**MỤC LỤC**

[**PHẦN I. MỞ ĐẦU** 3](#_Toc182650053)

[1. Giới thiệu ứng dụng 3](#_Toc182650054)

[2. Phân tích yêu cầu ứng dụng 4](#_Toc182650055)

[2.1. Yêu cầu chức năng phía client 4](#_Toc182650056)

[2.2. Yêu cầu chức năng phía server 4](#_Toc182650057)

[2.3. Yêu cầu phi chức năng 5](#_Toc182650058)

[3. Bảng phân công chức năng 6](#_Toc182650059)

[**PHẦN II. PHÂN TÍCH THIẾT KẾ TỔNG QUAN ỨNG DỤNG** 6](#_Toc182650060)

[1. Kiến trúc tổng quan 6](#_Toc182650061)

[2. Sơ đồ khối chức năng của Server và Client 7](#_Toc182650062)

[3. Sơ đồ Use case tổng quan 7](#_Toc182650063)

[**PHẦN III. PHÂN TÍCH THIẾT KẾ CHI TIẾT CHỨC NĂNG CÁ NHÂN** 8](#_Toc182650064)

[1. Sơ đồ Use case chi tiết 8](#_Toc182650065)

[2. Biểu đồ lớp 9](#_Toc182650066)

[3. Biểu đồ tuần tự 10](#_Toc182650067)

[3.1. Biểu đồ tuần tự kết nối Client với Server 10](#_Toc182650068)

[3.2. Biểu đồ tuần tự xem bảng xếp hạng 11](#_Toc182650069)

[3.3. Biểu đồ tuần tự xem thông tin người chơi khác 12](#_Toc182650070)

[3.4. Biểu đồ tuần tự xem lịch sử đấu 13](#_Toc182650071)

[3.5. Biểu đồ tuần tự đổi avatar 14](#_Toc182650072)

[3.6. Biểu đồ tuần tự trò chuyện 15](#_Toc182650073)

[4. Biểu đồ quan hệ 16](#_Toc182650074)

[**PHẦN IV. KẾT QUẢ ỨNG DỤNG** 16](#_Toc182650075)

[1. Kiến trúc ứng dụng 16](#_Toc182650076)

[2. Cài đặt và triển khai 17](#_Toc182650077)

[3. Kết quả thực hiện 19](#_Toc182650078)

[4. Tài liệu tham khảo 23](#_Toc182650079)

[5. Source code 23](#_Toc182650080)

**DANH MỤC HÌNH ẢNH**

[Hình 1 Kiến trúc tổng quan 6](#_Toc182650263)

[Hình 2 Sơ đồ khối các chức năng 7](#_Toc182650264)

[Hình 3 Sơ đồ Use case tổng quan 8](#_Toc182650265)

[Hình 4 Sơ đồ Use case chi tiết 9](#_Toc182650266)

[Hình 5 Biểu đồ lớp 9](#_Toc182650267)

[Hình 6 Biểu đồ tuần tự kết nối Client với Server 10](#_Toc182650268)

[Hình 7 Biểu đồ tuần tự cho chức năng xem bảng xếp hạng 11](#_Toc182650269)

[Hình 8 Biểu đồ tuần tự xem thông tin người chơi 12](#_Toc182650270)

[Hình 9 Biểu đồ tuần tự chức năng xem lịch sử đấu 13](#_Toc182650271)

[Hình 10 Biểu đồ tuần tự chức năng đổi avatar 14](#_Toc182650272)

[Hình 11 Biểu đồ tuần tự chức năng trò chuyện 15](#_Toc182650273)

[Hình 12 Biểu đồ quan hệ 16](#_Toc182650274)

[Hình 13 Kiến trúc ứng dụng 16](#_Toc182650275)

**DANH MỤC BẢNG**

[Bảng 1 Bảng phân công chức năng cá nhân 6](#_Toc182650376)

1. **MỞ ĐẦU**
2. **Giới thiệu ứng dụng**

"Rút Bài May Mắn" là một tựa game nhiều người chơi trực tuyến, phát triển bằng Java theo kiến trúc client-server, mang đến trải nghiệm gay cấn và hấp dẫn. Người chơi sẽ tham gia vào những ván bài đối kháng, lần lượt rút các lá bài từ bộ bài chung để giành được điểm số cao nhất. Trò chơi kết hợp yếu tố may mắn và chiến thuật, thách thức người chơi đánh bại đối thủ trong mỗi ván đấu. Với khả năng kết nối bạn bè và chơi cùng nhau, game hứa hẹn mang lại những khoảnh khắc giải trí sôi động và thú vị.

* 1. Hệ thống: Hệ thống bao gồm một server và nhiều client kết nối với nhau. Server sẽ quản lý các trận đấu và lưu trữ thông tin người chơi.
  2. Đăng nhập: Người chơi cần đăng nhập vào tài khoản của mình từ một máy client. Sau khi đăng nhập thành công, ứng dụng điều hướng đến trang chủ với các chức năng chơi game, tạo phòng chơi, vào phòng, xem danh sách người chơi, bảng xếp hạng, lịch sử đấu, hiển thị thông tin cá nhân, chat toàn server.
  3. Bắt đầu trận đấu: Người chơi có thể bắt đầu trận đấu bằng cách ấn chơi nhanh hoặc tạo phòng.
  4. Chơi nhanh: Server sẽ tìm kiếm một người chơi khác cũng nhấn chơi nhanh trong thời điểm đó để ghép trận.
  5. Tạo phòng: Server tạo phòng với một mã phòng, người chơi có thể gửi mã phòng cho người chơi khác để vào phòng chơi.
  6. Luật chơi:
* Server đưa ra 18 quân bài úp được lấy ngẫu nhiên trong 36 quân bài từ A đến 9.
* Mỗi người chơi khi đến lượt sẽ chọn ra 3 quân bài trong số các quân bài còn lại trên bàn. Sau khi chọn xong, bài của người đó sẽ được lật lên.
* Khi chọn xong bài, server sẽ tính điểm mỗi người chơi. Điểm được tính bằng tổng điểm của 3 quân bài đã chọn chia dư cho 10.
* Người có điểm số cuối cùng cao hơn sẽ là người chiến thắng. Trong trường hợp 2 người bằng điểm nhau thì tính là hoà.
* Người thắng nhận được 2 điểm, hoà nhận 1 điểm và thua không có điểm.
  1. Giao diện: Giao diện trò chơi gồm các quân bài úp chưa được chọn và nút “Chọn”. Sau khi người chơi chọn xong, màn hình hiển thị các quân bài lật.
  2. Kết thúc trận đấu: người chơi sẽ được chuyển về trang chủ.
  3. Thoát khỏi trận đấu: Người chơi có thể thoát khỏi trận đấu bất cứ lúc nào, tuy nhiên sẽ bị tính là thua.
  4. Xếp hạng: Kết quả của các trận đấu sẽ được lưu trên server. Mỗi người có thể xem bảng xếp hạng của toàn hệ thống (tổng điểm tất cả trận đấu) của mỗi người và lịch sử đấu.

1. **Phân tích yêu cầu ứng dụng**
   1. **Yêu cầu chức năng phía client**

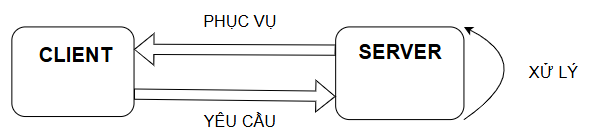
* Kết nối với server.
* Đăng ký: người dùng nhập tên đăng nhập, mật khẩu và tên hiển thị để đăng ký tài khoản.
* Đăng nhập: người dùng nhập tên đăng nhập và mật khẩu.
* Đăng xuất: người dùng đăng xuất trở lại giao diện đăng nhập.
* Thông tin cá nhân: hiển thị thông tin cá nhân gồm avatar, tên, số trận đã chơi, số trận thắng, điểm số, xếp hạng.
* Chức năng chơi game:
* Có thể ghép cặp chơi nhanh, mời bạn bè hoặc tạo phòng chơi.
* Hiển thị 16 quân bài úp để 2 người chơi, mỗi người chọn 3 quân bài.
* Chọn một quân bài sẽ lật quân bài đó cho cả 2 người chơi cùng biết.
* Trận đấu kết thúc khi hết thời gian 30 giây hoặc mỗi người cùng chọn đủ 3 quân bài. Nếu hết thời gian mà không chọn đủ bài thì client sẽ chọn ngẫu nhiên giúp người chơi.
* Điểm số được tính bằng cách tính tổng điểm của 3 quân bài đã chọn chia dư cho 10.
* Có thể chat trong trận đấu.
* Thông báo kết quả trận đấu (thắng, thua, hoà) và điểm số hai người chơi.
* Tính điểm: thắng được 2 điểm, hoà được 1 điểm, thua 0 điểm.
* Xem bảng xếp hạng: hiển thị bảng xếp hạng toàn server với thông tin về tên, thứ hạng, điểm số của người chơi.
* Xem thông tin người chơi: hiển thị thông tin người chơi gồm tên, avatar, số trận đã chơi, điểm số, xếp hạng…
* Kết bạn: có thể gửi yêu cầu kết bạn đến người chơi khác.
* Danh sách bạn bè: hiển thị danh sách bạn bè gồm thông tin về tên và trạng thái (online, offline, playing).
* Xem lịch sử đấu: có thể xem lịch sử đấu của bản thân gồm thời gian, kết quả, tỉ số, tên đối thủ.
* Ngắt kết nối client.
  1. **Yêu cầu chức năng phía server**
* Lắng nghe và chấp nhận kết nối từ client.
* Xử lý đăng ký
* Thông báo tài khoản đã tồn tại nếu nhập tên đăng nhập trùng với tên tài khoản dã tồn tại.
* Cập nhật trạng thái đang online tại cơ sở dữ liệu nếu đăng ký thành công
* Xử lý đăng nhập
* Thông báo sai tài khoản, mật khẩu nếu nhập sai
* Thông báo tài khoản đang được đăng nhập ở thiết bị khác nếu tài khoản đó được đăng nhập ở thiết bị khác.
* Cập nhật trạng thái đang online tại cơ sở dữ liệu nếu đăng nhập thành công
* Xử lý đăng xuất: cập nhật trạng thái offline tại cơ sở dữ liệu
* Xử lý trò chuyện: nhận tin nhắn từ người chơi này và gửi tới người chơi khác.
* Xử lý ghép cặp chơi nhanh: tìm kiếm người chơi khác đang nhấn chơi nhanh để ghép cặp chơi game.
* Xử lý tạo phòng: tạo phòng có mật khẩu hoặc không có mật khẩu.
* Xử lý vào phòng
* Thông báo sai mật khẩu hoặc phòng không tồn tại nếu nhập sai
* Chuyển đến game chính nếu nhập đúng phòng
* Xử lý tìm phòng
* Yêu cầu nhập mật khẩu nếu phòng có yêu cầu mật khẩu.
* Chuyển đến game chính nếu phòng không yêu cầu mật khẩu.
* Xử lý kết bạn: gửi thông báo lời mời kết bạn đến người chơi khác
* Xử lý xem danh sách bạn bè: gửi danh sách bạn bè tới client
* Xử lý logic game:
* Cập nhật trạng thái đang chơi game tại cơ sở dữ liệu.
* Trò chuyện giữa 2 người chơi trong trận đấu
* Hiển thị đồng bộ về các quân bài, có thể cho đối thủ biết quân bài mình đã chọn.
* Lưu lịch sử đấu: lưu lịch sử đấu vào cơ sở dữ liệu
* Cập nhật điểm số: cập nhật điểm số, thắng thua cho người chơi ở cơ sở dữ liệu.
* Xử lý xem bảng xếp hạng: xử lý dữ liệu thông tin người chơi từ cơ sở dữ liệu theo bảng xếp hạng rồi gửi tới client.
* Xử lý xem lịch sử đấu: gửi dữ liệu về lịch sử đấu từ cơ sở dữ liệu tới client.
* Xử lý thoát game: cập nhật trạng thái offlin tại cơ sở dữ liệu.
  1. **Yêu cầu phi chức năng**
* Hiệu năng: hệ thống hoạt động mượt mà trơn tru cho khoảng 5 client hoạt động cùng lúc.
* Thời gian thực: hệ thống phải hoạt động trong thời gian thực để đảm bảo người chơi không bị trễ trong việc gửi và xử lý chọn bài.
* Bảo mật truy cập: người chơi không được chỉnh sửa các thông tin về kết quả chơi game, bảng xếp hạng, lịch sử đấu.
* Tính khả dụng: giao diện người dùng thân thiện và dễ sử dụng.
* Chính xác thông tin: dữ liệu về cuộc trò chuyện và chơi game cần chính xác.
* Hiển thị đồng bộ: hai người chơi đều thấy được bài mà đối phương đã chọn ngay sau khi đối phương chọn một quân bài.

1. **Bảng phân công chức năng**

Bảng 1 Bảng phân công chức năng cá nhân

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ tên** | **Nội dung cá nhân** |
| 1 | Đàm Minh Anh | Thiết kế CSDL  Giao diện đăng nhập, đăng ký, trang chủ, kết bạn  Code chức năng đăng nhập, đăng xuất, đăng ký, thoát game, hiển thị thông tin cá nhân, kết bạn |
| 2 | Lưu Trung Kiên | Giao diện tạo phòng, tìm phòng, vào phòng, xem danh sách bạn bè, mời bạn chơi game.  Code chức năng tạo phòng, tìm phòng, vào phòng, xem danh sách bạn bè, mời bạn chơi game. |
| 3 | Hà Cường Thịnh | Giao diện chơi nhanh, game chính, kết quả game  Code chức năng chơi nhanh, game chính, kết quả game |
| **4** | **An Quốc Việt** | **Kết nối server và client**  **Giao diện bảng xếp hạng, thông tin đối thủ, lịch sử đấu, trang chủ**  **Code chức năng xem bảng xếp hạng, thông tin đối thủ, lích sử đấu, đổi avatar, chat** |

1. **PHÂN TÍCH THIẾT KẾ TỔNG QUAN ỨNG DỤNG**
2. **Kiến trúc tổng quan**



Hình 1 Kiến trúc tổng quan

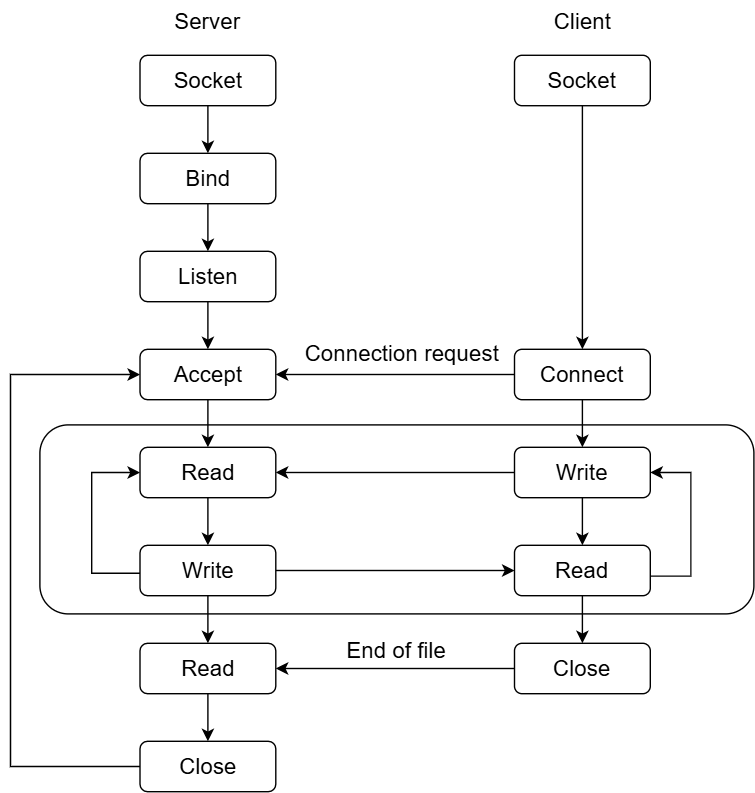
Client:

* Tạo ra 1 yêu cầu
* Gửi yêu cầu qua Server
* Chờ Server xử lý
* Nhận kết quả trả về, và xử lý theo mục đích riêng

Server

* Lắng nghe yêu cầu
* Nhận yêu cầu
* Xử lý yêu cầu
* Gửi kết quả đã xử lý về cho Client

1. **Sơ đồ khối chức năng của Server và Client**



Hình 2 Sơ đồ khối các chức năng

Chức năng của Server:

* Mở Socket
* Lắng nghe yêu cầu từ phía Client
* Chấp nhận kết nối từ Client
* Nhận yêu cầu từ Client và xử lý
* Phản hồi lại cho Client

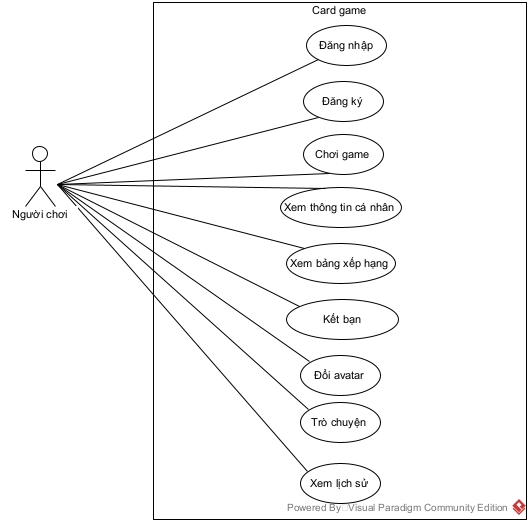
Chức năng của Client:

* Mở Socket
* Kết nối đến Server
* Gửi yêu cầu đến Server
* Nhận phản hồi từ phía Server

1. **Sơ đồ Use case tổng quan**

Mô tả Use case:

* Đăng nhập: cho phép người dùng đăng nhập vào tài khoản.
* Đăng ký: cho phép người dùng đăng ký tài khoản mới.
* Chơi game: cho phép người dùng ghép cặp hoặc thách đấu để chơi game.
* Xem thông tin cá nhân: cho phép người dùng được xem thông tin cá nhân bao gồm tên, số trận đã chơi, số trận thắng, số trận hoà, tỉ lệ thắng, điểm số, thứ hạng.
* Xem bảng xếp hạng: cho phép người dùng xem thông tin người chơi toàn server.
* Xem lịch sử đấu: cho phép người dùng xem lịch sử đấu của bản thân.
* Kết bạn: cho phép người dùng kết bạn với người khác và xem danh sách bạn bè.
* Đổi avatar: cho phép người dùng thay đổi ảnh đại diện của mình và hiển thị ảnh.
* Trò chuyện: cho phép người dùng trò chuyện với đối thủ trong khi chơi hoặc trò chuyện với người dùng toàn server.

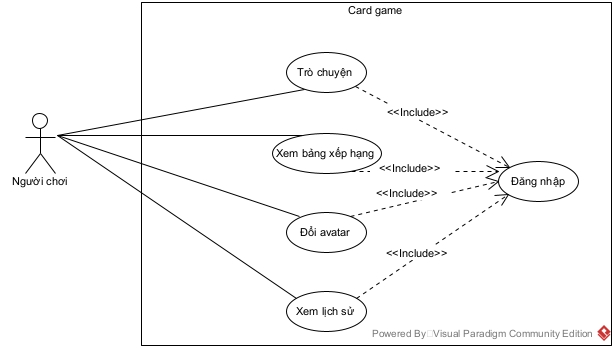


Hình 3 Sơ đồ Use case tổng quan

1. **PHÂN TÍCH THIẾT KẾ CHI TIẾT CHỨC NĂNG CÁ NHÂN**
2. **Sơ đồ Use case chi tiết**

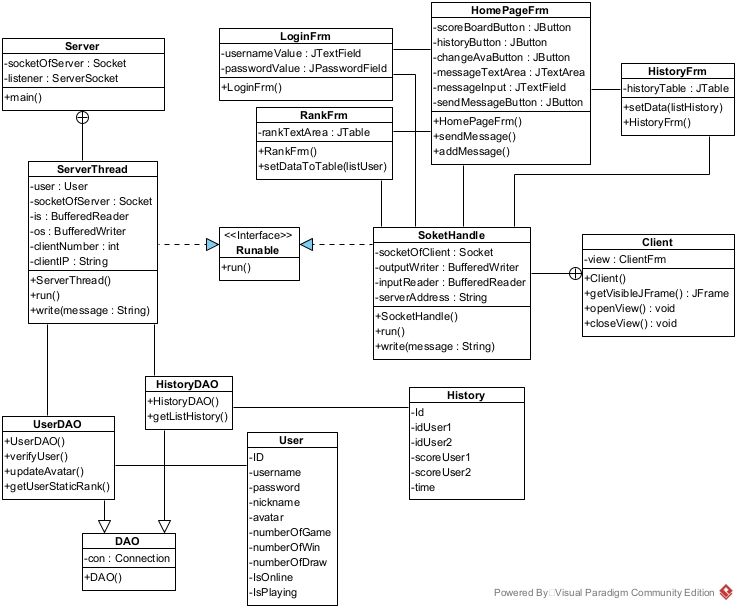
Đặc tả Use case:

* Xem bảng xếp hạng: cho phép người dùng xem thông tin người chơi toàn server.
* Xem lịch sử đấu: cho phép người dùng xem lịch sử đấu của bản thân.
* Đổi avatar: cho phép người dùng thay đổi ảnh đại diện của mình và hiển thị ảnh.
* Trò chuyện: cho phép người dùng trò chuyện với đối thủ trong khi chơi hoặc trò chuyện với người dùng toàn server.



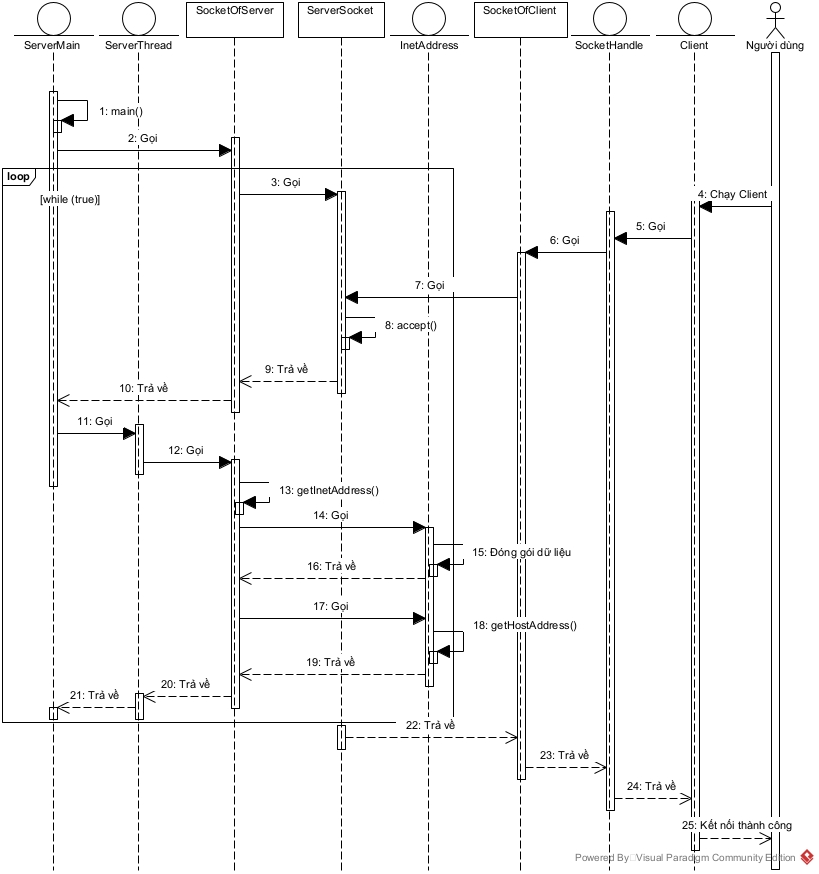
Hình 4 Sơ đồ Use case chi tiết

1. **Biểu đồ lớp**



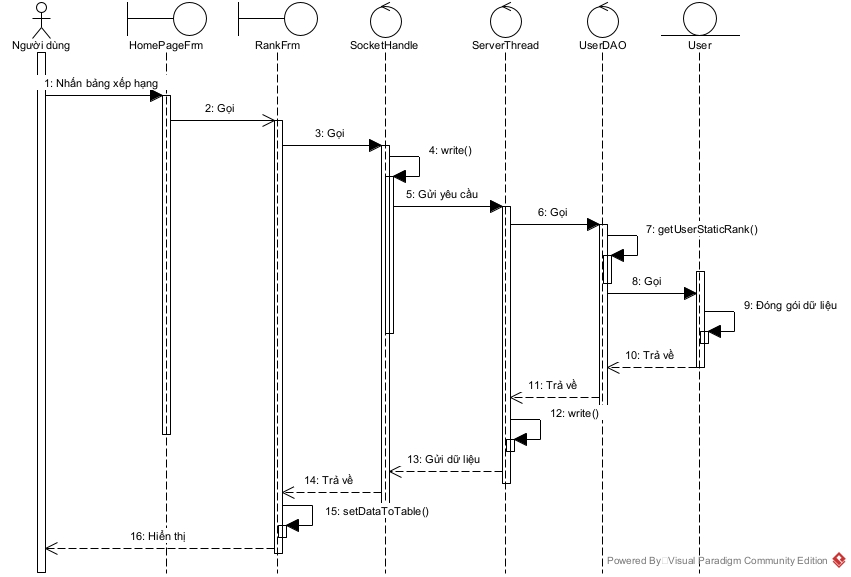
Hình 5 Biểu đồ lớp

1. **Biểu đồ tuần tự**
   1. **Biểu đồ tuần tự kết nối Client với Server**



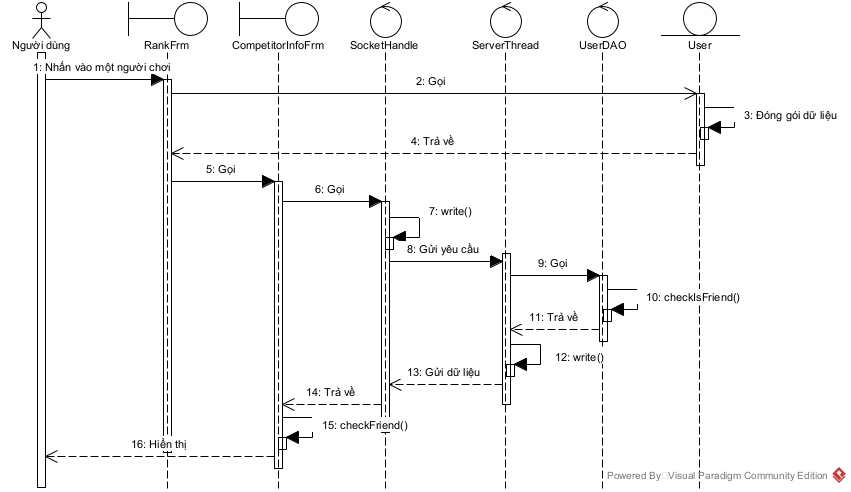
Hình 6 Biểu đồ tuần tự kết nối Client với Server

1. Chạy Server bằng cách gọi hàm main().
2. ServerMain gọi tới Socket của Server.
3. Socket của Server gọi tới ServerSocket.
4. Người dùng chạy Client.
5. Client gọi tới SocketHandle.
6. SocketHandle gọi tới Socket của Client.
7. Socket của Client gửi yêu cầu kết nối tới ServerSocket.
8. ServerSocket gọi phương thức accept().
9. Phương thức accept() trả vể Socket của Server.
10. Socket của Server trả về Server.
11. Server gọi đến ServerThread.
12. ServerThread gọi đến Socket của Server.
13. Socket của Server gọi phương thức getInetAddress().
14. Phương thức getInetAddress() gọi đến lớp InetAddress.
15. InetAddress đóng gói dữ liệu.
16. InetAddress trả về Socket của Server.
17. Socket của Server gọi tới InetAddress.
18. InetAddress gọi phương thức getHostAddress().
19. Phương thức getHostAddress() trả về Socket của Server địa chỉ IP của Client.
20. Socket của Server trả về ServerThread địa chỉ IP của Client.
21. ServerThread trả về lời gọi từ ServerMain.
22. ServerSocket trả về SocketOfClient yêu cầu kết nối.
23. Socket của Client trả về SocketHandle
24. SocketHandle trả về Client.
25. Client thông báo tới người dùng đã kết nối thành công tới Server.
    1. **Biểu đồ tuần tự xem bảng xếp hạng**



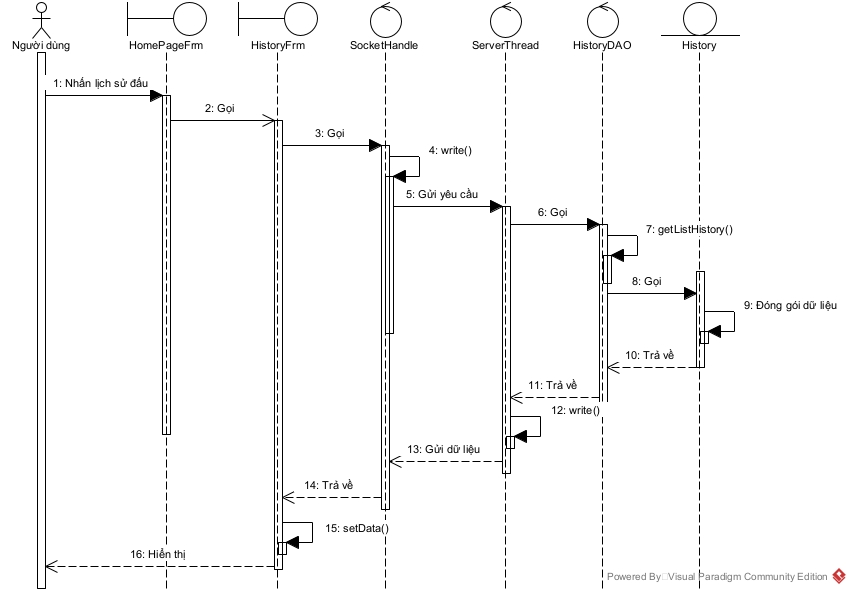
Hình 7 Biểu đồ tuần tự cho chức năng xem bảng xếp hạng

1. Người dùng nhấn nút bảng xếp hạng tại giao diện HomePageFrm.
2. Giao diện HomePageFrm gọi đến giao diện RankFrm.
3. Giao diện RankFrm gọi đến SocketHandle.
4. SocketHandle gọi phương thức write() để gửi dữ liệu.
5. Phương thức write() gửi yêu cầu tới server.
6. ServerThread nhận yêu cầu và gọi tới lớp UserDAO.
7. UserDAO gọi phương thức getUserStaticRank().
8. UserDAO gọi tới lớp User.
9. User đóng gói các thuộc tính của nó.
10. User trả về UserDAO danh sách các đối tượng User.
11. Phương thức getUserStaticRank() trả về ServerThread danh sách các User đã được sắp xếp theo thứ hạng.
12. ServerThread gọi phương thức write().
13. Phương thức write() gửi dữ liệu về phía SocketHandle
14. SocketHanldle gửi dữ liệu về RankFrm.
15. RankFrm gọi phương thức setDataToTable() để gửi dữ liệu ra bảng.
16. Giao diện RankFrm với bảng xếp hạng hiển thị tới người dùng.
    1. **Biểu đồ tuần tự xem thông tin người chơi khác**

****

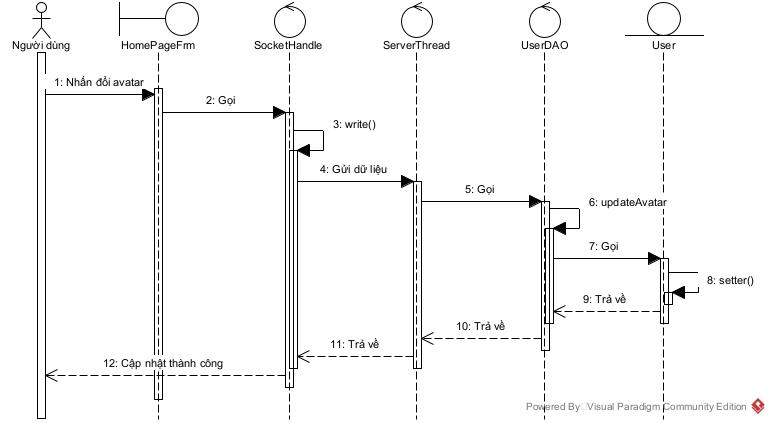
Hình 8 Biểu đồ tuần tự xem thông tin người chơi

1. Người dùng nhấn vào một người chơi tại giao diện RankFrm.
2. Giao diện RankFrm gọi tới lớp User.
3. Lớp User đóng gói dữ liệu
4. Lớp User trả về giao diện RankFrm.
5. Giao diện RankFrm gọi tới giao diện CompetitorInfoFrm
6. Giao diện CompetitorInfoFrm gọi tới SocketHandle.
7. SocketHandle gọi phương thức write().
8. Phương thức write() gửi yêu cầu kiểm tra bạn bè tới ServerThread.
9. ServerThread gọi tới UserDAO.
10. UserDAO gọi phương thức checkIsFriend().
11. Phương thức checkIsFriend() trả về UserDAO.
12. UserDAO trả kết quả về SocketHandle.
13. SocketHandle gọi phương thức write().
14. Phương thức write() gửi dữ liệu về kết quả kiểm tra bạn bè cho SocketHandle.
15. SocketHandle trả về CompetitorFrm.
16. CompetitorFrm gọi phương thức checkFriend().
17. Competitor hiển thị thông tin đối thủ và cho biết người đó có phải bạn bè không tới người chơi.
    1. **Biểu đồ tuần tự xem lịch sử đấu**



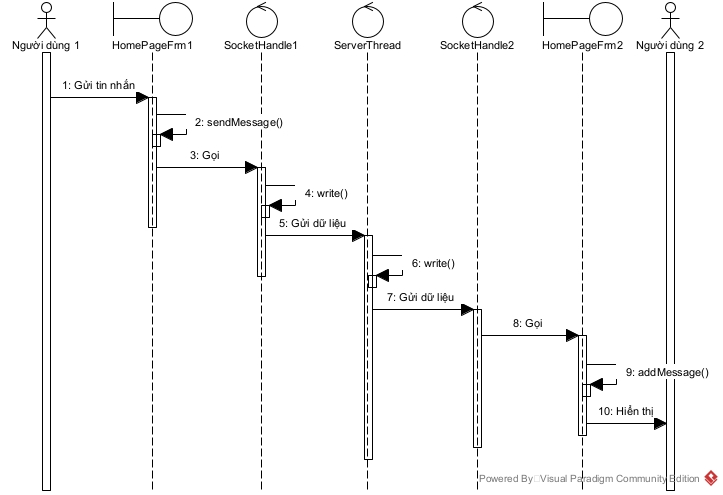
Hình 9 Biểu đồ tuần tự chức năng xem lịch sử đấu

1. Người dùng nhấn nút bảng xếp hạng tại giao diện HomePageFrm.
2. Giao diện HomePageFrm gọi đến giao diện HistoryFrm.
3. Giao diện HistoryFrm gọi đến SocketHandle.
4. SocketHandle gọi phương thức write() để gửi dữ liệu.
5. Phương thức write() gửi yêu cầu tới server.
6. ServerThread nhận yêu cầu và gọi tới lớp HistoryDAO.
7. HistoryDAO gọi phương thức getListHistory().
8. HistoryDAO gọi tới lớp History.
9. History đóng gói các thuộc tính của nó.
10. History trả về HistoryDAO danh sách các đối tượng History.
11. Phương thức getListHistory() trả về ServerThread danh sách các History.
12. ServerThread gọi phương thức write().
13. Phương thức write() gửi dữ liệu về phía SocketHandle
14. SocketHanldle gửi dữ liệu về HistoryFrm.
15. HistoryFrm gọi phương thức setData() để gửi dữ liệu ra bảng.
16. Giao diện HistoryFrm với lịch sử đấu hiển thị tới người dùng.
    1. **Biểu đồ tuần tự đổi avatar**



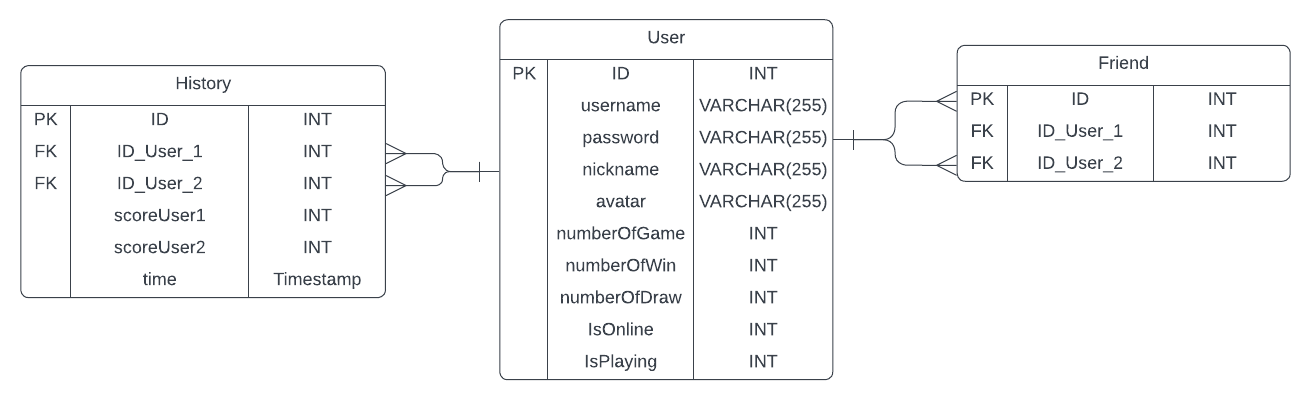
Hình 10 Biểu đồ tuần tự chức năng đổi avatar

1. Người dùng nhấn đối avatar và chọn ảnh tại giao diện HomePageFrm.
2. Giao diện HomePageFrm gọi đến SocketHandle.
3. SocketHandle gọi phương thức write().
4. Phương thức write() gửi dữ liệu tới ServerThread.
5. ServerThread gọi lớp UserDAO.
6. UserDAO gọi phương thức updateAvatar().
7. UserDAO gọi tới lớp User.
8. User gọi phương thức setter().
9. User trả về kết quả cho phương thức updateAvatar().
10. Phương thức updateAvartar() trả về kết quả cho ServerThread.
11. ServerThread trả kết quả về SocketHandle.
12. SocketHandle hiển thị thông báo thành công tới người dùng qua log.
    1. **Biểu đồ tuần tự trò chuyện**



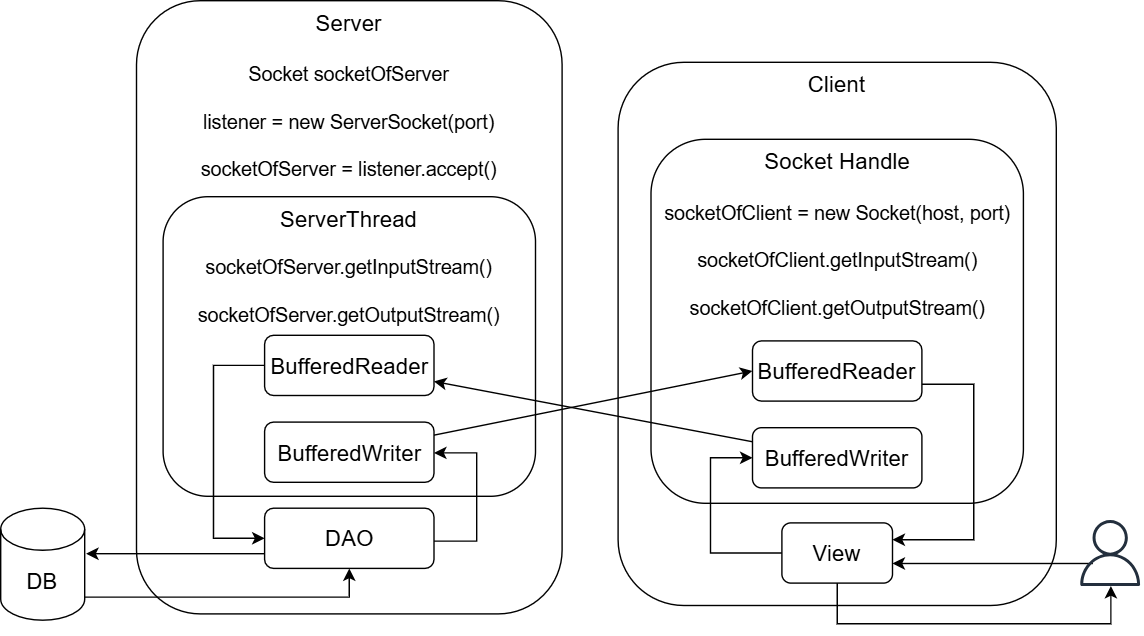
Hình 11 Biểu đồ tuần tự chức năng trò chuyện

1. Người dùng 1 nhập tin nhắn và nhấn nút gửi tại giao diện HomePageFrm1 của người đó.
2. Giao diện HomePageFrm1 gọi phương thức sendMessage().
3. Phương thức sendMessage() gọi tới SocketHandle1.
4. SocketHandle1 gọi phương thức write().
5. Phương thức write() của SocketHandle1 gửi dữ liệu về tin nhắn tới ServerThread.
6. ServerThread gọi phương thức write().
7. Phương thức write() của ServerThread gửi dữ liệu tới SocketHandle2 của người dùng 2.
8. SocketHandle2 của người dùng 2 gọi tới giao diện HomePageFrm2.
9. Giao diện HomePageFrm2 gọi phương thức addMessage() để thêm đoạn tin nhắn vào hộp chat.
10. Giao diện HomePageFrm2 hiển thị tin nhắn của người dùng 1 tới người dùng 2.
11. **Biểu đồ quan hệ**



Hình 12 Biểu đồ quan hệ

1. **KẾT QUẢ ỨNG DỤNG**
2. **Kiến trúc ứng dụng**



Hình 13 Kiến trúc ứng dụng

Mô tả kiến trúc ứng dụng:

Client:

* Client xử lý dữ liệu thông qua SocketHandle.
* SocketHandle kết nối tới Server thông qua một đối tượng Socket.
* Socket lấy luồng đầu vào và luồng đầu ra.
* BufferedReader và BufferedWriter trao đổi dữ liệu giữa Client và Server.
* Người dùng tương tác với ứng dụng thông qua view.
* View tương tác BufferedWriter để gửi dữ liệu tới Server
* BufferedReader nhận dữ liệu từ Server để gửi tới view.

Server:

* Server tạo một đối tượng Socket để lắng nghe yêu cầu từ phía Client.
* Server chấp nhận yêu cầu kết nối từ Client.
* Server xử lý dữ liệu thông qua ServerThread.
* Socket lấy luồng đầu vào và luồng đầu ra.
* BufferedReader và BufferedWriter trao đổi dữ liệu giữa Client và Server.
* BufferedReader nhận dữ liệu từ Client để Server xử lý và lưu vào cơ sở dữ liệu.
* Server lấy dữ liệu từ cơ sở dữ liệu để BufferedWriter gửi tới Client.

1. **Cài đặt và triển khai**
   1. **Cài đặt cơ sở dữ liệu**

Chạy lệnh SQL trong MySQL Workbench hoặc XAMPP

CREATE DATABASE btlviet;

USE btlviet;

CREATE TABLE `user` (

`ID` int NOT NULL AUTO\_INCREMENT,

`username` varchar(255) DEFAULT NULL,

`password` varchar(255) DEFAULT NULL,

`nickname` varchar(255) DEFAULT NULL,

`avatar` varchar(255) DEFAULT NULL,

`numberOfGame` int DEFAULT '0',

`numberOfWin` int DEFAULT '0',

`numberOfDraw` int DEFAULT '0',

`IsOnline` int DEFAULT '0',

`IsPlaying` int DEFAULT '0',

PRIMARY KEY (`ID`),

UNIQUE KEY `username` (`username`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

CREATE TABLE `history` (

`id` int NOT NULL AUTO\_INCREMENT,

`idUser1` int NOT NULL,

`idUser2` int NOT NULL,

`scoreUser1` int NOT NULL,

`scoreUser2` int NOT NULL,

`time` timestamp NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=47 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

CREATE TABLE `friend` (

`ID\_User1` int NOT NULL,

`ID\_User2` int NOT NULL,

PRIMARY KEY (`ID\_User1`,`ID\_User2`),

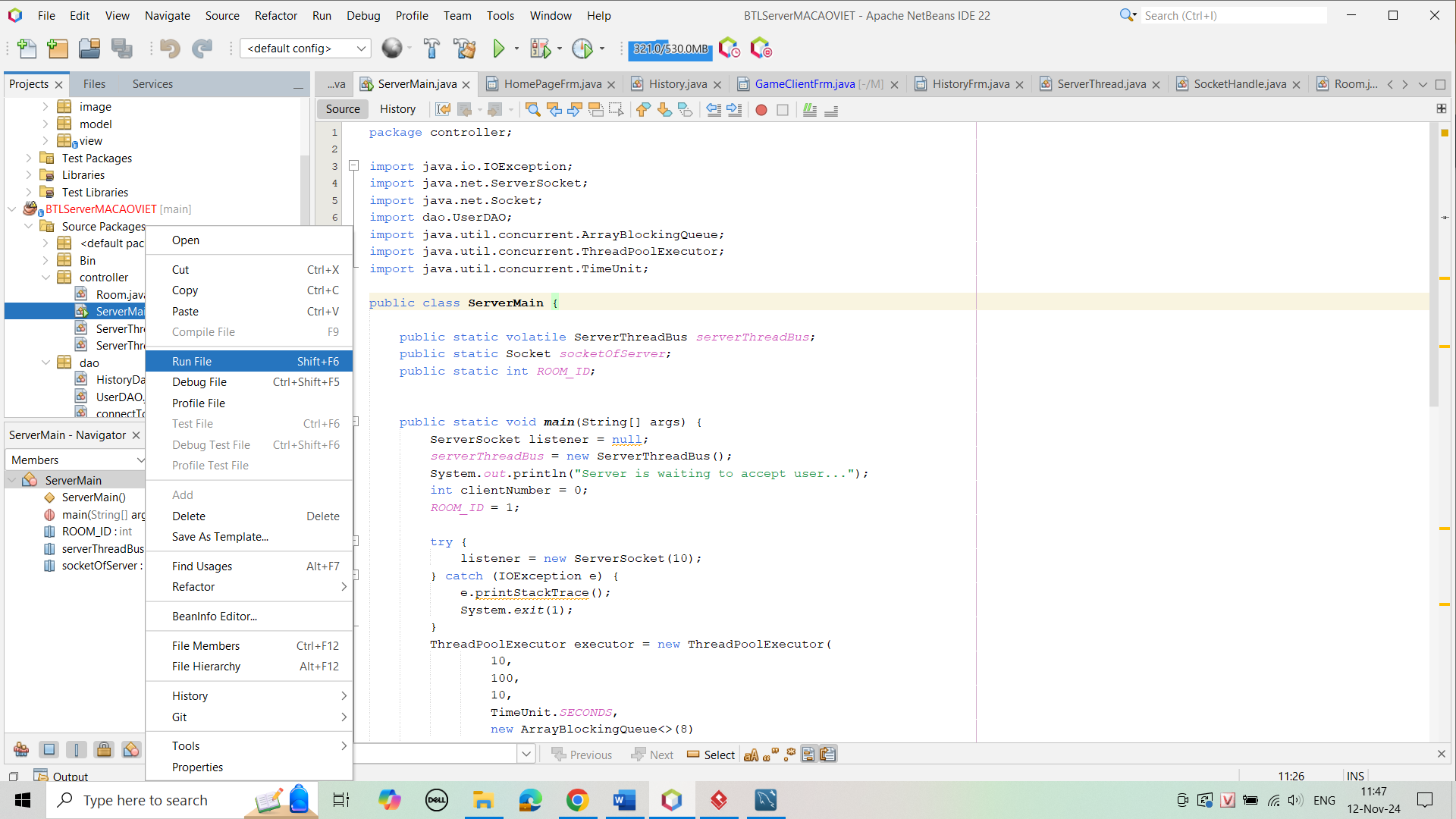
KEY `ID\_User2` (`ID\_User2`),

CONSTRAINT `friend\_ibfk\_1` FOREIGN KEY (`ID\_User1`) REFERENCES `user` (`ID`),

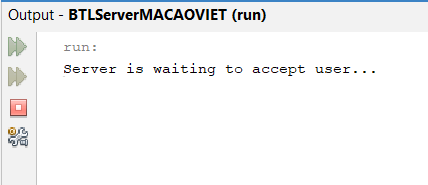
CONSTRAINT `friend\_ibfk\_2` FOREIGN KEY (`ID\_User2`) REFERENCES `user` (`ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

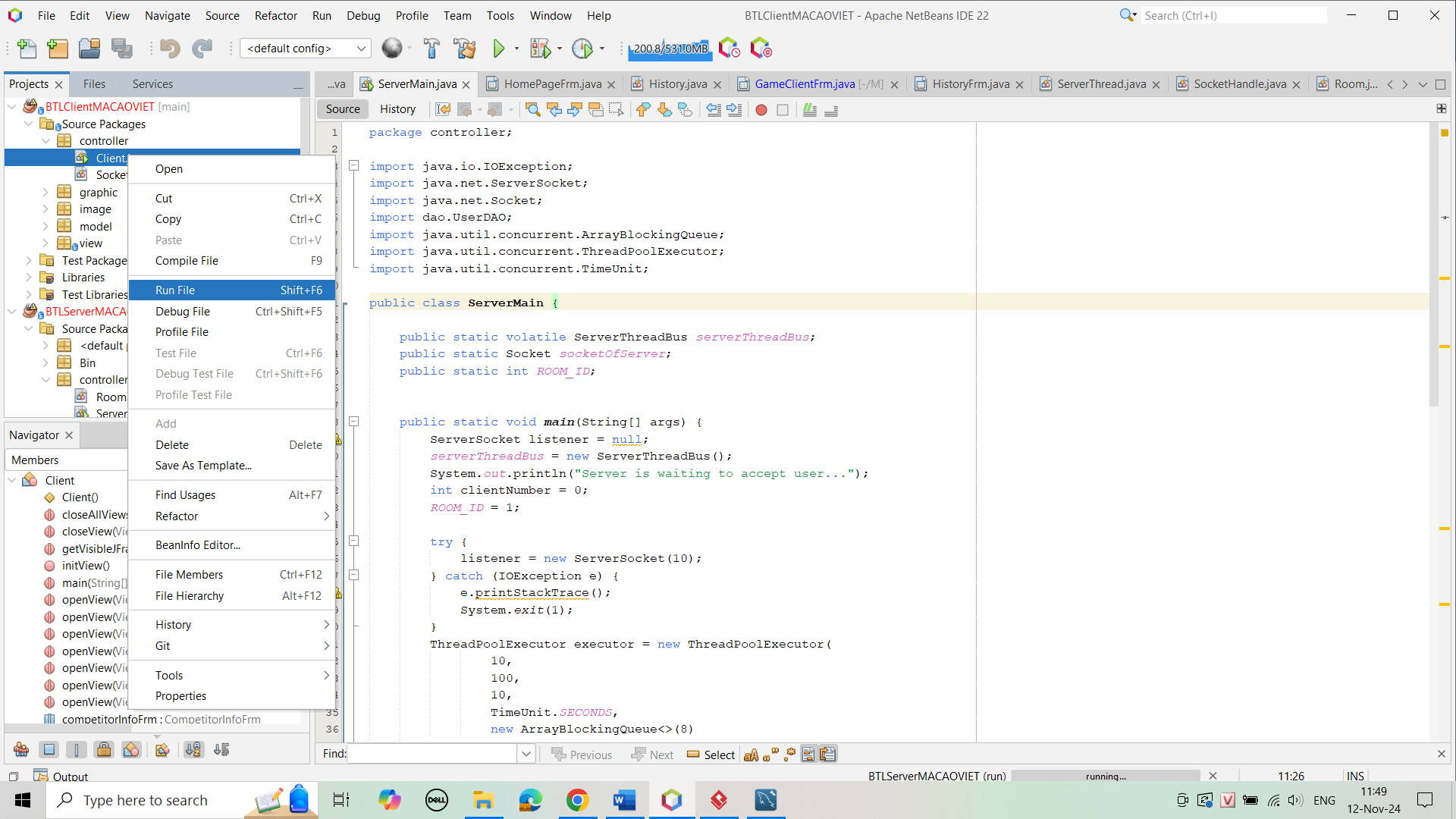
* 1. **Thêm các thư viện cần thiết**
* apache-mime4j-0.6.jar
* commons-logging-1.3.4.jar
* cloudinary-core-1.30.0.jar
* cloudinary-http44-1.30.0.jar
* flatlaf-3.5.2.jar
* httpclient-4.5.13.jar
* httpcore-4.4.13.jar
* httpmime-4.5.13.jar
* mysql-connector-j-9.0.0.jar
  1. **Khởi chạy ứng dụng**
     1. **Chạy file ServerMain.java**



* + 1. **Kết quả chạy thành công:**



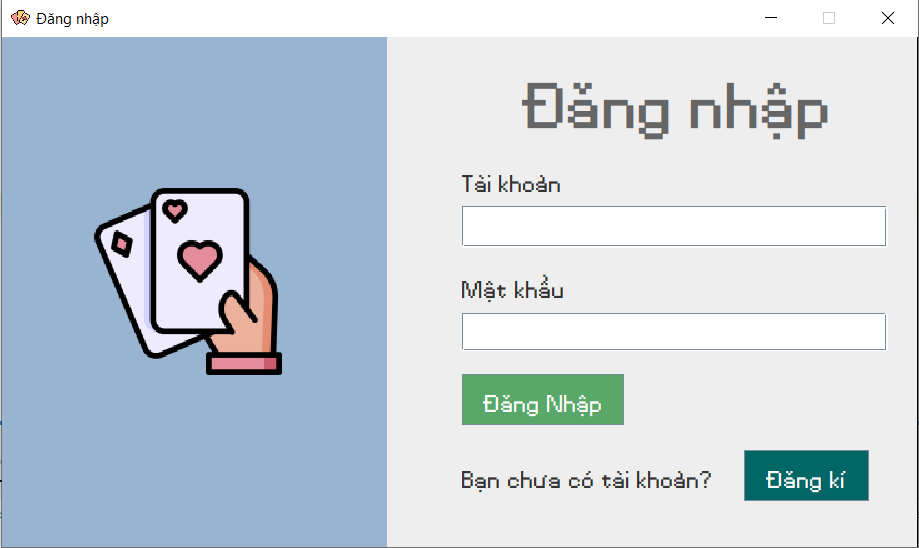
* + 1. **Chạy file Client.java**



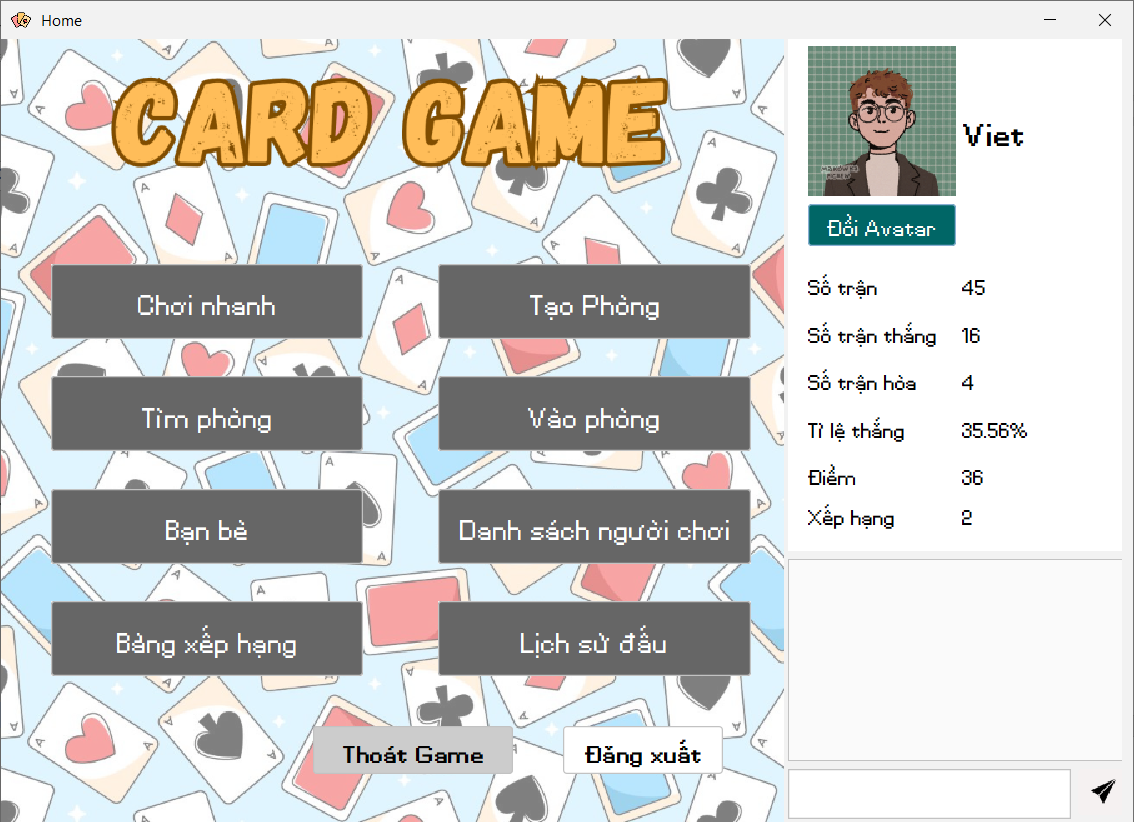
* + 1. **Kết quả hiển thị chạy Client thành công**



Đồng thời giao diện đăng nhập được mở lên



1. **Kết quả thực hiện**
   1. **Giao diện Trang chủ**



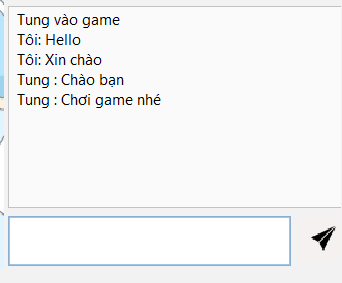
* 1. **Chức năng đổi avatar**

Người dùng nhấn nút Đổi avatar và chọn một ảnh từ máy tính

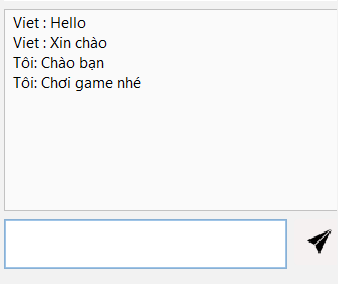
Kết quả sau khi đổi avatar:



* 1. **Chức năng trò chuyện**
* Ô trò chuyện tại giao diện trang chủ của người chơi thứ nhất hiển thị các tin nhắn gồm có tên người gửi và nội dung



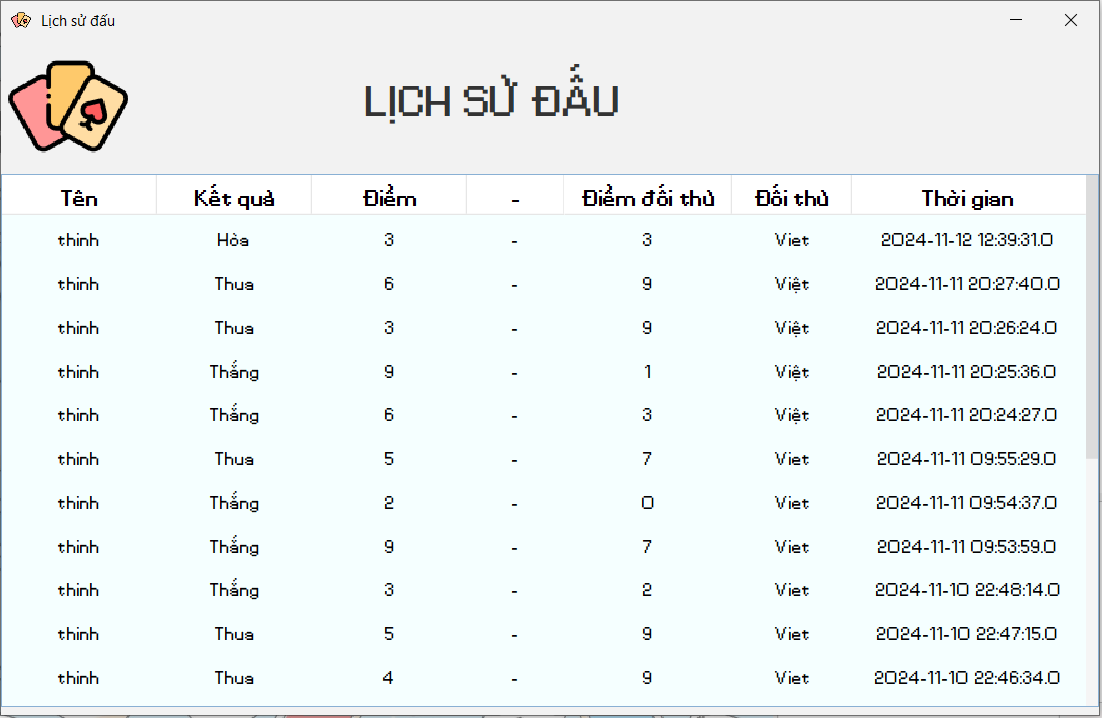
* Ô trò chuyện tại giao diện trang chủ của người chơi thứ hai hiển thị các tin nhắn



* 1. **Chức năng xem lịch sử đấu**

Người dùng nhấn nút Lịch sử đấu tại giao diện trang chủ.

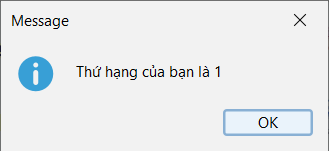
Giao diện Lịch sử đấu sẽ hiển thị:



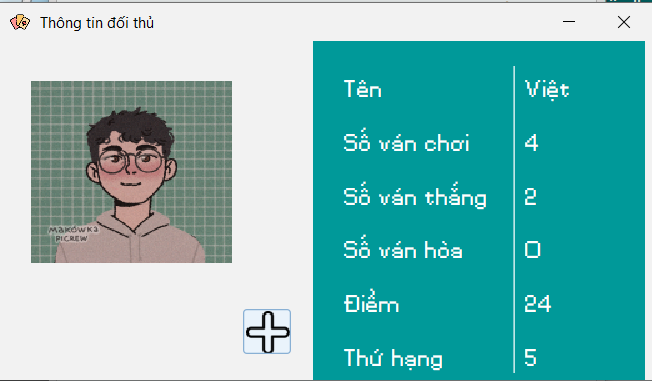
* 1. **Chức năng xem bảng xếp hạng và xem thông tin người chơi**
* Nhấn nút bảng xếp hạng tại giao diện trang chủ, giao diện bảng xếp hạng hiển thị:



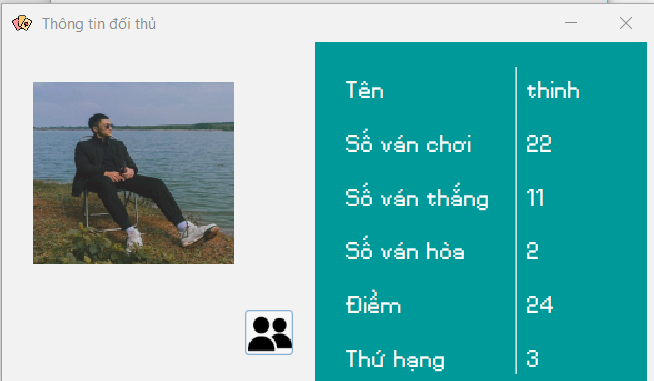
* Nếu nhấn vào chính mình tại giao diện bảng xếp hạng, một thông báo hiện ra



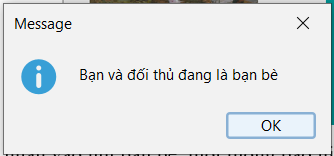
* Nếu nhấn vào một người chơi khác tại giao diện bảng xếp hạng, giao diện thông tin đối thủ hiển thị. Người dùng có thể nhấn nút thêm bạn để gửi lời mời kết bạn.



* Giao diện thông tin đối thủ khi click vào một người chơi không phải bạn bè của mình tại bảng xếp hạng



Nếu nhấn vào nút bạn bè, một thông báo hiện ra



1. **Tài liệu tham khảo**

* GPCoder, "Lập trình đa luồng trong Java (Java Multi-threading)"
* ShareProgramming, "Java Socket Archives"
* BroCode,"JavaSwing Full Course” [Online]
* TITV,"Khóa học lập trình mạng với java” [Online]

1. **Source code**

Quét mã QR tại đây

