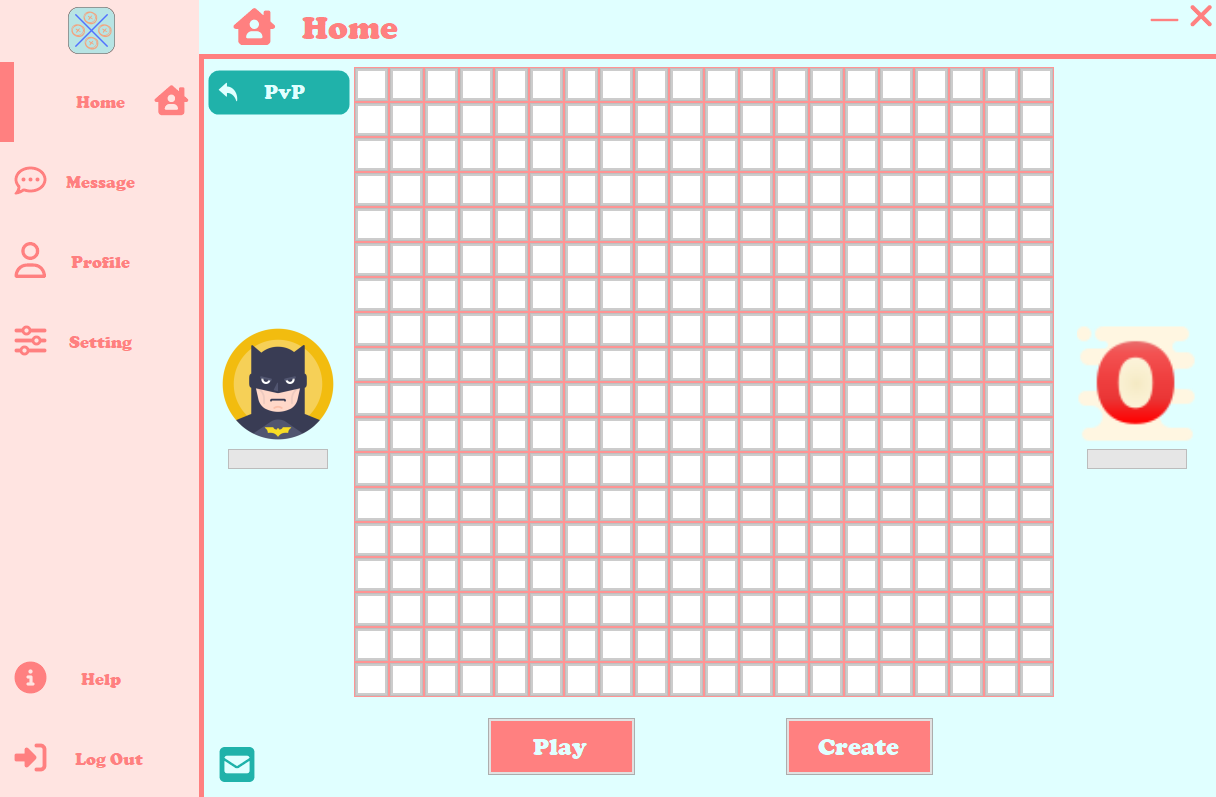
##### Hình 40.Chế độ Practice.

* btnPvBot: Đấu với máy



##### Hình 41.Chế độ PvBot.

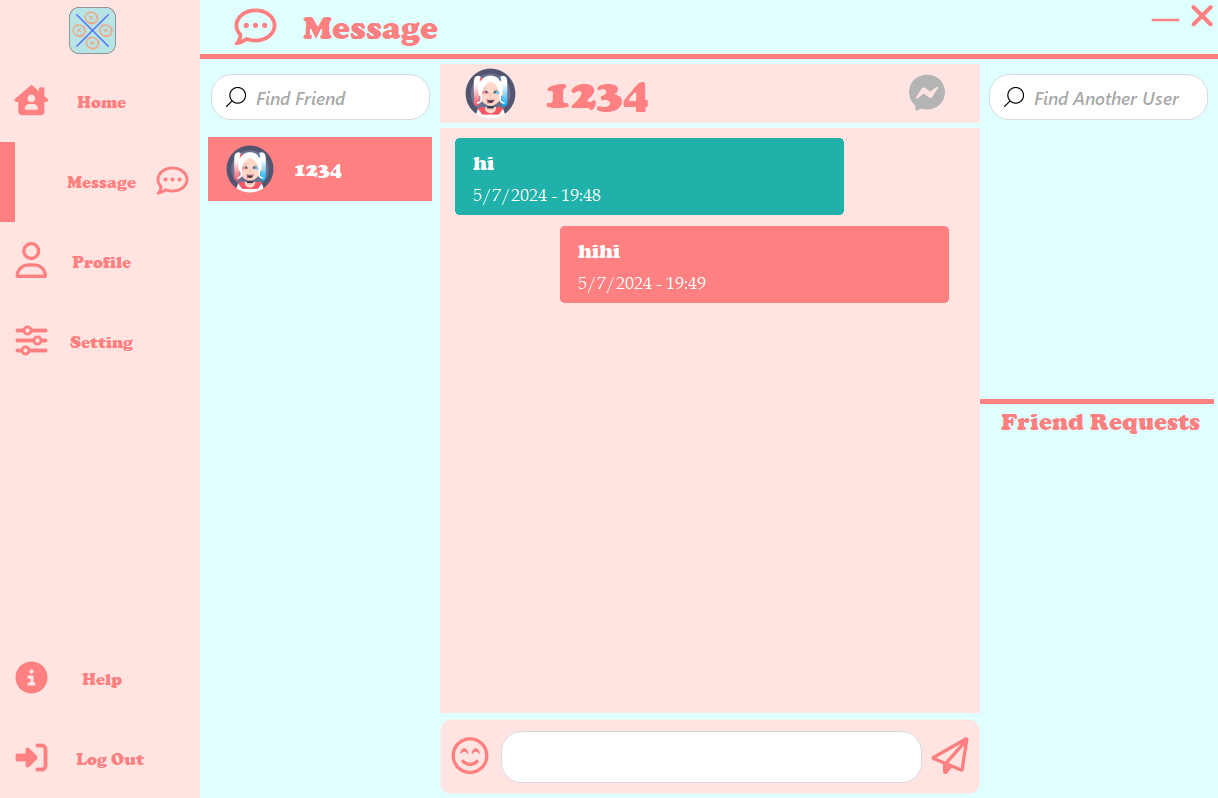
* btnPvP: Đấu với người chơi khác



##### Hình 42.Chế độ PvP.

* pnChessboard: Tạo bàn cờ từ các button

## Giao diện nhắn tin (Form Message)



##### Hình 43. Form Message

* Tự động duyệt danh sách Friend, Another User, Friend Requests
* tbxFindFriend: Tìm kiếm bạn bè
* tbtFindAnotherUser: Tìm kiếm người chơi khác
* btnFriend (các button hiện trong danh sach Friend):
* Right click: Hiện 1 contextmenustrip cho phép xem thông tin hoặc xóa kết bạn
* Left click: Nhắn tin với User, tự động duyệt tin nhắn trước đó trong pannelShowMessage
* btnAnotherUser (các button hiện trong danh sách Another User):
* Right click: Hiện 1 contextmenustrip cho phép xem thông tin hoặc kết bạn
* btnFriendRequest (các button hiện trong danh sách Friend Requests):
* Right click: Hiện 1 contextmenustrip cho phép xem thông tin
* Left click: Hiện MessageBox hỏi có chấp nhận lời mời không
* btnSendMessage: Gửi tin nhắn trong tbxMessage đến User
* btnSendIcon: Gửi icon đến User

## Giao diện cá nhân (Form Profile)



##### Hình 44. Form Profile

* Tự động hiện thông tin, avatar người chơi vào các textbox
* Tự động duyệt danh sách lịch sử đấu, thông tin trận đấu của người chơi

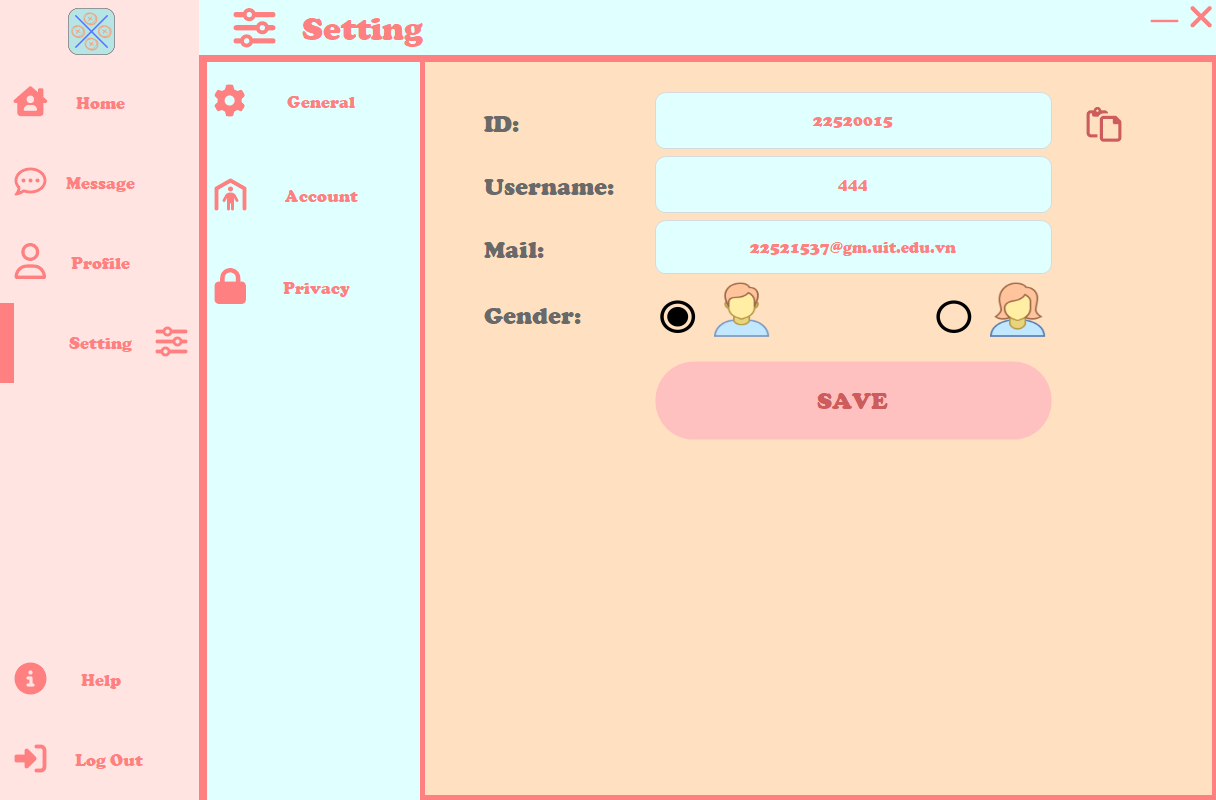
## Giao diện cài đặt (Form Setting)

* General:



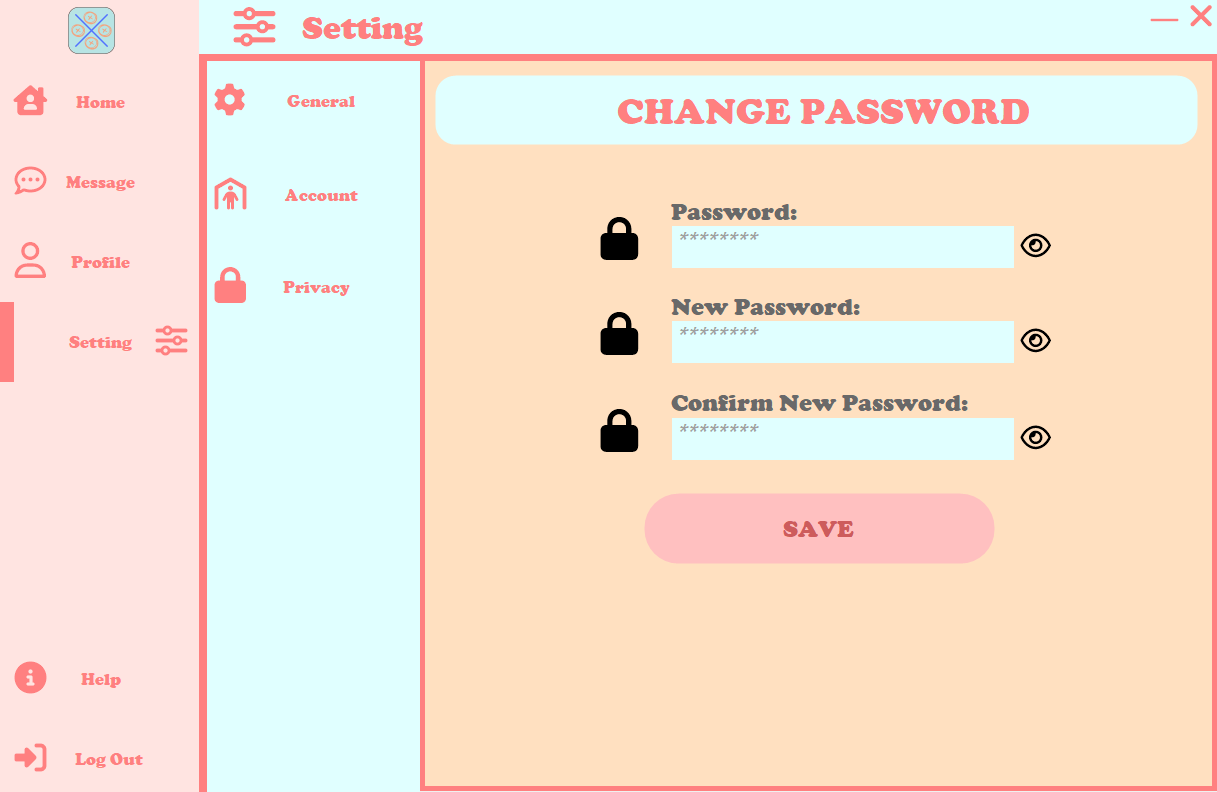
##### Hình 45. Form General

* btnOffMusic: Tắt nhạc nền
* btnOnMusic: Bật nhạc nền
* btnOffEffect: Tắt hiệu ứng
* btnOnEffect: Bật hiệu ứng
* btnSave: Phát nhạc nền đã chọn trong comboboxSong
* Account:



##### Hình 46. Form Account

* Privacy:



##### Hình 47. Form Privacy.

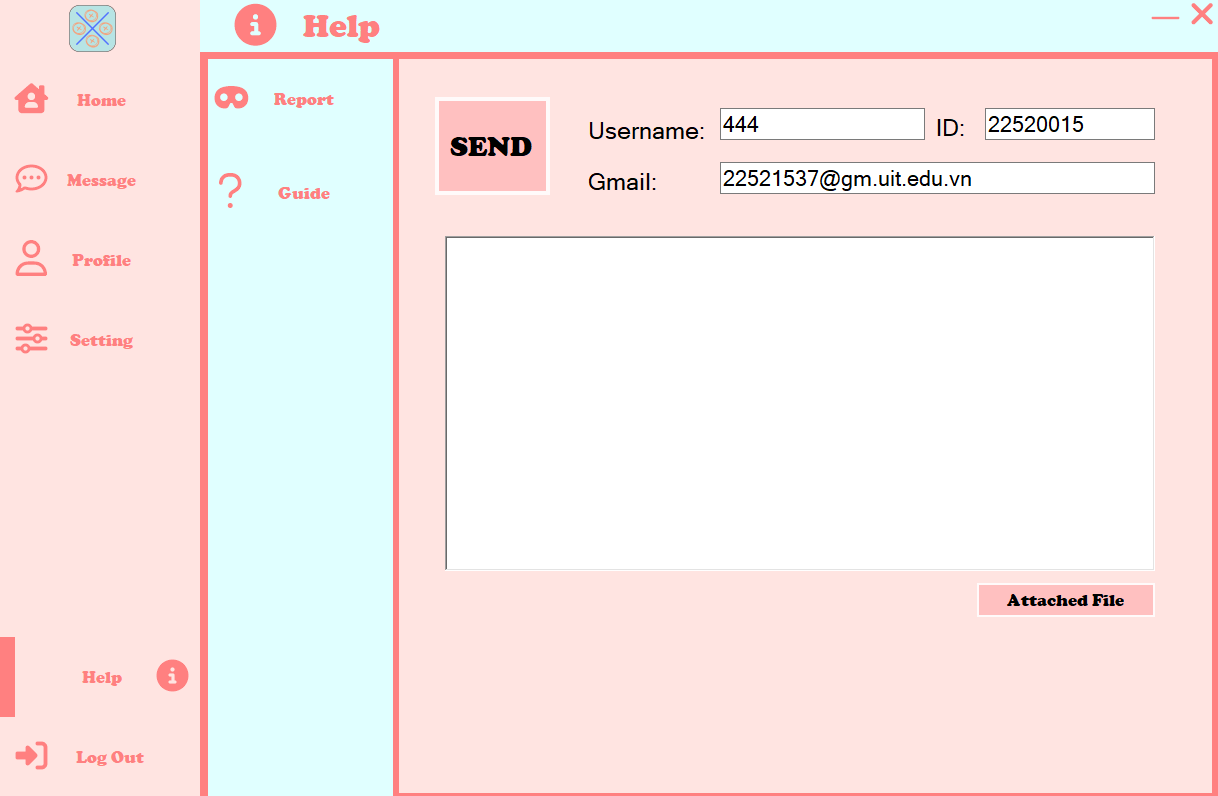
* tbxPassword: Nhập mật khẩu hiện tại
* tbxNewPassword: Nhập mật khẩu mới
* tbxConfirmNewPass: Nhập lại mật khẩu mới
* btnSave: Kiểm tra mật khẩu hiện tại và mật khẩu nhập lại có đúng không, nếu đúng sẽ cập nhật lại mật khẩu mới

## Giao diện Hướng dẫn (Form Guide)



##### Hình 48. Form Guide.

## Giao diện Báo cáo (Form Report)



##### Hình 49. Form Report.

* btnSend: Gửi mail(các câu hỏi, báo cáo lỗi game, góp ý…) tới Server.
* btnAttachedFile: Đính kèm thông tin hình ảnh hoặc file.

# Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Nhóm đã xây dựng thành công một ứng dụng game Caro theo kiến trúc client – server với các tính năng xác thực, phân quyền, kết nối Internet và cơ sở dữ liệu riêng sử dụng Firebase.

Tính năng chính

* Xác thực và phân quyền: Ứng dụng yêu cầu người dùng đăng nhập, giúp bảo vệ thông tin cá nhân và đảm bảo chỉ những người có quyền mới có thể truy cập vào các tính năng nhất định của trò chơi.
* Lưu trữ dữ liệu trên Firebase: Firebase được sử dụng làm cơ sở dữ liệu chính để lưu trữ thông tin về người dùng, lịch sử trận đấu và các dữ liệu khác liên quan đến game. Điều này giúp dữ liệu luôn được cập nhật và có thể truy cập từ bất kỳ đâu có kết nối Internet.
* Đa truy cập: Ứng dụng cho phép nhiều người dùng truy cập và chơi game cùng lúc, hỗ trợ tính năng đấu trực tuyến giữa các người chơi.
* Kết bạn và nhắn tin: Các người chơi có thể kết bạn và nhắn tin với nhau trực tiếp trên ứng dụng. Các tin nhắn này sẽ được lưu trữ trên Firebase, giúp người dùng có thể xem lại lịch sử trò chuyện bất cứ lúc nào.

Lợi ích của ứng dụng:

* Cải thiện trải nghiệm người dùng: Giúp người chơi có thể giao lưu, kết nối với nhau dễ dàng hơn, tạo ra một cộng đồng game Caro trực tuyến thân thiện và sôi động.
* Bảo mật và tiện lợi: Với việc lưu trữ dữ liệu và tin nhắn trên Firebase, người dùng có thể yên tâm về tính bảo mật và sự tiện lợi khi dữ liệu luôn được cập nhật và đồng bộ.

Cải thiện, hướng phát triển: Ứng dụng vẫn còn nhiều hạn chế và thiếu sót, nhóm dự định sẽ triển khai trong tương lai:

* Cải thiện giao diện, hiệu suất ứng dụng
* Có thể chơi game kết nối khác LAN
* Cập nhật thêm tính năng: tích hợp thêm nạp tiền; có thể dùng xu để cược mỗi ván đấu hoặc mua vật phẩm; upload ảnh avatar…..

# TÀI LIỆU THAM KHẢO

# BẢNG PHÂN CHIA CÔNG VIỆC

| **Họ tên** | MSSV | **Nội dung phụ trách** | Đánh giá |
| --- | --- | --- | --- |
| Mai Kim Trinh | 22521537 | - Giao diện màn hình chính  - Thiết kế giao diện tìm kiếm người chơi thông qua ID,username  - Thiết kế giao diện cài đặt trò chơi ( Âm thanh, Thông báo, Ngôn ngữ, ánh sáng, ..)  - Code chức năng duyệt danh sách bạn bè  - Code chức năng AddFriend, UnFriend, Show info người chơi  - Chức năng gửi report  - Viết báo cáo | 100% |
| Đào Nhật Duy | 22520312 | - Thiết kế giao diện người chơi bao gồm thông tin, ID, số lượng bạn bè, cấp độ, tỉ lệ thắng  - Thiết kế giao diện đổi mật khẩu, ranking  - Thiết kế giao diện lịch sử trận đấu  - Code giao diện trang cá nhân người dùng, chức năng đổi avatar  - Code chức năng xem lịch sử trận đấu  - Code chức năng quên mật khẩu | 100% |
| Lê Nguyễn Minh Hiếu | 22520436 | - Thiết kế giao diện chơi game  - Thiết kế giao diện kết bạn và nhắn tin  - Thiết kế giao diện cập nhật thông tin cá nhân  - API Chơi game, mời bạn bè chơi  - Code chức năng tìm kiếm ID, kết bạn  - Chức năng local cache Database | 100% |
| Lê Thị Tú Uyên | 22521639 | - Thiết kế giao diện đăng ký và đăng nhập  - Thiết kế giao diện thông báo  - Thiết kế giao diện danh sách bạn bè  - Code Cài đặt, âm thanh  - Code chắc năng đăng nhập, đăng kí  - Code chức năng tin nhắn | 100% |