

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGOẠI NGỮ-TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO KẾT THÚC HỌC PHẦN
Lập Trình Mạng

ĐỀ TÀI:
XÂY DỰNG TRÒ CHƠI CARO

Giảng viên hướng dẫn: ThS. Phạm Phú Thiện

Sinh viên thực hiện: Nhóm 2

- Hứa Nguyễn Tường Vy
- Phùng Ngọc Bách
- Nguyễn Ngọc Anh Thư
- Nguyễn Thị Kim Doanh

TP HCM, ngày 27, tháng 07, năm 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGOẠI NGỮ TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
MÔN: Lập Trình Mạng
ĐỀ TÀI:
XÂY DỰNG TRÒ CHƠI CARO

Giảng viên hướng dẫn : Phạm Phú Thiện

Sinh viên thực hiện : Hứa Nguyễn Tường Vy – 22DH114348

Phùng Ngọc Bách – 22DH114463

Nguyễn Ngọc Anh Thư – 22DH113640

Nguyễn Thị Kim Doanh – 22DH110511

TP HCM, ngày 27, tháng 07, năm 2024

MỤC LỤC

DANH MỤC HÌNH ẢNH	6
BẢNG PHÂN CÔNG CÔNG VIỆC	7
LỜI CẢM ƠN.....	8
Chương I: GIỚI THIỆU ĐỀ TÀI.....	9
1.1 Giới Thiệu.....	9
1.1.1 Mở đầu.....	9
1.1.2 Lý do.....	9
1.1.3 Mục đích của trò chơi.....	10
1.1.4 Bàn cờ và quân cờ.....	10
1.1.5 Luật chơi.....	10
1.1.6 Chiến lược.....	10
1.1.7 Biến thể.....	10
1.1.8 Lịch sử và phổ biến.....	11
1.1.9 Lợi ích của trò chơi.....	11
1.2 Mục tiêu nghiên cứu	11
1.2.1 Phát triển trí tuệ nhân tạo.....	11
1.2.2 Nghiên cứu chiến lược tối ưu hoá.....	11
1.2.3 Phát triển giáo dục và đào tạo	11
1.2.4 Nghiên cứu khoa học và toán học.....	11
1.2.5 Phát triển và Cải thiện trò chơi	12
1.2.6 Xã hội và văn hoá.....	12
1.3 Phạm vi nghiên cứu	12
1.3.1 Khoa học máy tính và trí tuệ nhân tạo.....	12
1.3.2 Toán học và lý thuyết trò chơi	13
1.3.3 Giáo dục và đào tạo	13
1.3.4 Văn hoá và xã hội.....	13
1.3.5 Phát triển và cải thiện trò chơi	13
1.3.6 Nghiên cứu tâm lý học	14
CHƯƠNG II: PHÂN TÍCH YÊU CẦU.....	14
2.1 Yêu cầu chức năng.....	14
2.1.1 Giao diện người dùng	14
2.1.2 Quản lý trận đấu	15
2.1.3 Xử lý trận đấu	15
2.1.4 Tùy chọn và Cài đặt.....	15

2.1.5 Trí tuệ nhân tạo.....	16
2.1.6 Lịch sử và thống kê.....	16
2.1.7 Hỗ trợ trực tuyến	16
2.1.8 Trợ giúp và hướng dẫn.....	16
2.1.9 Bảo mật và quản lý tài khoản	16
2.1.10 Khả năng tùy biến và mở rộng	17
2.2 Yêu cầu phi chức năng	17
2.2.1 Hiệu năng (Performance).....	17
2.2.2 Bảo mật (Security).....	17
2.2.3 Độ tin cậy (Reliability).....	18
2.3 Các kịch bản sử dụng	18
2.3.1 Đăng ký tài khoản	18
2.3.2 Đăng nhập.....	18
2.3.3 Bắt đầu ván mới.....	19
2.3.4 Chơi ván cờ.....	19
2.3.5 Đăng xuất.....	19
CHƯƠNG III : THIẾT KẾ HỆ THỐNG	21
3.1 Kiến trúc hệ thống	21
3.2 Thiết kế chi tiết.....	21
3.2.1 Sơ đồ chức năng	21
3.2.2 Mô hình tổng quan.....	21
3.2.3 Thiết kế cơ sở dữ liệu	22
3.3 Thiết kế các module và các luồng dữ liệu	23
3.3.1 Sơ đồ đăng ký	23
3.3.2 Sơ đồ đăng nhập.....	24
3.3.3 Sơ đồ yêu cầu ghép đấu	25
3.3.4 Sơ đồ các nước đi	26
3.3.5 Sơ đồ tạo phòng.....	27
3.3.6 Sơ đồ vào phòng	28
Chương IV: MÃ NGUỒN	29
4.1 Môi trường phát triển.....	29
4.1.1 Công cụ phát triển	29
4.1.2 Môi trường phát triển.....	29
4.2 Cấu trúc mã nguồn	30
4.2.1 Thư mục gốc	30
4.2.2 Thư mục src.....	31

4.2.3 Thư mục lib	32
4.2.4 Thư mục assets	33
4.2.5 Thư mục build	35
4.3 Quá trình cài đặt và triển khai	38
4.3.1 Yêu cầu hệ thống.....	38
4.3.2 Thiết lập môi trường phát triển.....	38
4.3.3 Thiết lập dự án	38
4.3.4 Các đoạn mã nguồn quan trọng	41
Chương V: ĐÁNH GIÁ VÀ KẾT LUẬN	50
5.1 Kết quả thực thi	50
5.1.1 Giao diện trang đăng nhập.....	50
5.1.2 Giao diện đăng ký	50
5.1.3 Giao diện chính	51
5.1.4 Giao diện bàn cờ	52
5.1.5 Giao diện kết thúc ván chơi.....	53
5.1.6 Giao diện lịch sử chơi	54
5.1.7 Giao diện thách đấu máy.....	55
5.1.8 Giao diện vào phòng	56
5.1.9 Giao diện tạo phòng.....	56
5.1.10 Giao diện danh sách bạn bè	57
5.1.11 Giao diện bảng xếp hạng và thông tin đối thủ.....	58
5.1.12 Giao diện Admin	59
5.2 Đánh giá hệ thống	60
5.2.1 Hiệu năng.....	60
5.2.2 Bảo mật	61
5.2.3 Độ tin cậy	61
5.3 Kết luận.....	61
5.3.1 Hiệu năng.....	61
5.3.2 Bảo mật	62
5.3.4 Độ tin cậy	62
5.4 Hạn chế và hướng phát triển	62
5.4.1 Hạn chế của đề tài về cờ caro.....	62
5.4.2 Hướng phát triển trong tương lai.....	63
Chương VI: TÀI LIỆU THAM KHẢO	64

DANH MỤC HÌNH ẢNH

Hình3.1. Kiến trúc hệ thống.....	21
Hình3.2 Sơ đồ chức năng.....	21
Hình3.3. Sơ đồ chức năng.....	22
Hình3.4. Thiết kế cơ sở dữ liệu.....	22
Hình3.5.Sơ đồ đăng ký.....	23
Hình3.6.Sơ đồ đăng nhập.....	24
Hình3.7. Sơ đồ yêu cầu ghép đấu.....	25
Hình3.8.Sơ đồ nước đi.....	26
Hình3.9 Sơ đồ tạo phòng.....	27
Hình3.10. Sơ đồ vào phòng.....	28
Hình4.1. Thư mục gốc.....	30
Hình4.2. Thư mục scr.....	32
Hình4.3. Thư mục lib.....	33
Hình4.4. Thư mục assets.....	35
Hình4.5. Thư mục build.....	37
Hình4.6.Cấu hình CSDL.....	39
Hình4.7.Cấu hình kết nối Database.....	41
Hình4.8.Kết nối Database.....	41
Hình4.9. Thiết lập máy chủ.....	42
Hình4.10 .Đăng nhập , đăng ký.....	43
Hình4.11.SocketHandle.....	43
Hình4.12.Thời gian cho 1 lượt đi.....	45
Hình4.13.Thời gian cho toàn trận đấu.....	46
Hình4.14.Kiểm tra hàng.....	46
Hình4.15.Kiểm tra cột.....	47
Hình4.16.Kiểm tra chéo phải.....	48
Hình4.17.Kiểm tra chéo trái.....	49
Hình5.1. Giao diện đăng nhập.....	50
Hình5.2. Giao diện đăng ký.....	51
Hình5.3. Giao diện chính.....	52
Hình5.4. Giao diện bàn cờ.....	53
Hình5.5. Giao diện kết thúc ván chơi.....	54
Hình5.6. Giao diện lịch sử chơi.....	55
Hình5.7. Giao diện đấu máy.....	56

Hình5.8. Giao diện vào phòng	56
Hình5.9. Giao diệntạo phòng.....	57
Hình5.10. Giao diện danh sách bạn bè	58
Hình5.11. Giao diện bảng xếp hạng và thông tin đối thủ	59
Hình5.12. Giao diện Admin.....	60

BẢNG PHÂN CÔNG CÔNG VIỆC

Tên thành viên	Công việc
Hứa Nguyễn Tường Vy	Chia công việc, tìm tài liệu tham khảo, làm báo cáo, các trận đấu, xem thành tích cá nhân, xếp hạng.
Nguyễn Ngọc Anh Thư	Tìm tài liệu tham khảo, làm báo cáo, đăng ký, đăng nhập.
Phùng Ngọc Bách	Tìm game tham khảo, làm báo cáo, tài khoản cá nhân, chỉnh sửa thông tin cá nhân, tạo, tìm và vào phòng.
Nguyễn Thị Kim Doan	Tìm game tham khảo, làm báo cáo , làm Powerpoint, trận đấu, danh sách bạn bè.

LỜI CẢM ƠN

Thay mặt nhóm sinh viên lớp Lập trình mạng, chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô đã tận tâm giảng dạy và hướng dẫn chúng em trong suốt thời gian qua.

Dưới sự dẫn dắt của quý thầy cô, chúng em đã có cơ hội tiếp cận và lĩnh hội những kiến thức quý báu về lập trình web. Những kiến thức này đã giúp chúng em hiểu rõ hơn về các khái niệm, nguyên lý và cách thức xây dựng một trang web.

Đặc biệt, trong quá trình thực hiện bài tập thực hành, quý thầy cô đã luôn tận tình giải đáp những thắc mắc của chúng em. Sự hỗ trợ nhiệt tình của quý thầy cô đã giúp chúng em vượt qua những khó khăn và hoàn thành bài tập một cách tốt nhất.

Dù đã cố gắng hết sức, nhưng chúng em vẫn nhận thấy rằng bài tập của mình còn có những thiếu sót. Chúng em rất mong nhận được những góp ý quý báu của quý thầy cô để chúng em có thể hoàn thiện hơn trong những bài thực hành tiếp theo.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô, bạn bè và gia đình đã giúp đỡ chúng em trong suốt quá trình học tập và nghiên cứu.

Trân trọng!

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

1.1 Giới Thiệu

1.1.1 Mở đầu

Game caro, còn được gọi là game "cờ caro", là một trò chơi ăn điểm cổ điển được phát triển từ những năm 1970. Đây là một trò chơi hai người chơi, mỗi người sử dụng một ký hiệu khác nhau (thường là X và O) để đánh lên bàn cờ vuông. Mục tiêu của mỗi người chơi là tạo được một đường liên tiếp 5 ô cùng loại trước đối phương.

Trò chơi caro đã trở thành một phần không thể thiếu trong văn hóa giải trí tại nhiều quốc gia, đặc biệt là Việt Nam. Nó không chỉ là một trò chơi giải trí đơn thuần mà còn là một công cụ phát triển các kỹ năng như logic, tư duy chiến lược và khả năng phản ứng nhanh. Người chơi phải liên tục tính toán, dự đoán và đưa ra các quyết định chiến lược để giành chiến thắng..

1.1.2 Lý do

- Tính phổ biến và tính truyền thống của game caro: Game caro là một trò chơi rất phổ biến và có bề dày lịch sử tại nhiều quốc gia, đặc biệt là Việt Nam. Việc xây dựng một ứng dụng game caro chất lượng cao sẽ giúp bảo tồn và phát triển di sản văn hóa này.
- Tiềm năng phát triển công nghệ: Game caro là một lĩnh vực lý tưởng để ứng dụng và phát triển các công nghệ mới như trí tuệ nhân tạo, học máy, xử lý ngôn ngữ tự nhiên, v.v. Điều này mang lại cơ hội nghiên cứu và cải tiến công nghệ.
- Tính thách thức và sáng tạo: Xây dựng một ứng dụng game caro hoàn chỉnh là một bài toán đầy thú vị và thách thức, đòi hỏi sự sáng tạo, kiến thức chuyên sâu và kỹ năng lập trình cao. Đây là cơ hội để chúng tôi phát triển các kỹ năng này.
- Tính ứng dụng trong thực tế: Game caro không chỉ là một trò chơi giải trí, mà còn có ứng dụng trong nhiều lĩnh vực như giáo dục, huấn luyện, quản lý. Việc xây dựng một ứng dụng game caro chất lượng cao có thể mở ra nhiều cơ hội ứng dụng thực tế.

1.1.3 Mục đích của trò chơi

Mục đích chính của trò chơi là để hai người chơi cố gắng xếp được một dãy liên tiếp của 5 quân cờ của mình trên một bảng cờ. Người chơi nào đạt được điều này trước thì sẽ thắng.

1.1.4 Bàn cờ và quân cờ

- **Bàn cờ:** Bàn cờ caro thường là một lưới ô vuông có kích thước 19x19, 15x15 hoặc 13x13. Tuy nhiên, trò chơi cũng có thể được chơi trên một lưới vô hạn lý thuyết.
- **Quân cờ:** Thường có hai loại quân cờ khác nhau để phân biệt người chơi, thường là X và O.

1.1.5 Luật chơi

- Hai người chơi lần lượt đặt quân cờ của mình vào một ô trống trên bàn cờ.
- Người chơi nào tạo được một hàng ngang, dọc, hoặc chéo gồm 5 quân cờ liên tiếp của mình trước thì sẽ thắng cuộc.
- Trò chơi kết thúc khi một người chơi thắng hoặc không còn ô trống nào để đặt quân cờ, lúc đó trò chơi sẽ hòa.

1.1.6 Chiến lược

- **Tấn công:** Cố gắng tạo ra các cơ hội để xếp được 5 quân cờ liên tiếp.
- **Phòng thủ:** Ngăn chặn đối phương không để họ tạo ra dãy 5 quân cờ liên tiếp.
- **Cạm bẫy:** Tạo ra các tình huống buộc đối phương phải chọn một cách đặt quân cờ khiến họ rơi vào thế thua.

1.1.7 Biến thể

Ngoài biến thể tiêu chuẩn, có một số biến thể khác của trò chơi caro:

- **Gomoku:** Luật chơi giống như caro nhưng có một số quy định chi tiết hơn về cách thức xếp quân cờ.
- **Renju:** Biến thể của Gomoku với thêm một số quy tắc để làm giảm lợi thế của người chơi đầu tiên.

1.1.8 Lịch sử và phổ biến

- Trò chơi caro có nguồn gốc từ Trung Quốc cổ đại và đã lan rộng ra khắp châu Á.
- Nó rất phổ biến tại Việt Nam, Hàn Quốc, Nhật Bản và một số nước khác.

1.1.9 Lợi ích của trò chơi

- Phát triển tư duy logic và chiến lược.
- Cải thiện khả năng tập trung và kiên nhẫn.
- Tạo ra những giờ phút giải trí và thư giãn.

1.2 Mục tiêu nghiên cứu

1.2.1 Phát triển trí tuệ nhân tạo

- Xây dựng thuật toán chơi cờ: Tạo ra các thuật toán có khả năng chơi cờ caro ở mức độ cao, từ đó kiểm tra và cải thiện khả năng của trí tuệ nhân tạo.
- Học máy và học sâu: Sử dụng các kỹ thuật học máy để phân tích và tối ưu hóa chiến lược chơi cờ caro.

1.2.2 Nghiên cứu chiến lược tối ưu hoá

- **Phân tích chiến lược:** Nghiên cứu và phát triển các chiến lược chơi tối ưu để giành chiến thắng.
- **Tối ưu hóa lượt đi:** Xác định những nước đi tốt nhất trong các tình huống cụ thể, từ đó cải thiện khả năng chơi của người chơi.

1.2.3 Phát triển giáo dục và đào tạo

- **Phương pháp giảng dạy:** Phát triển các phương pháp giảng dạy cờ caro để giúp học sinh và người chơi mới nhanh chóng nắm vững luật chơi và chiến lược cơ bản.
- **Phát triển kỹ năng tư duy:** Sử dụng cờ caro như một công cụ để phát triển tư duy logic, chiến lược và khả năng ra quyết định.

1.2.4 Nghiên cứu khoa học và toán học

- **Lý thuyết trò chơi:** Sử dụng cờ caro như một mô hình để nghiên cứu lý thuyết trò chơi và các khái niệm liên quan.
- **Toán học tổ hợp:** Phân tích các mô hình và cấu trúc toán học trong trò chơi để hiểu rõ hơn về các quy luật và chiến lược.

1.2.5 Phát triển và Cải thiện trò chơi

- **Tạo ra các biến thể mới:** Nghiên cứu để phát triển các biến thể mới của cờ caro, từ đó tạo ra nhiều sự lựa chọn cho người chơi.
- **Cải thiện trải nghiệm người chơi:** Nghiên cứu để tìm ra các cách cải thiện trải nghiệm chơi cờ caro, bao gồm giao diện người dùng, tính năng trò chơi và các công cụ hỗ trợ

1.2.6 Xã hội và văn hoá

- **Phân tích văn hóa:** Nghiên cứu vai trò của cờ caro trong văn hóa và xã hội của các quốc gia nơi trò chơi này phổ biến.
- **Giao lưu và kết nối:** Khám phá cách mà cờ caro có thể được sử dụng như một công cụ để giao lưu và kết nối giữa các nền văn hóa khác nhau

1.3 Phạm vi nghiên cứu

Phạm vi nghiên cứu về cờ caro có thể rất rộng, bao gồm nhiều lĩnh vực khác nhau từ khoa học máy tính, toán học, giáo dục đến văn hóa xã hội. Dưới đây là một số phạm vi nghiên cứu cụ thể:

1.3.1 Khoa học máy tính và trí tuệ nhân tạo

- **Thuật toán chơi cờ:** Nghiên cứu và phát triển các thuật toán để tạo ra một chương trình chơi cờ caro mạnh mẽ. Điều này bao gồm việc xây dựng các chiến lược, sử dụng cây tìm kiếm, và áp dụng kỹ thuật như Minimax, Alpha-beta pruning.
- **Học máy và Học sâu:** Ứng dụng các kỹ thuật học máy và học sâu để huấn luyện AI chơi cờ caro. Nghiên cứu này có thể bao gồm việc thu thập và phân tích dữ liệu từ các ván cờ, xây dựng các mô hình dự đoán và tối ưu hóa.
- **Phân tích và dự đoán kết quả:** Sử dụng các mô hình thống kê và học máy để dự đoán kết quả của các ván cờ dựa trên các nước đi và chiến lược cụ thể.

1.3.2 Toán học và lý thuyết trò chơi

- **Phân tích tổ hợp:** Nghiên cứu các cấu trúc tổ hợp của bàn cờ caro và các cách thức xếp quân cờ để đạt được chiến thắng.
- **Lý thuyết trò chơi:** Ứng dụng lý thuyết trò chơi để phân tích các chiến lược tối ưu và các quyết định của người chơi trong các tình huống khác nhau.
- **Mô hình toán học:** Xây dựng các mô hình toán học để mô phỏng và phân tích các chiến lược và kết quả của trò chơi.

1.3.3 Giáo dục và đào tạo

- **Phương pháp giảng dạy:** Phát triển các tài liệu và phương pháp giảng dạy để dạy cờ caro cho học sinh và người mới bắt đầu. Nghiên cứu này có thể bao gồm cả việc xây dựng các chương trình giáo dục và các công cụ hỗ trợ học tập.
- **Phát triển kỹ năng tư duy:** Sử dụng cờ caro như một công cụ để phát triển các kỹ năng tư duy logic, chiến lược, và ra quyết định. Nghiên cứu có thể tập trung vào việc đo lường và cải thiện những kỹ năng này thông qua việc chơi cờ.

1.3.4 Văn hoá và xã hội

- **Phân tích văn hóa:** Nghiên cứu vai trò của cờ caro trong các nền văn hóa khác nhau, đặc biệt là ở các quốc gia châu Á. Điều này bao gồm việc tìm hiểu về lịch sử, sự phổ biến và ý nghĩa văn hóa của trò chơi.
- **Giao lưu và kết nối:** Khám phá cách cờ caro có thể được sử dụng như một công cụ để kết nối và giao lưu giữa các nền văn hóa và cộng đồng khác nhau.

1.3.5 Phát triển và cải thiện trò chơi

- **Biến thể trò chơi:** Nghiên cứu và phát triển các biến thể mới của cờ caro để làm phong phú thêm trải nghiệm chơi.
- **Trải nghiệm người chơi:** Nghiên cứu để cải thiện trải nghiệm người chơi, bao gồm giao diện người dùng, tính năng trò chơi và các công cụ hỗ trợ trực tuyến.

1.3.6 Nghiên cứu tâm lý học

- **Hành vi người chơi:** Phân tích hành vi và động lực của người chơi trong trò chơi cờ caro, bao gồm cách họ ra quyết định và ứng phó với các tình huống khác nhau.
- **Tâm lý chiến thuật:** Nghiên cứu về các yếu tố tâm lý ảnh hưởng đến chiến thuật và hiệu suất chơi cờ.

CHƯƠNG II: PHÂN TÍCH YÊU CẦU

2.1 Yêu cầu chức năng

2.1.1 Giao diện người dùng

- **Bàn cờ:** Hiển thị bàn cờ với các ô vuông rõ ràng.

- **Quân cờ:** Biểu thị quân cờ của người chơi (X và O) trên bàn cờ.
- **Nút chức năng:** Bao gồm các nút như bắt đầu trò chơi mới, cài đặt, thoát, và hiển thị kết quả.
- **Thông báo:** Hiển thị thông báo cho người chơi về lượt đi, kết quả trận đấu (thắng, thua, hòa).

2.1.2 Quản lý trận đấu

- **Bắt đầu trận đấu:** Cho phép người chơi bắt đầu một trận đấu mới.
- **Lượt đi:** Quản lý lượt đi của từng người chơi, đảm bảo mỗi người chơi chỉ có thể đi một lượt tại một thời điểm.
- **Kiểm tra luật chơi:** Đảm bảo người chơi tuân thủ các quy tắc của trò chơi (chỉ được đặt quân cờ vào ô trống, kiểm tra điều kiện thắng).

2.1.3 Xử lý trận đấu

- **Cập nhật bàn cờ:** Cập nhật bàn cờ sau mỗi lượt đi của người chơi.
- **Kiểm tra điều kiện thắng:** Kiểm tra xem có bất kỳ người chơi nào xếp được 5 quân cờ liên tiếp (ngang, dọc, hoặc chéo) hay không.
- **Xử lý kết thúc trận đấu:** Xử lý các tình huống khi một người chơi thắng hoặc khi bàn cờ đầy (hòa).

2.1.4 Tùy chọn và Cài đặt

- **Chế Độ Chơi:** Cho phép người chơi chọn chế độ chơi (đấu với người, đấu với máy).
- **Cài Đặt AI:** Nếu có chế độ chơi với máy, cung cấp tùy chọn điều chỉnh mức độ khó của AI.

2.1.5 Trí tuệ nhân tạo

- **Thuật Toán Chơi Cờ:** Phát triển các thuật toán để AI có thể chơi cờ caro, bao gồm chiến lược tấn công và phòng thủ.
- **Mức Độ Khó:** Cung cấp nhiều cấp độ khó cho AI để người chơi có thể chọn.

2.1.6 Lịch sử và thống kê

- **Lưu Trữ Trận Đấu:** Lưu trữ lịch sử các trận đấu đã chơi.
- **Thống Kê Kết Quả:** Hiển thị thống kê kết quả trận đấu của người chơi, bao gồm số trận thắng, thua, hòa.

2.1.7 Hỗ trợ trực tuyến

- **Chơi Online:** Cung cấp chức năng chơi cờ caro trực tuyến với người chơi khác.
- **Chat:** Hỗ trợ tính năng chat để người chơi có thể giao tiếp với nhau trong trận đấu trực tuyến.

2.1.8 Trợ giúp và hướng dẫn

- **Hướng Dẫn Chơi:** Cung cấp hướng dẫn chi tiết về cách chơi cờ caro, bao gồm luật chơi và chiến lược cơ bản.
- **Trợ Giúp:** Cung cấp tính năng trợ giúp để người chơi có thể giải đáp các thắc mắc liên quan đến trò chơi.

2.1.9 Bảo mật và quản lý tài khoản

- **Đăng Nhập và Đăng Ký:** Cho phép người chơi tạo tài khoản, đăng nhập và quản lý thông tin cá nhân.
- **Bảo Mật Dữ Liệu:** Đảm bảo dữ liệu cá nhân và lịch sử trận đấu của người chơi được bảo mật.

2.1.10 Khả năng tùy biến và mở rộng

- **Tùy Chỉnh Giao Diện:** Cho phép người chơi tùy chỉnh giao diện trò chơi theo sở thích cá nhân.
- **Mở Rộng Tính Năng:** Dễ dàng tích hợp các tính năng mới và cập nhật trò chơi.

2.2 Yêu cầu phi chức năng

2.2.1 Hiệu năng (Performance)

- **Tốc độ phản hồi:** Ứng dụng phải có khả năng phản hồi các thao tác của người chơi trong vòng 1 giây.
- **Khả năng mở rộng:** Ứng dụng phải hỗ trợ ít nhất 1000 người dùng đồng thời mà không ảnh hưởng đến tốc độ phản hồi.
- **Tối ưu hóa tài nguyên:** Ứng dụng phải sử dụng tài nguyên (CPU, RAM) một cách hiệu quả để đảm bảo hiệu suất ổn định.
- **Thời gian tải trang:** Trang web hoặc ứng dụng phải tải trong vòng 3 giây trên các thiết bị thông thường và kết nối mạng trung bình.

2.2.2 Bảo mật (Security)

- **Xác thực và phân quyền:** Ứng dụng phải có hệ thống xác thực người dùng bằng mật khẩu mạnh, và phân quyền để đảm bảo chỉ những người dùng hợp lệ mới có thể truy cập các chức năng cụ thể.
- **Mã hóa dữ liệu:** Mọi dữ liệu nhạy cảm (như thông tin đăng nhập) phải được mã hóa cả khi lưu trữ và khi truyền tải.
- **Bảo vệ chống tấn công:** Ứng dụng phải có cơ chế bảo vệ chống lại các loại tấn công phổ biến như SQL injection, Cross-Site Scripting (XSS), và Cross-Site Request Forgery (CSRF).
- **Kiểm tra bảo mật:** Thực hiện các kiểm tra bảo mật thường xuyên để phát hiện và khắc phục kịp thời các lỗ hổng bảo mật.

2.2.3 Độ tin cậy (Reliability)

- **Thời gian hoạt động (Uptime):** Ứng dụng phải đạt thời gian hoạt động tối thiểu là 99.9%, đảm bảo rằng người dùng có thể truy cập và sử dụng ứng dụng bất kỳ lúc nào.
- **Khả năng khôi phục sau sự cố:** Ứng dụng phải có cơ chế sao lưu và khôi phục dữ liệu để đảm bảo dữ liệu không bị mất mát và hệ thống có thể khôi phục nhanh chóng sau các sự cố.
- **Tính nhất quán:** Dữ liệu và trạng thái trò chơi phải được duy trì nhất quán giữa các phiên làm việc và giữa các người dùng khác nhau.
- **Giám sát hệ thống:** Ứng dụng phải có hệ thống giám sát để theo dõi tình trạng hoạt động và cảnh báo ngay khi phát hiện các sự cố hoặc hành vi bất thường.

2.3 Các kịch bản sử dụng

2.3.1 Đăng ký tài khoản

Người dùng: Người chơi mới

Mô tả: Người chơi mới đăng ký tài khoản để có thể lưu lại kết quả và tiến

trình chơi của mình.

Bước 1: Người chơi truy cập trang đăng ký.

Bước 2: Người chơi nhập thông tin: tên và mật khẩu.

Bước 3: Người chơi xác nhận đăng ký.

Bước 4: Tài khoản được tạo và người chơi có thể đăng nhập.

2.3.2 Đăng nhập

Người dùng: Người chơi đã có tài khoản

Mô tả: Người chơi đăng nhập vào hệ thống để truy cập các chức năng nâng cao.

Bước 1: Người chơi truy cập trang đăng nhập.

Bước 2: Người chơi nhập tên đăng nhập và mật khẩu.

Bước 3: Hệ thống xác thực thông tin và cho phép truy cập nếu đúng.

2.3.3 Bắt đầu ván mới

Người dùng: Người chơi đã đăng nhập

Mô tả: Người chơi bắt đầu một ván cờ caro mới.

Bước 1: Người chơi chọn chế độ chơi (tạo phòng hoặc chơi với máy).

Bước 2: Người chơi (hoặc cả hai người chơi) tạo phòng và vào phòng chơi.

Bước 3: Bàn cờ được hiển thị và trò chơi bắt đầu.

2.3.4 Chơi ván cờ

Người dùng: Người chơi đã đăng nhập

Mô tả: Người chơi thực hiện các nước đi trên bàn cờ caro.

Bước 1: Người chơi lần lượt đặt các quân cờ (X hoặc O) vào các ô trên bàn cờ.

Bước 2: Hệ thống kiểm tra xem có người thắng cuộc hay không sau mỗi nước đi.

Bước 3: Trò chơi kết thúc khi một người chơi thắng hoặc bàn cờ đầy.

2.3.5 Đăng xuất

Người dùng: Người chơi đã đăng nhập

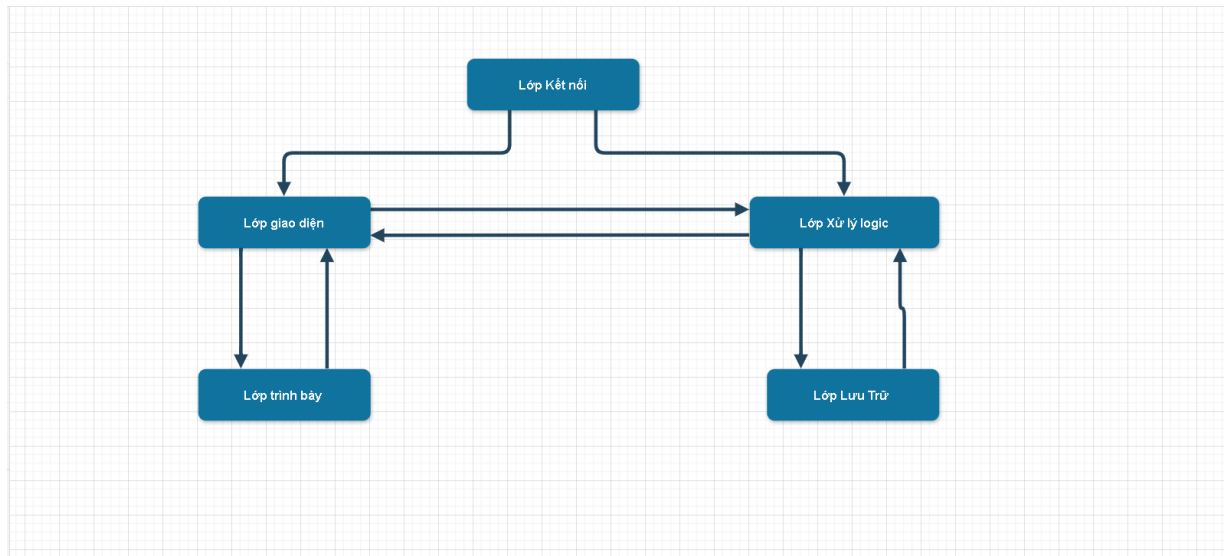
Mô tả: Người chơi đăng xuất khỏi hệ thống sau khi sử dụng.

Bước 1: Người chơi chọn tùy chọn "Đăng xuất".

Bước 2: Hệ thống kết thúc phiên làm việc của người chơi và trở về trang chủ.

CHƯƠNG III: THIẾT KẾ HỆ THỐNG

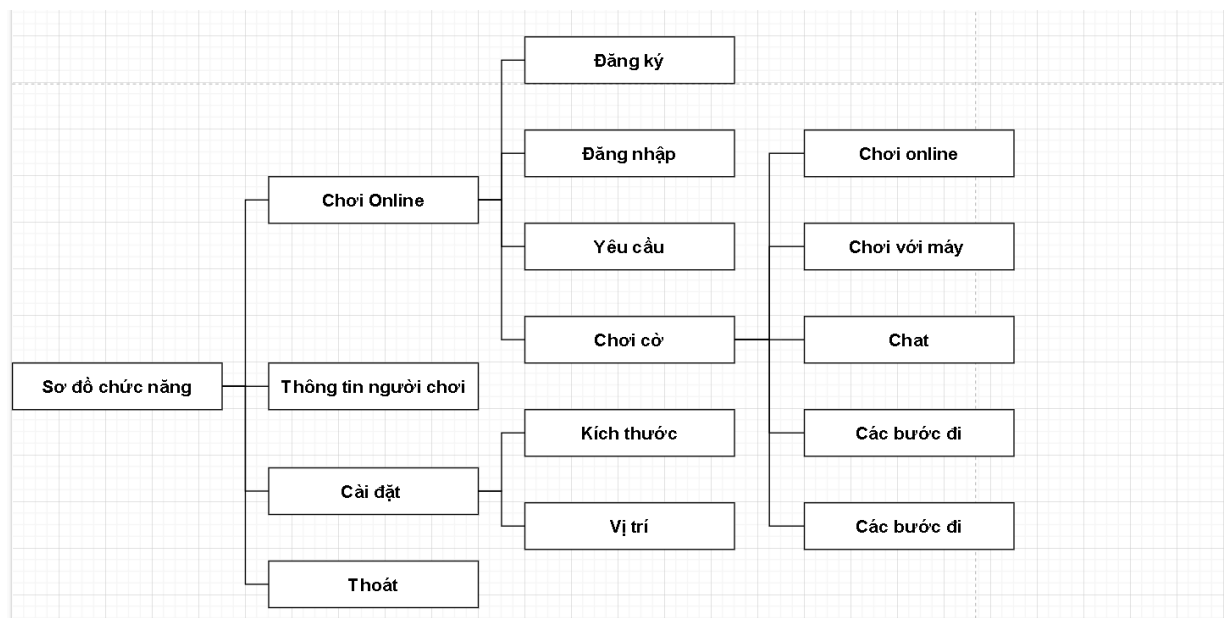
3.1 Kiến trúc hệ thống



Hình3.1. Kiến trúc hệ thống

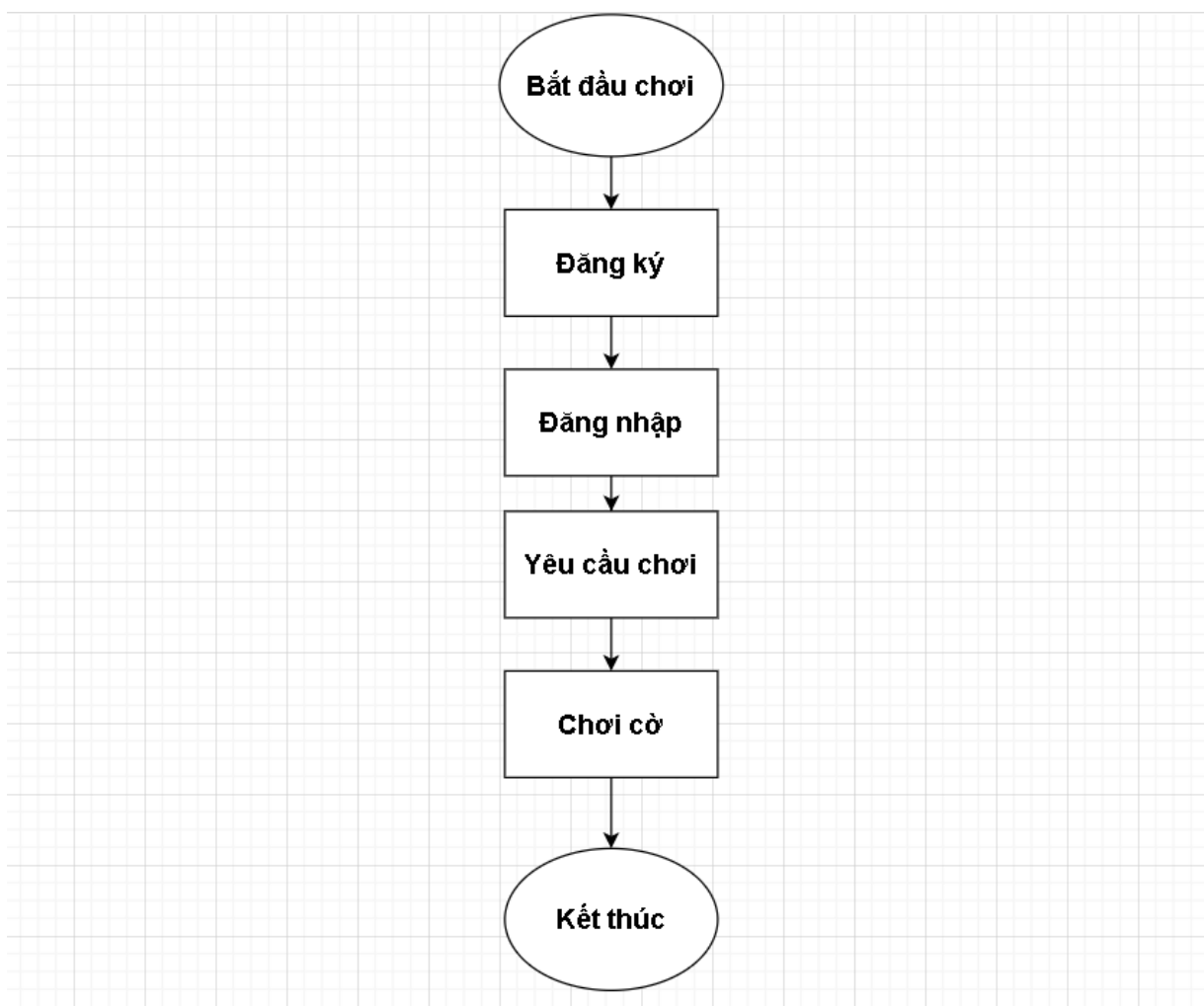
3.2 Thiết kế chi tiết

3.2.1 Sơ đồ chức năng



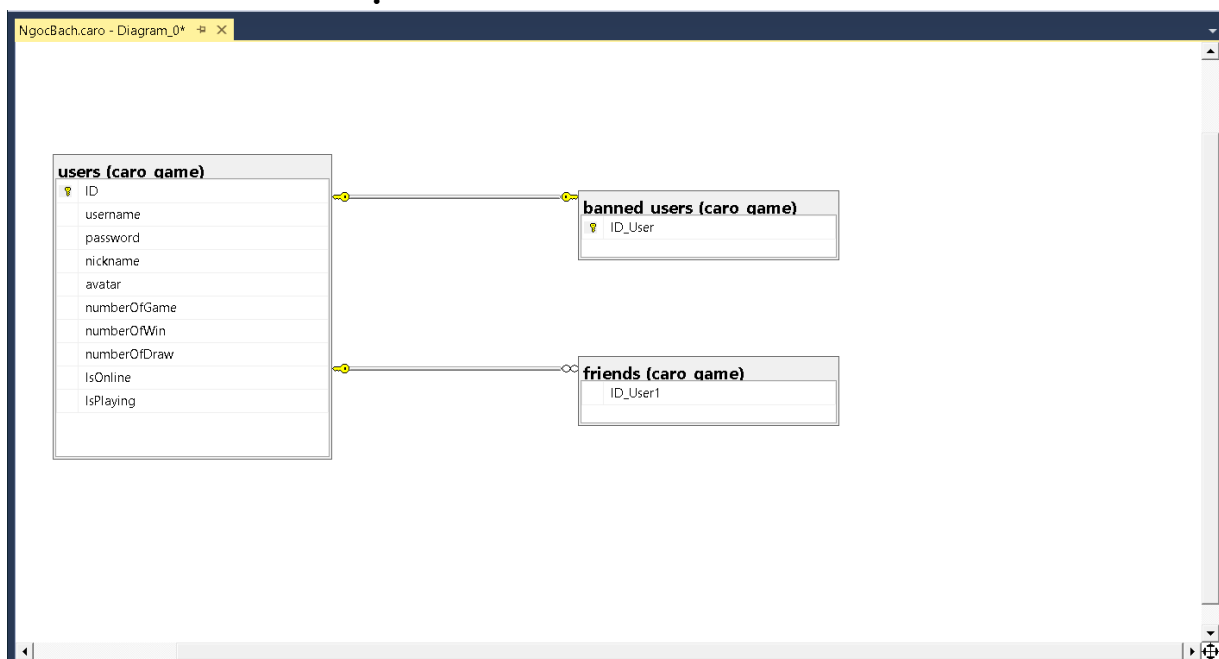
Hình3.2 Sơ đồ chức năng

3.2.2 Mô hình tổng quan



Hình3.3. Sơ đồ chức năng

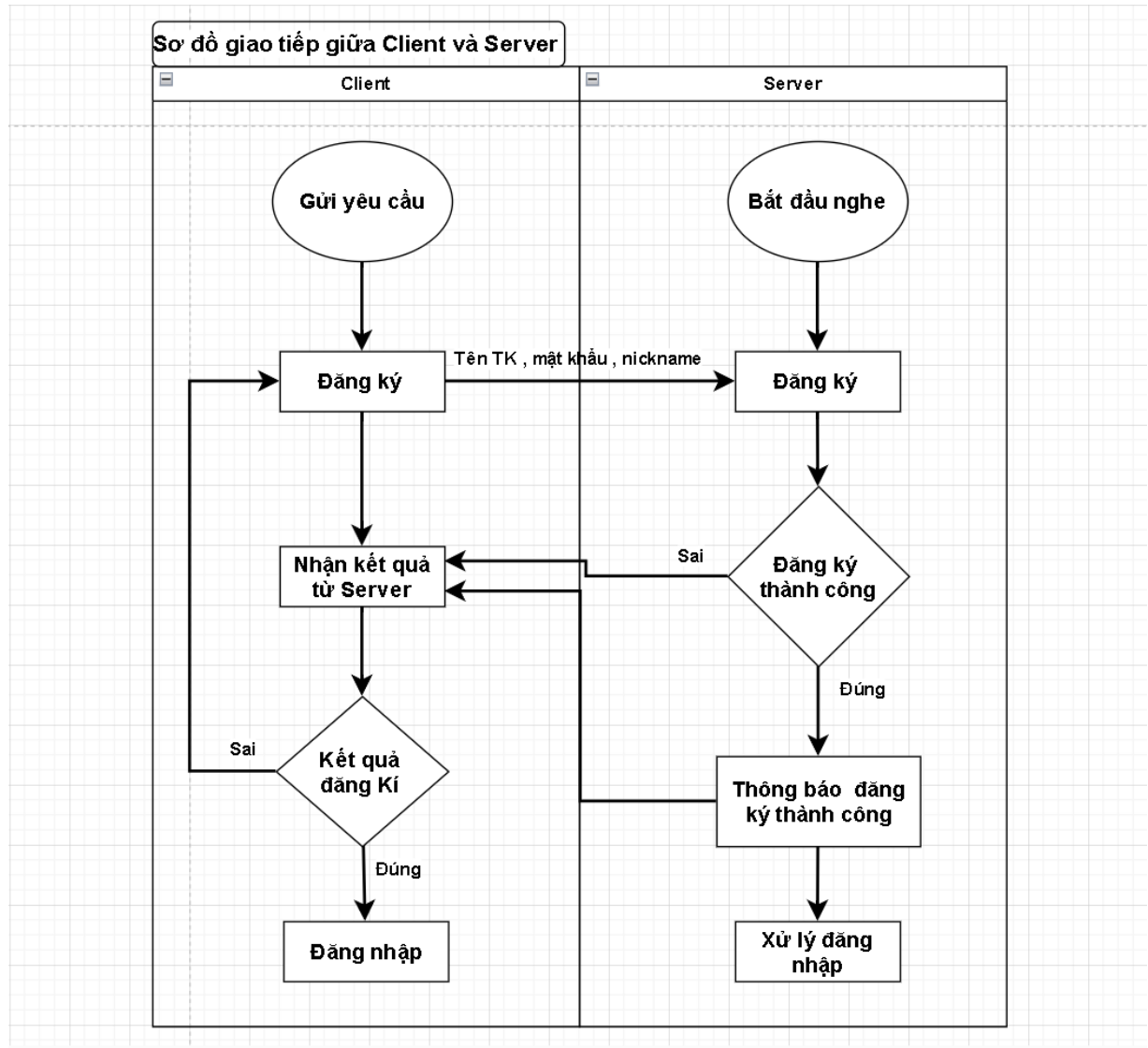
3.2.3 Thiết kế cơ sở dữ liệu



Hình3.4. Thiết kế cơ sở dữ liệu

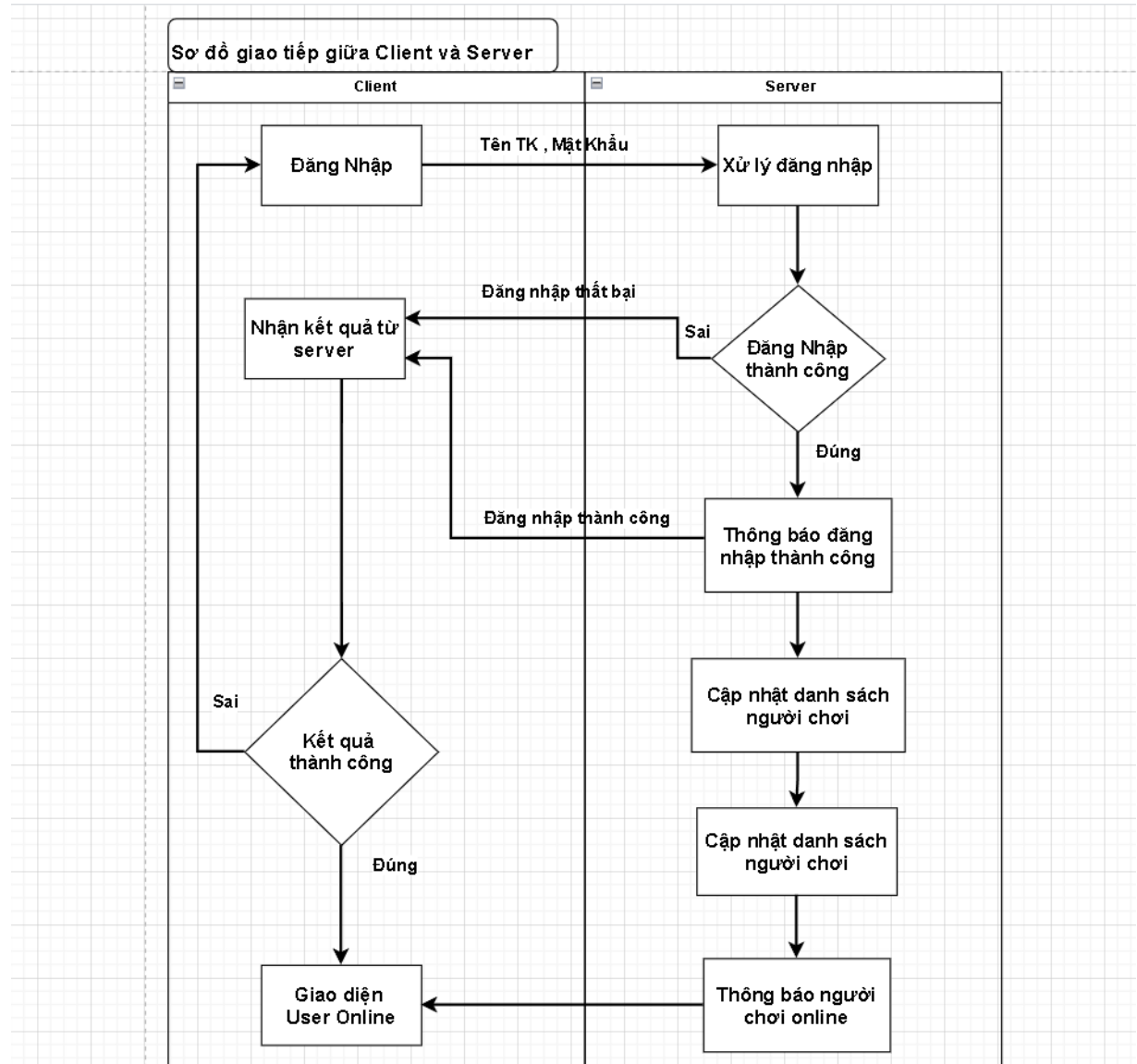
3.3 Thiết kế các module và các luồng dữ liệu

3.3.1 Sơ đồ đăng ký



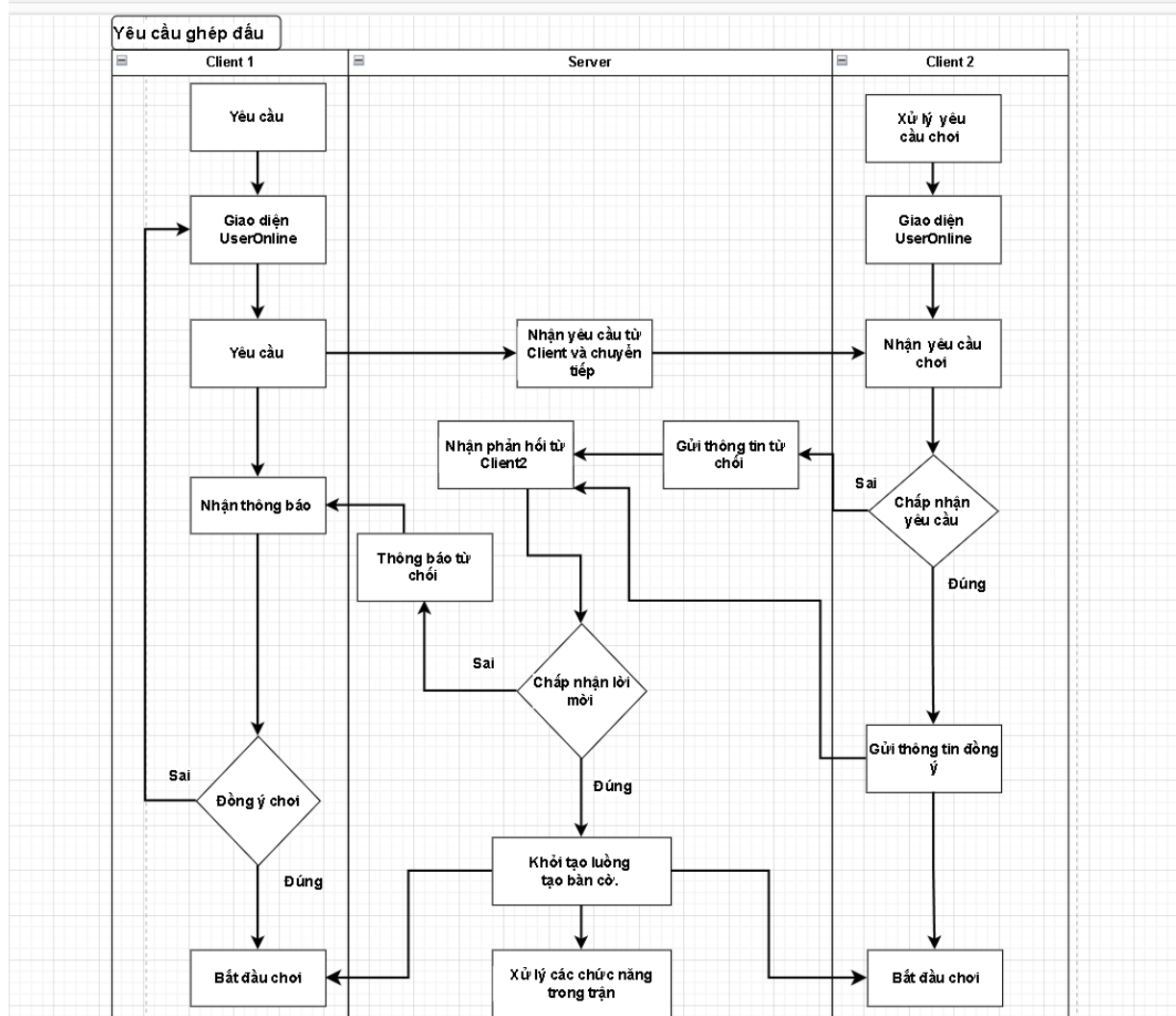
Hình 3.5. Sơ đồ đăng ký

3.3.2 Sơ đồ đăng nhập



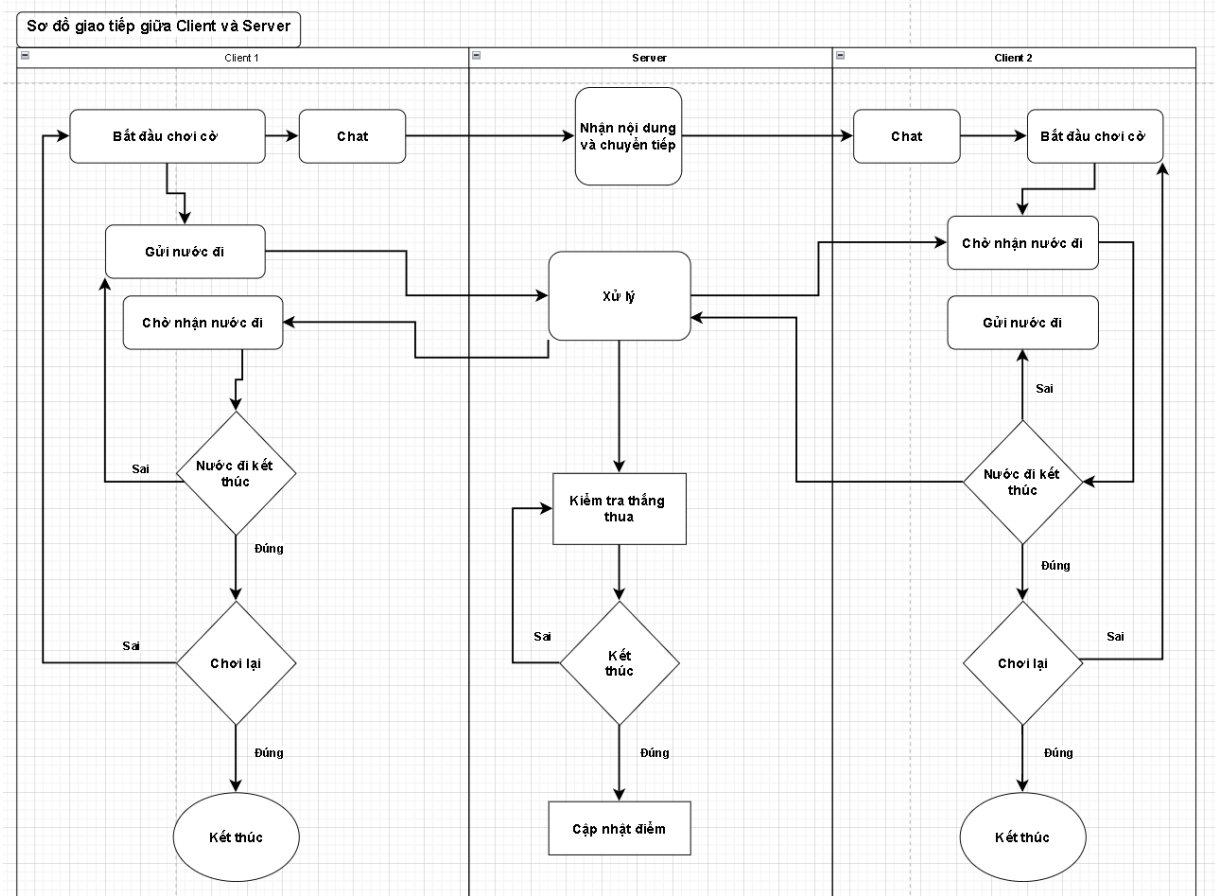
Hình 3.6. Sơ đồ đăng nhập

3.3.3 Sơ đồ yêu cầu ghép đấu



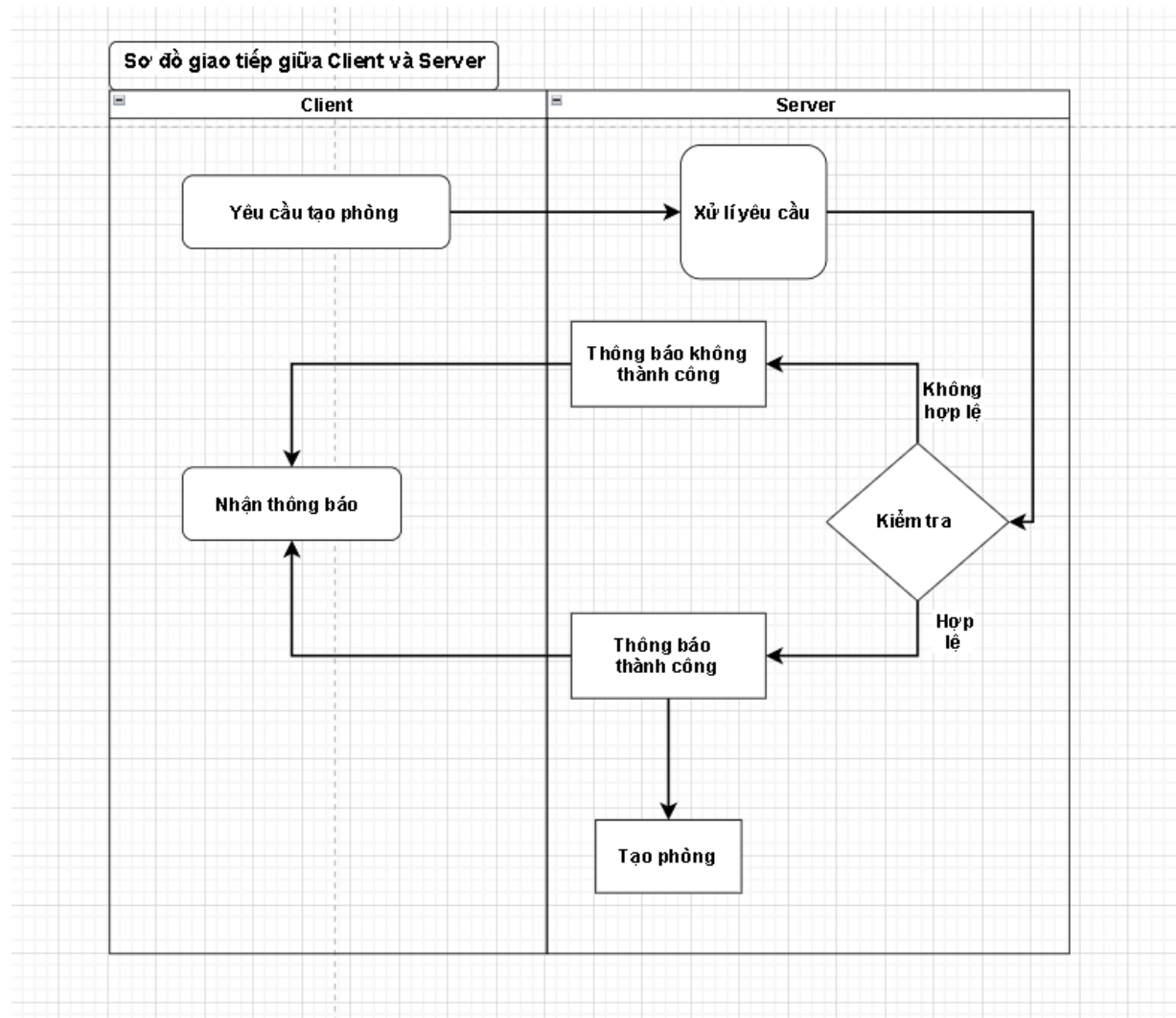
Hình3.7. Sơ đồ yêu cầu ghép đấu

3.3.4 Sơ đồ các nước đi



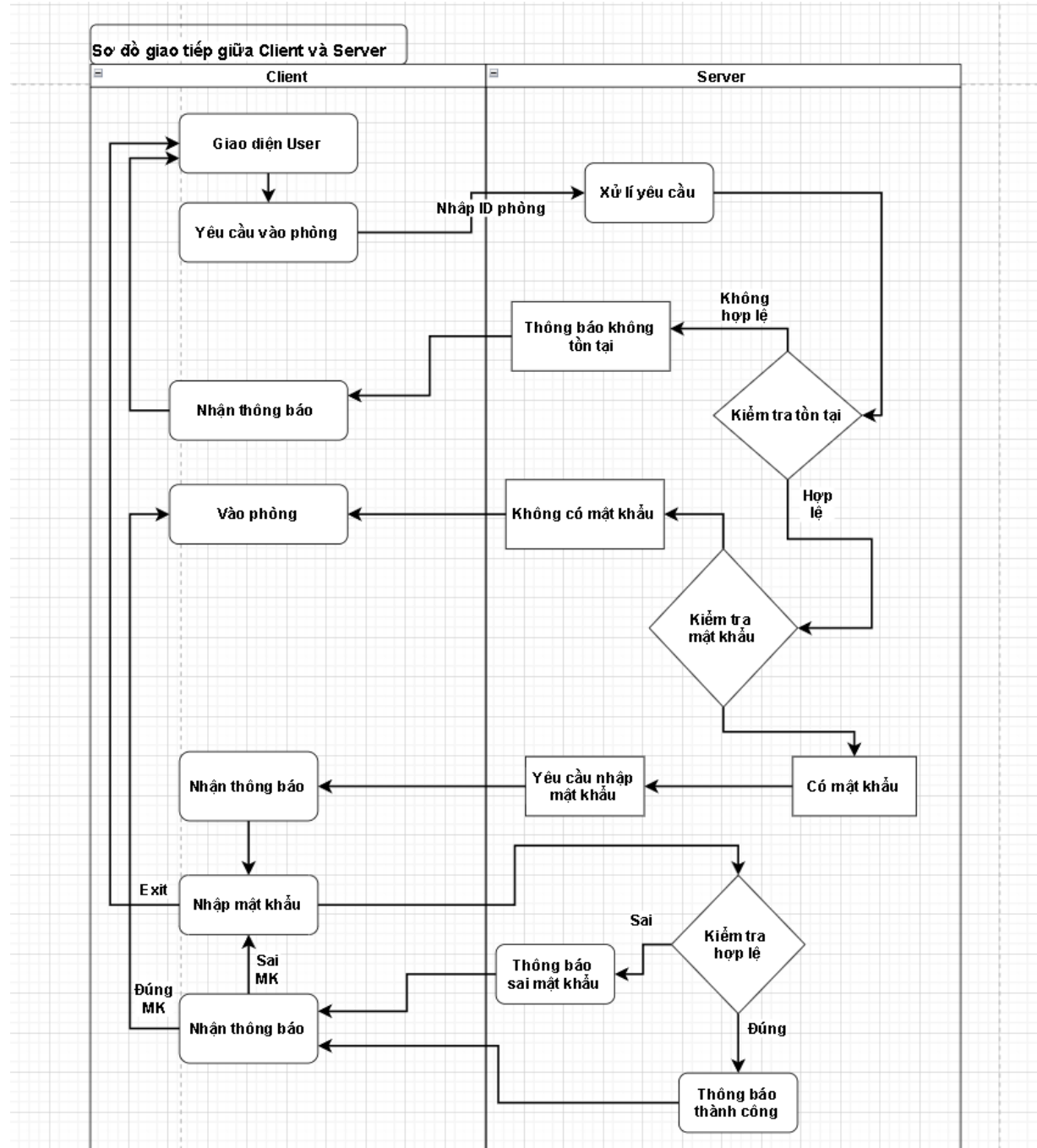
Hình 3.8. Sơ đồ nước đi

3.3.5 Sơ đồ tạo phòng



Hình 3.9 Sơ đồ tạo phòng

3.3.6 Sơ đồ vào phòng



Hình 3.10. Sơ đồ vào phòng

Chương IV: MÃ NGUỒN

4.1 Môi trường phát triển

4.1.1 Công cụ phát triển

- **IDE (Integrated Development Environment):**
 - **NetBeans:** Dùng để phát triển, viết mã nguồn, gỡ lỗi và xây dựng dự án Java.
- **Build Tools:**
 - **Apache Ant:** Công cụ xây dựng dự án, được sử dụng để tự động hóa các tác vụ xây dựng như biên dịch mã nguồn, chạy thử nghiệm và đóng gói ứng dụng.
- **Database Management Systems (DBMS):**
 - **MySQL:** Hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến.
- **Libraries and Frameworks:**
 - **JDBC (Java Database Connectivity):** API chuẩn để kết nối và làm việc với các cơ sở dữ liệu quan hệ.
 - **Swing:** Thư viện tiêu chuẩn của Java để xây dựng giao diện người dùng.

4.1.2 Môi trường phát triển

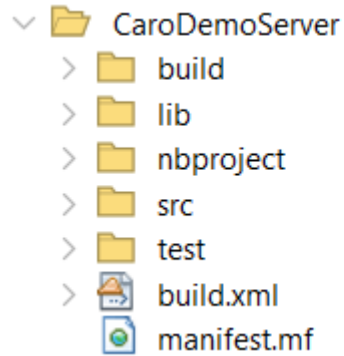
- **Operating Systems:**
 - **Windows:** Phổ biến và dễ sử dụng, phù hợp với nhiều công cụ phát triển phần mềm.
 - **Linux (Ubuntu, CentOS, Fedora):** Được nhiều lập trình viên và sysadmin ưa chuộng vì tính ổn định và khả năng tùy biến.
 - **macOS:** Hệ điều hành dành cho các thiết bị của Apple, phù hợp với các lập trình viên phát triển ứng dụng cho hệ sinh thái của Apple.
- **Java Development Kit (JDK):**
 - **JDK 8, JDK 11, JDK 17:** Các phiên bản LTS (Long-Term Support) của JDK.

4.2 Cấu trúc mã nguồn

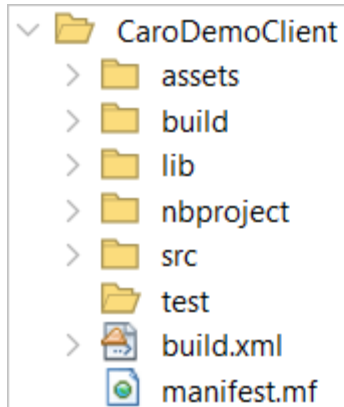
4.2.1 Thư mục gốc

Thư mục chứa các tệp và thư mục chính của dự án

- **Server:**



- **Client:**

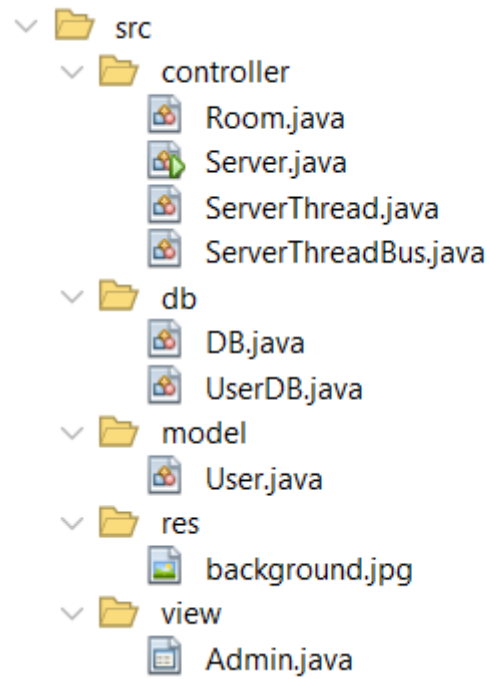


Hình 4.1. Thư mục gốc

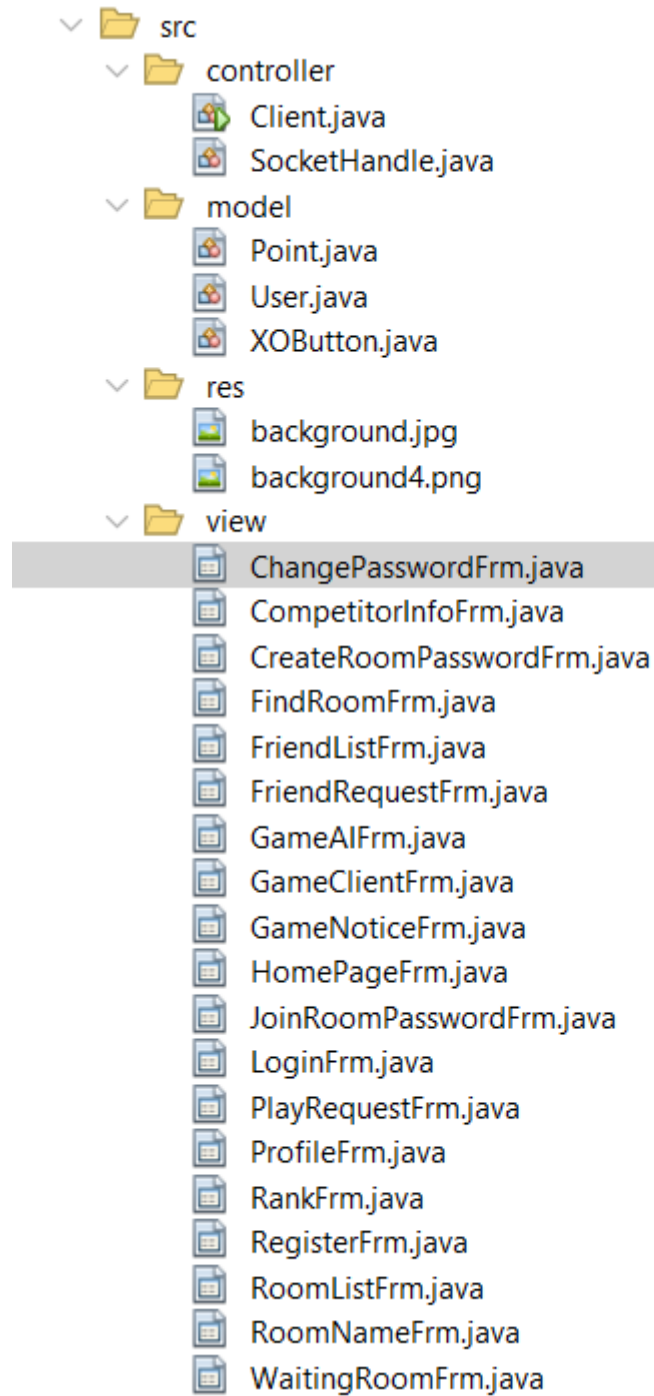
4.2.2 Thư mục src

Chứa các mã nguồn chính của ứng dụng

- **Server:**



- **Client:**

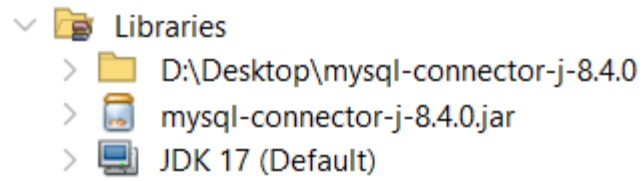


Hình 4.2. Thư mục src

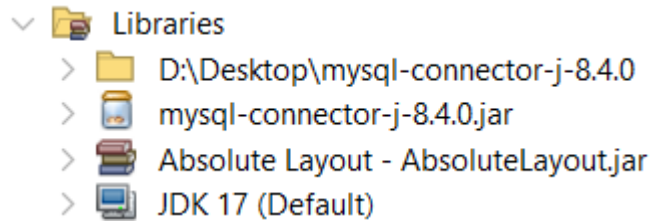
4.2.3 Thư mục lib

Chức các thư viện bên ngoài (JAR files) mà dự án phụ thuộc vào

- **Server:**



- **Client:**

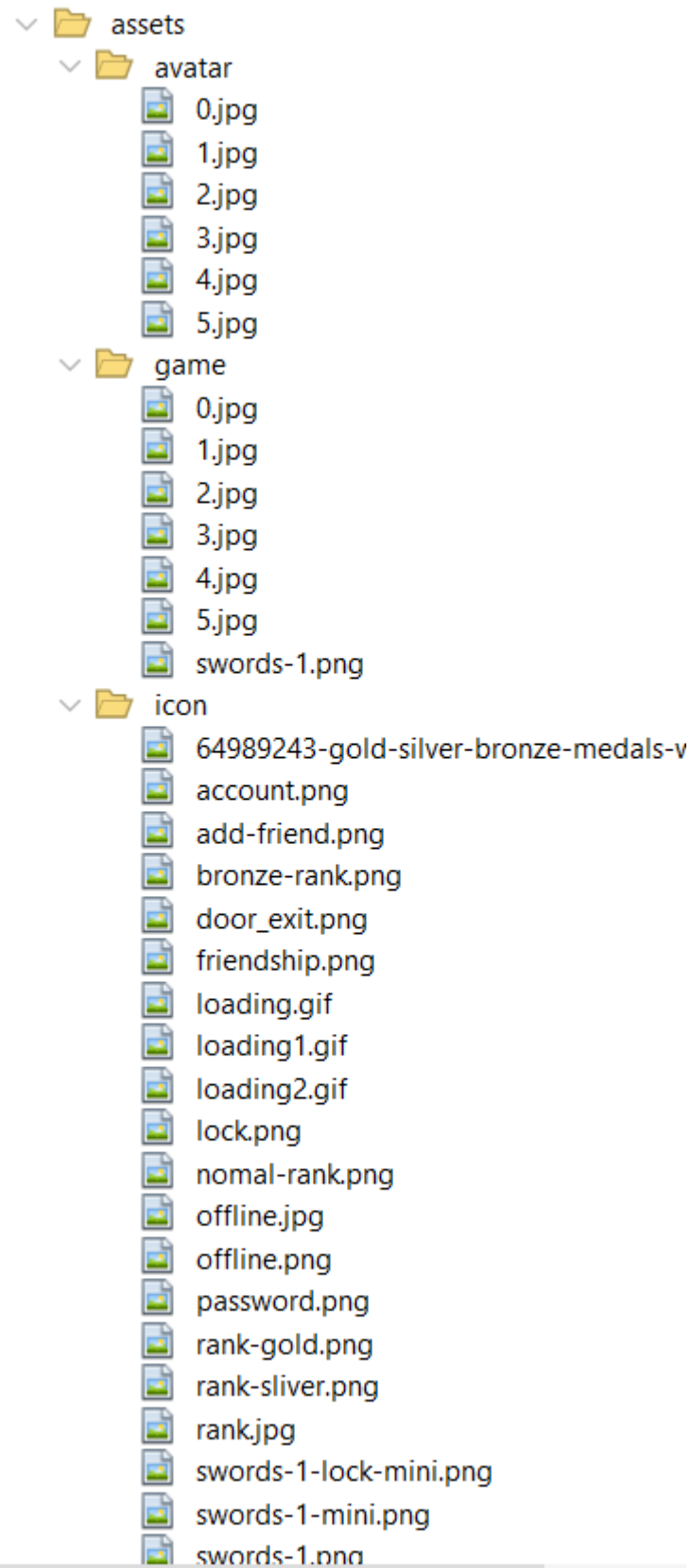


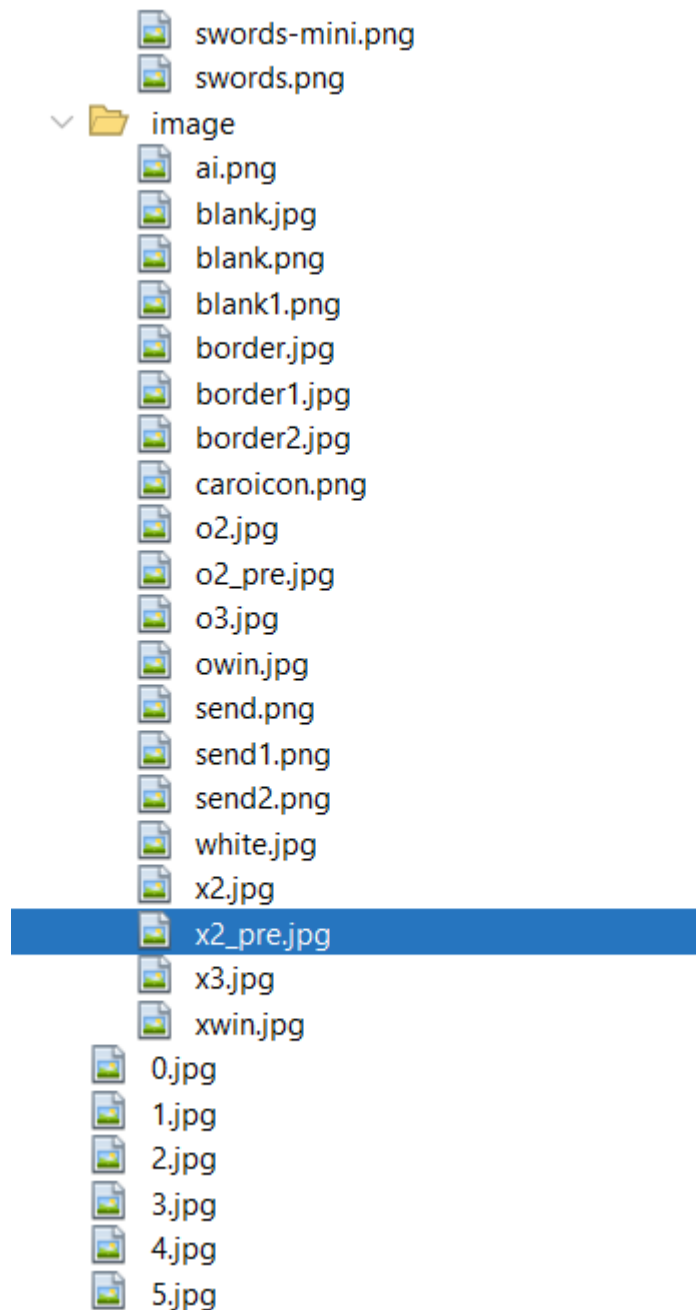
Hình 4.3. Thư mục lib

4.2.4 Thư mục assets

Chứa các tài nguyên hình ảnh

- **Client:**

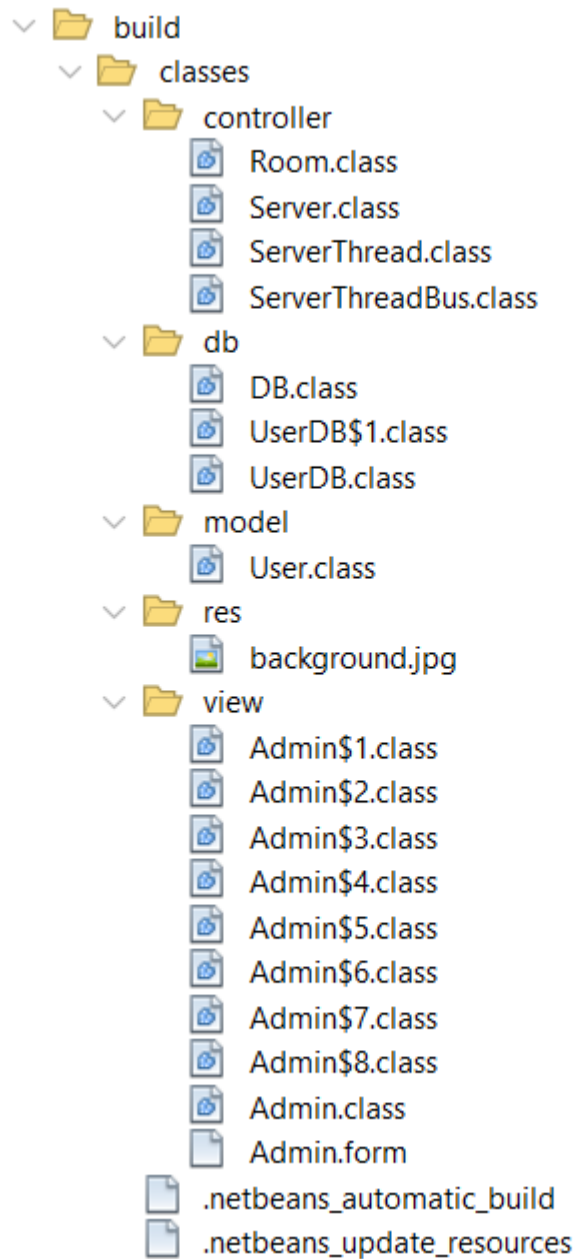




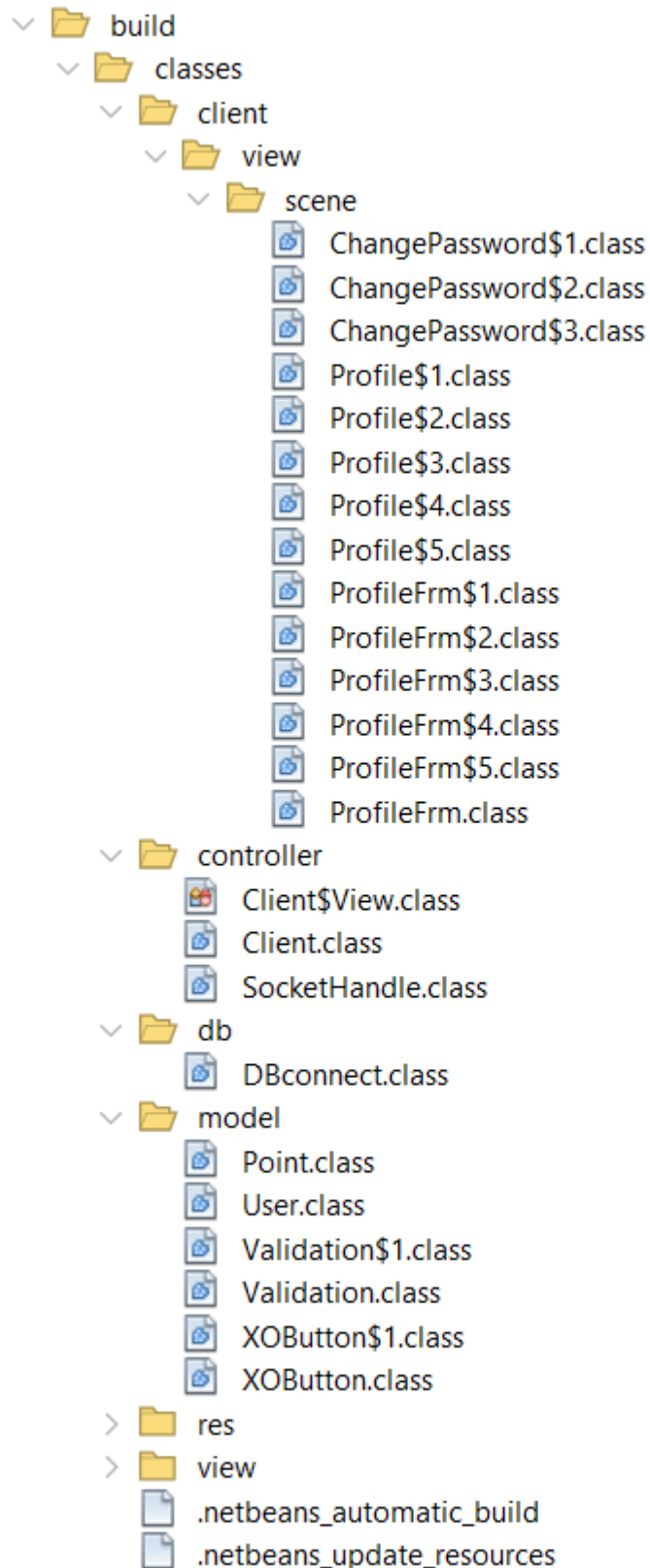
Hình 4.4. Thư mục assets

4.2.5 Thư mục build

- **Server:**



- **Client:**



Hình 4.5. Thư mục build

4.3 Quá trình cài đặt và triển khai

4.3.1 Yêu cầu hệ thống

- **Java Development Kit (JDK):** Phiên bản 8 hoặc cao hơn.
- **MySQL Database:** Phiên bản 5.7 hoặc cao hơn.
- **Apache Ant:** Để tự động hóa quá trình xây dựng và triển khai.
- **NetBeans IDE:** Để phát triển và quản lý dự án (tùy chọn).

MySQL Connector/J: Thư viện JDBC để kết nối Java với MySQL

4.3.2 Thiết lập môi trường phát triển

- **Cài đặt JDK:**

Tải và cài đặt JDK từ [Oracle](#) hoặc [OpenJDK](#).

Thiết lập biến môi trường JAVA_HOME và thêm bin của JDK vào biến PATH.

- **Cài đặt MySQL:**

Tải và cài đặt MySQL từ [MySQL Community Downloads](#).

Thiết lập tài khoản và mật khẩu cho MySQL.

- **Cài đặt Apache Ant:**

Tải và cài đặt Apache Ant từ [Ant Apache](#).

Thiết lập biến môi trường ANT_HOME và thêm bin của Ant vào biến PATH.

4.3.3 Thiết lập dự án

- **Cấu hình MySQL:**

Tạo cơ sở dữ liệu:

```
1 • create schema caro_game
```

Tạo các bảng:

```

4  CREATE TABLE `user` (
5      ID int AUTO_INCREMENT PRIMARY KEY,
6      `username` varchar(255) UNIQUE,
7      `password` varchar(255),
8      nickname varchar(255),
9      avatar varchar(255),\
10     numberOfGame int DEFAULT 0,
11     numberOfWin int DEFAULT 0,
12     numberOfDraw int DEFAULT 0,
13     IsOnline int DEFAULT 0,
14     IsPlaying int DEFAULT 0
15 );
16 CREATE TABLE friend(
17     ID_User1 int NOT NULL,
18     ID_User2 int NOT NULL,
19     FOREIGN KEY (ID_User1) REFERENCES `user` (ID),
20     FOREIGN KEY (ID_User2) REFERENCES `user` (ID),
21     CONSTRAINT PK_friend PRIMARY KEY (ID_User1,ID_User2)
22 );
23 CREATE TABLE BANNED_USER(
24     ID_User int PRIMARY KEY NOT NULL,
25     FOREIGN KEY (ID_User) REFERENCES `user` (ID)
26 );

```

Hình4.6.Cấu hình CSDL

- **Cấu hình kết nối Database:**

Tại file DB.java, thiết lập kết nối đến MySQL

```
private static final String URL = "jdbc:mysql://localhost:3306/caro_game";
private static final String USER = "root"; // TODO FILL YOUR DATABASE USER
private static final String PASSWORD = "123456"; // TODO FILL YOUR DATABASE PASSWORD
private static Connection connection = null;

public static Connection getConnection() {
    if (connection == null) {
        try {
            connection = DriverManager.getConnection(URL, USER, PASSWORD);
            System.out.println("Connection successful!");
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("Connection failed!");
        }
    }
    return connection;
}
```


Hình4.7.Cấu hình kết nối Database

4.3.4 Các đoạn mã nguồn quan trọng

- **Kết nối Database (DB.java):** Lớp này cung cấp một phương thức tĩnh để tạo và quản lý kết nối với cơ sở dữ liệu MySQL, giúp cho việc truy cập dữ liệu dễ dàng hơn và giảm bớt việc lặp lại mã kết nối trong toàn bộ dự án.

```
private static final String URL = "jdbc:mysql://localhost:3306/caro_game";
private static final String USER = "root"; // TODO FILL YOUR DATABASE USER
private static final String PASSWORD = "123456"; // TODO FILL YOUR DATABASE PAS
private static Connection connection = null;

public static Connection getConnection() {
    if (connection == null) {
        try {
            connection = DriverManager.getConnection(URL, USER, PASSWORD);
            System.out.println("Connection successful!");
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("Connection failed!");
        }
    }
    return connection;
}
```

Hình4.8.Kết nối Database

- **Kết nối client-server, truy cập dữ liệu người dùng:**
 - **Kết nối client-server qua gói Socket:**
 - Thiết lập máy chủ (Server):
Trong Server.java, thiết lập một socket lắng nghe kết nối của máy khách trên cổng 7777. Quản lý các luồng để xử lý kết nối máy khách. Có tối thiểu 10 luồng và tối đa 100 luồng. Sử dụng hàng đợi để giữ các tác vụ đang chờ thực thi.

```

public class Server {

    public static volatile ServerThreadBus serverThreadBus;
    public static Socket socketOfServer;
    public static int ROOM_ID;
    public static volatile Admin admin;

    public static void main(String[] args) {
        ServerSocket listener = null;
        serverThreadBus = new ServerThreadBus();
        System.out.println("Server is waiting to accept user...");
        int clientNumber = 0;
        ROOM_ID = 100;

        try {
            listener = new ServerSocket(7777);
        } catch (IOException e) {
            e.printStackTrace();
            System.exit(1);
        }

        ThreadPoolExecutor executor = new ThreadPoolExecutor(
            10, //min lương
            100, //max lương
            10, //waiting time
            TimeUnit.SECONDS,
            new ArrayBlockingQueue<>(8) //sức chứa hàng đợi
        );
    }
}

```

Hình 4.9. Thiết lập máy chủ

Liên tục chờ kết nối của máy khách. Khi máy khách kết nối, nó tạo ra một ServerThread để xử lý kết nối và thêm nó vào serverThreadBus.

```

try {
    while (true) {
        socketOfServer = listener.accept();
        System.out.println(socketOfServer.getInetAddress().getHostAddress());
        ServerThread serverThread = new ServerThread(socketOfServer, clientNumber++);
        serverThreadBus.add(serverThread);
        System.out.println("Số thread đang chạy là: " + serverThreadBus.getLength());
        executor.execute(serverThread);
    }
} catch (IOException ex) {
    ex.printStackTrace();
} finally {
    try {
        listener.close();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
}
}

```

- **Đăng nhập/Đăng ký:** Chạy Client.java để kết nối đến Server và tiến hành đăng nhập/đăng ký. Thông tin đăng nhập/đăng ký được gửi đến cho SocketHandle xử lý.

```

private void loginButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String taiKhoan = usernameValue.getText();
        if (taiKhoan.isEmpty())
            throw new Exception("Vui lòng nhập tên tài khoản");
        String matKhai = String.valueOf(passwordValue.getPassword());
        if (matKhai.isEmpty())
            throw new Exception("Vui lòng nhập mật khẩu");
        Client.closeAllViews();
        Client.openView(Client.View.GAME_NOTICE, "Đăng nhập", "Đang xác thực thông tin đăng nhập");
        Client.socketHandle.write("client-verify," + taiKhoan + "," + matKhai);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(rootPane, ex.getMessage());
    }
}

```

Hình4.10 .Đăng nhập , đăng ký

- **Xử lý trong SocketHandle:** Xử lý giao tiếp phía máy khách với máy chủ, xử lý các thông báo nhận được từ máy chủ. Ví dụ ở đây là khi kết nối đến server và đăng nhập thành công.

```

public void run() {
    try {
        socketOfClient = new Socket("127.0.0.1", 7777);
        System.out.println("Kết nối thành công!");
        outputWriter = new BufferedWriter(new OutputStreamWriter(socketOfClient.getOutputStream()));
        BufferedReader inputReader = new BufferedReader(new InputStreamReader(socketOfClient.getInputStream()));
        String message;
        while (true) {
            message = inputReader.readLine();
            if (message == null) {
                break;
            }
            String[] messageSplit = message.split(",");
            if (messageSplit[0].equals("server-send-id")) {
                int serverId = Integer.parseInt(messageSplit[1]);
            }
            //Đăng nhập thành công
            if (messageSplit[0].equals("login-success")) {
                System.out.println("Đăng nhập thành công");
                Client.closeAllViews();
                Client.user = getUserFromString(1, messageSplit);
                Client.openView(Client.View.HOME PAGE);
            }
        }
    }
}

```

Hình4.11.SocKetHandle

- **Phương thức verifyUser truy cập dữ liệu người dùng:** Kiểm tra xem thông tin User được cung cấp khi đăng nhập (tên người dùng và mật khẩu) có khớp với cơ sở dữ liệu hay không. Nếu khớp, nó sẽ trả về một đối tượng user có thông tin chi tiết được lấy từ cơ sở dữ liệu. Nếu không, nó sẽ trả về null.

```

public User verifyUser(User user) {
    try {
        PreparedStatement preparedStatement = con.prepareStatement("SELECT *\n"
            + "FROM user\n"
            + "WHERE username = ?\n"
            + "AND password = ?"
        );
        preparedStatement.setString(1, user.getUsername());
        preparedStatement.setString(2, user.getPassword());
        ResultSet rs = preparedStatement.executeQuery();
        if (rs.next()) {
            return new User(rs.getInt(1),
                rs.getString(2),
                rs.getString(3),
                rs.getString(4),
                rs.getString(5),
                rs.getInt(6),
                rs.getInt(7),
                rs.getInt(8),
                (rs.getInt(9) != 0),
                (rs.getInt(10) != 0),
                getRank(rs.getInt(1)));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return null;
}

```

Khởi tạo một đối tượng User với tất cả các thuộc tính.

```

public class User {
    private int ID;
    private String username;
    private String password;
    private String nickname;
    private String avatar;
    private int numberOfGame;
    private int numberOfWin;
    private int numberOfDraw;
    private boolean isOnline;
    private boolean isPlaying;
    private int rank;

    public User() {
    }

    public User(int ID, String username, String password, String nickname, String avatar, int numberOfGame, int numberOfWin, int numberOfDraw, boolean isOnline, boolean isPlaying, int rank) {
        this.ID = ID;
        this.username = username;
        this.password = password;
        this.nickname = nickname;
        this.avatar = avatar;
        this.numberOfGame = numberOfGame;
        this.numberOfWin = numberOfWin;
        this.numberOfDraw = numberOfDraw;
        this.isOnline = isOnline;
        this.isPlaying = isPlaying;
        this.rank = rank;
    }
}

```

- **Thuật toán cờ caro:**
 - **Code thiết lập thời gian:**
 - **Thời gian cho 1 lượt đi:** Thời gian cho 1 lượt đi của người chơi là 30 giây, sau 30 giây nếu người chơi chưa đánh ra một

nước thì bị xử thua và hệ thống tiến hành thiết lập ván chơi mới.

```
// Thiết lập bộ đếm thời gian cho từng lượt đánh
second = 30;
minute = 0;
timer = new Timer(1000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String temp = minute < 10 ? "0" + minute : minute.toString();
        String temp1 = second < 10 ? "0" + second : second.toString();
        countdownLabel.setText("Thời gian lượt đánh: " + temp + ":" + temp1);

        if (second == 0) {
            if (minute == 0) {
                // Người chơi hiện tại thua do hết thời gian
                second = 30;
                minute = 0;
                try {
                    Client.openView(Client.View.GAME_CLIENT, "Bạn đã thua do quá thời gian", "Đang thiết lập ván chơi mới");
                    increaseWinMatchToCompetitor();
                    Client.socketHandle.write("lose,");
                } catch (IOException ex) {
                    JOptionPane.showMessageDialog(rootPane, ex.getMessage());
                }
            } else {
                minute--;
                second = 59;
            }
        } else {
            second--;
        }
    }
});
```

Hình4.12.Thời gian cho 1 lượt đi

- **Thời gian cho toàn trận đấu:** Thời gian cho 1 trận đấu là 10 phút, sau 10 phút nếu 2 người chơi chưa phân thắng thua thì trận đấu sẽ xử hòa và hệ thống tiến hành thiết lập ván chơi mới.

```
// Thiết lập bộ đếm thời gian cho toàn bộ trận đấu
minuteGame = 10;
secondGame = 0;
timerGame = new Timer(1000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String temp = minuteGame < 10 ? "0" + minuteGame : minuteGame.toString();
        String temp1 = secondGame < 10 ? "0" + secondGame : secondGame.toString();
        countdownLabel1.setText("Thời gian trận đấu còn: " + temp + ":" + temp1);

        if (minuteGame == 0 && secondGame == 0) {
            // Trận đấu hòa do hết thời gian
            minuteGame = 10;
            secondGame = 0;
            try {
                Client.openView(Client.View.GAME_CLIENT, "Ván chơi hòa", "Đang thiết lập ván chơi mới");
                displayDrawGame();
                Client.socketHandle.write("draw,");
            } catch (IOException ex) {
                JOptionPane.showMessageDialog(rootPane, ex.getMessage());
            }
        } else {
            if (secondGame == 0) {
                secondGame = 59;
                minuteGame--;
            } else {
                secondGame--;
            }
        }
    }
});
```

Hình 4.13. Thời gian cho toàn trận đấu

- **Code kiểm tra nước đi, kiểm tra thắng/thua:**
- **Kiểm tra hàng:** Kiểm tra các hàng ngang để xác định xem có 5 ô liên tiếp trong một hàng không.

```

public int checkRow() {
    int win = 0, hang = 0;
    boolean check = false;
    List<JButton> list = new ArrayList<>();
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            if (check) {
                if (competitorMatrix[i][j] == 1) {
                    hang++;
                    list.add(button[i][j]);
                    if (hang > 4) {
                        for (JButton jButton : list) {
                            jButton.setDisabledIcon(new ImageIcon(winItem[numberOfMatch % 2]));
                        }
                        win = 1;
                        break;
                    }
                    continue;
                } else {
                    list = new ArrayList<>();
                    check = false;
                    hang = 0;
                }
            }
            if (competitorMatrix[i][j] == 1) {
                check = true;
                list.add(button[i][j]);
                hang++;
            } else {
                list = new ArrayList<>();
                check = false;
            }
        }
        list = new ArrayList<>();
        hang = 0;
    }
    return win;
}

```

Hình 4.14. Kiểm tra hàng

- **Kiểm tra cột:** Kiểm tra các cột dọc để xác định xem có 5 ô liên tiếp trong một cột không.

```

public int checkColumn() {
    int win = 0, cot = 0;
    boolean check = false;
    List<JButton> list = new ArrayList<>();
    for (int j = 0; j < size; j++) {
        for (int i = 0; i < size; i++) {
            if (check) {
                if (competitorMatrix[i][j] == 1) {
                    cot++;
                    list.add(button[i][j]);
                    if (cot > 4) {
                        for (JButton jButton : list) {
                            jButton.setDisabledIcon(new ImageIcon(winItem[numberOfMatch % 2]));
                        }
                        win = 1;
                        break;
                    }
                    continue;
                } else {
                    check = false;
                    cot = 0;
                    list = new ArrayList<>();
                }
            }
            if (competitorMatrix[i][j] == 1) {
                check = true;
                list.add(button[i][j]);
                cot++;
            } else {
                list = new ArrayList<>();
                check = false;
            }
        }
        list = new ArrayList<>();
        cot = 0;
    }
    return win;
}

```

Hình 4.15. Kiểm tra cột

- **Kiểm tra chéo phải:** Kiểm tra các đường chéo từ phải qua trái (từ trên xuống dưới) để xác định xem có 5 ô liên tiếp trên một đường chéo không.

```

public int checkRightCross() {
    int win = 0, cheop = 0, n = 0;
    boolean check = false;
    List<JButton> list = new ArrayList<>();
    for (int i = size - 1; i >= 0; i--) {
        for (int j = 0; j < size; j++) {
            if (check) {
                if (check) {
                    if (n - j >= 0 && competitorMatrix[n - j][j] == 1) {
                        cheop++;
                        list.add(button[n - j][j]);
                        if (cheop > 4) {
                            for (JButton jButton : list) {
                                jButton.setDisabledIcon(new ImageIcon(winItem[numberOfMatch %
                                ]));
                            }
                            win = 1;
                            break;
                        }
                    }
                    continue;
                } else {
                    list = new ArrayList<>();
                    check = false;
                    cheop = 0;
                }
            }
            if (competitorMatrix[i][j] == 1) {
                n = i + j;
                check = true;
                list.add(button[i][j]);
                cheop++;
            } else {
                check = false;
                list = new ArrayList<>();
            }
        }
    }
    cheop = 0;
    cheop = 0;
    check = false;
    list = new ArrayList<>();
}
return win;
}

```

Hình 4.16. Kiểm tra chéo phải

- **Kiểm tra chéo trái:** Kiểm tra các đường chéo từ trái qua phải (từ trên xuống dưới) để xác định xem có 5 ô liên tiếp trên một đường chéo không.


```

public int checkLeftCross() {
    int win = 0, cheot = 0, n = 0;
    boolean check = false;
    List<JButton> list = new ArrayList<>();
    for (int i = 0; i < size; i++) {
        for (int j = size - 1; j >= 0; j--) {
            if (check) {
                if (n - j - 2 * cheot >= 0 && competitorMatrix[n - j - 2 * cheot][j] == 1) {
                    list.add(button[n - j - 2 * cheot][j]);
                    cheot++;
                    System.out.print("+" + j);
                    if (cheot > 4) {
                        for (JButton jButton : list) {
                            jButton.setDisabledIcon(new ImageIcon(winItem[numberOfMatch %
                                ]));
                        }
                        win = 1;
                        break;
                    }
                    continue;
                } else {
                    list = new ArrayList<>();
                    check = false;
                    cheot = 0;
                }
            }
            if (competitorMatrix[i][j] == 1) {
                list.add(button[i][j]);
                n = i + j;
                check = true;
                cheot++;
            } else {
                check = false;
            }
        }
    }

    list = new ArrayList<>();
    n = 0;
    cheot = 0;
    check = false;
}
return win;
}

```

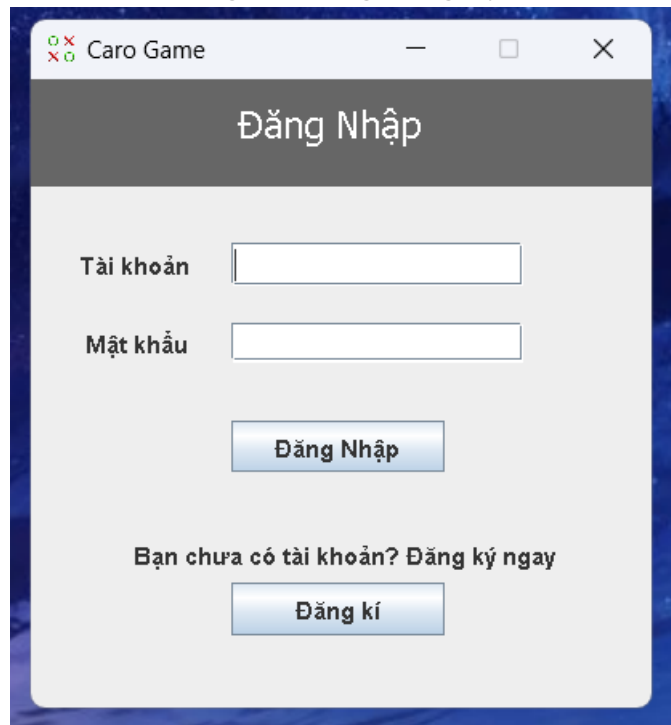
Hình 4.17. Kiểm tra chéo trái

Chương V: ĐÁNH GIÁ VÀ KẾT LUẬN

5.1 Kết quả thực thi

5.1.1 Giao diện trang đăng nhập

- Người dùng có thể nhập tên đăng nhập và mật khẩu để đăng nhập vào hệ thống.
- Có liên kết để người dùng đăng ký tài khoản mới.

The image shows a web browser window titled "Caro Game". Inside the window, there is a login form with a dark grey header containing the text "Đăng Nhập". Below the header, there are two input fields: "Tài khoản" (Username) and "Mật khẩu" (Password). Below these fields is a blue button labeled "Đăng Nhập". At the bottom of the form, there is a link that says "Bạn chưa có tài khoản? Đăng ký ngay" (Don't have an account? Register now) and a blue button labeled "Đăng kí".

Hình5.1. Giao diện đăng nhập

5.1.2 Giao diện đăng ký

- Người dùng nhập thông tin cá nhân như tên và mật khẩu để tạo tài khoản mới.
- Có các yêu cầu về độ mạnh của mật khẩu và xác nhận mật khẩu.



Đăng Kí

Tài Khoản

Mật Khẩu

NickName

Chọn Avatar 

[Đã có tài khoản, đăng nhập tại đây](#)

Hình 5.2. Giao diện đăng ký

5.1.3 Giao diện chính

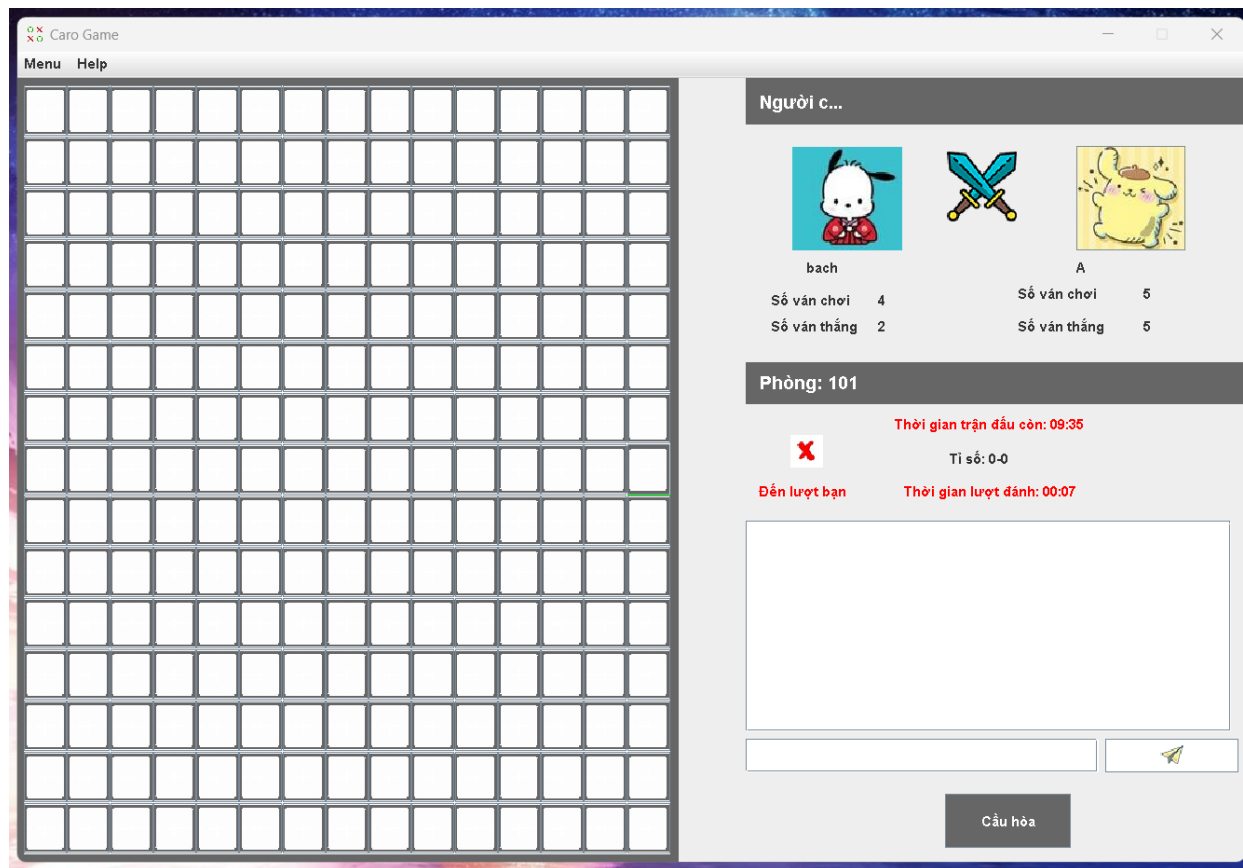
- Người dùng có thể bắt đầu ván chơi mới, xem lịch sử các ván chơi trước.
- Các thông tin cá nhân và trạng thái hiện tại của người dùng được hiển thị.



Hình5.3. Giao diện chính

5.1.4 Giao diện bàn cờ

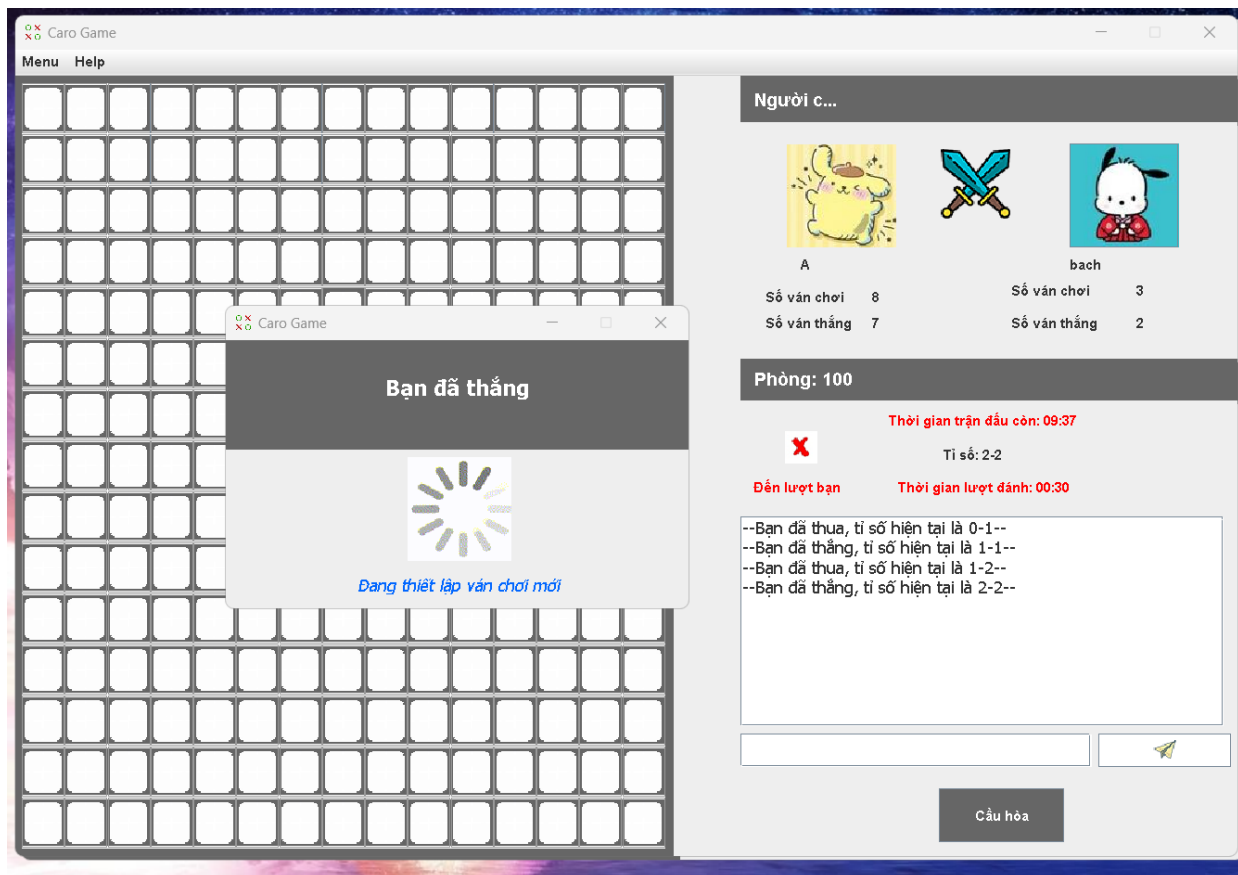
- Bàn cờ caro được hiển thị với các ô vuông, người chơi có thể nhấp vào các ô để đặt quân cờ.
- Hiển thị tên người chơi, điểm số và trạng thái của ván chơi.
- Có các nút chức năng như thoát, cầu hoà, nhắn tin và bắt đầu lại ván chơi mới.



Hình5.4. Giao diện bàn cờ

5.1.5 Giao diện kết thúc ván chơi

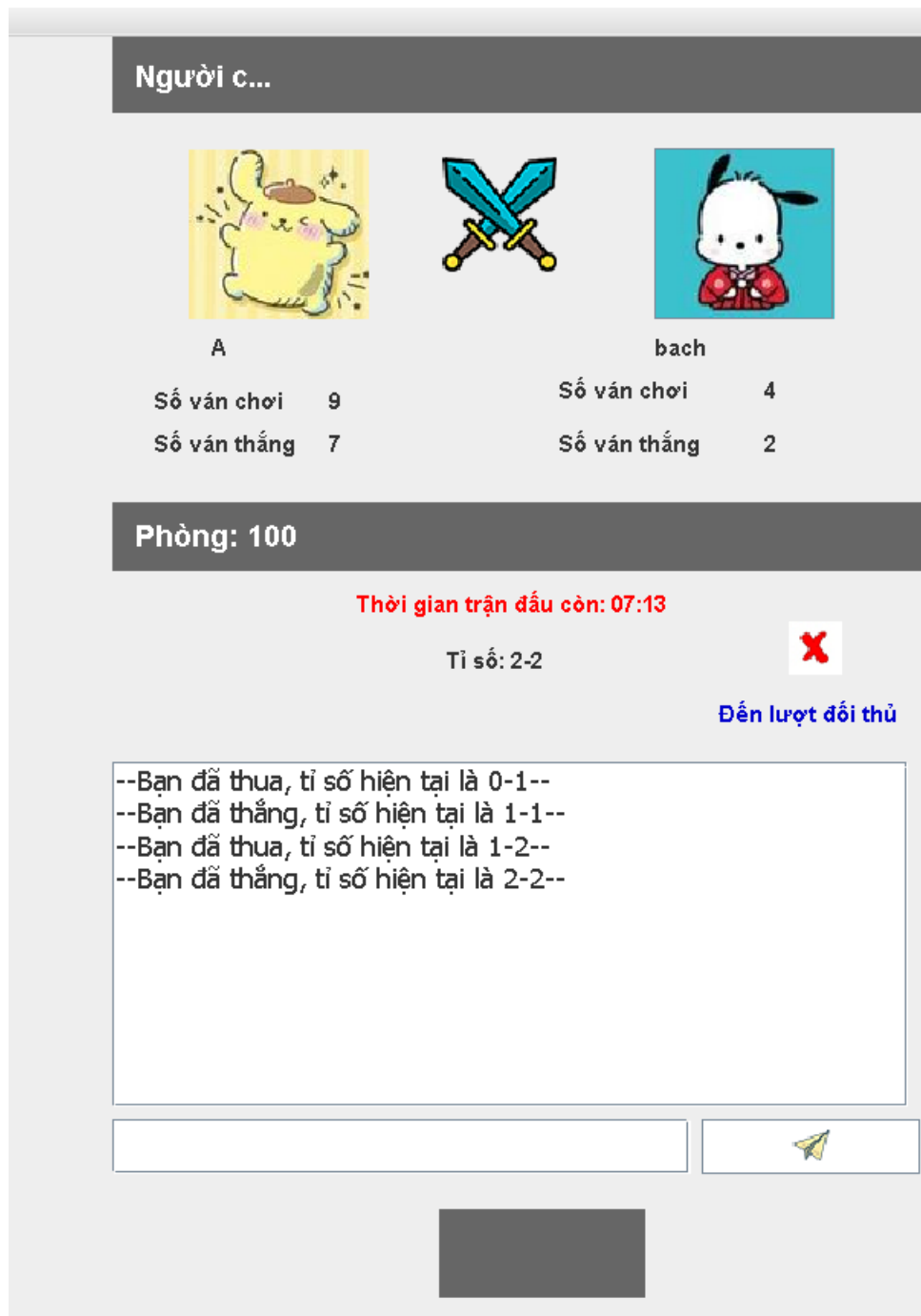
- Hiển thị kết quả của ván chơi.



Hình 5.5. Giao diện kết thúc ván chơi

5.1.6 Giao diện lịch sử chơi

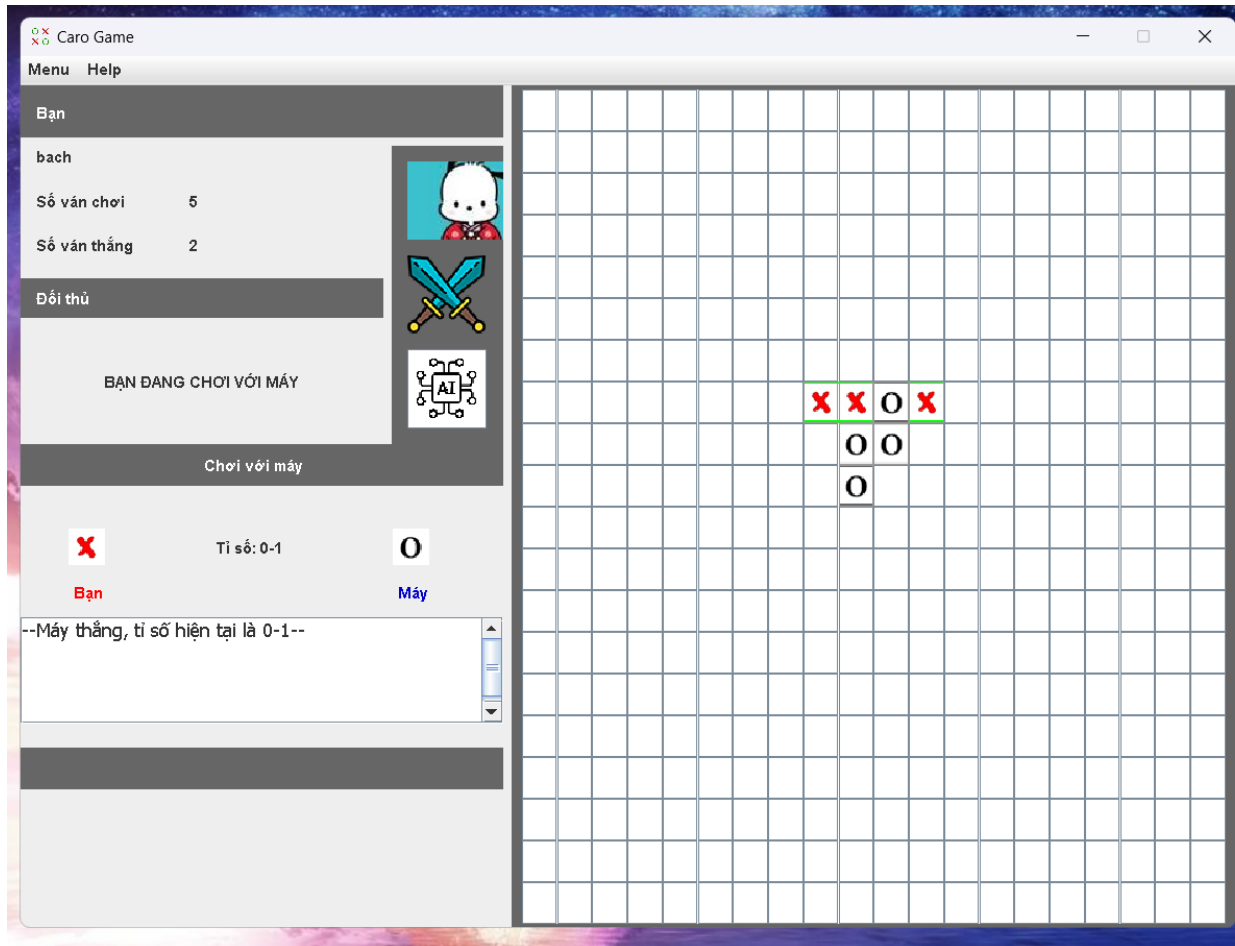
- Hiện thị danh sách các ván chơi đã hoàn thành và kết quả chi tiết.



Hình 5.6. Giao diện lịch sử chơi

5.1.7 Giao diện thách đấu máy

- Người dùng chọn chế độ chơi với máy.



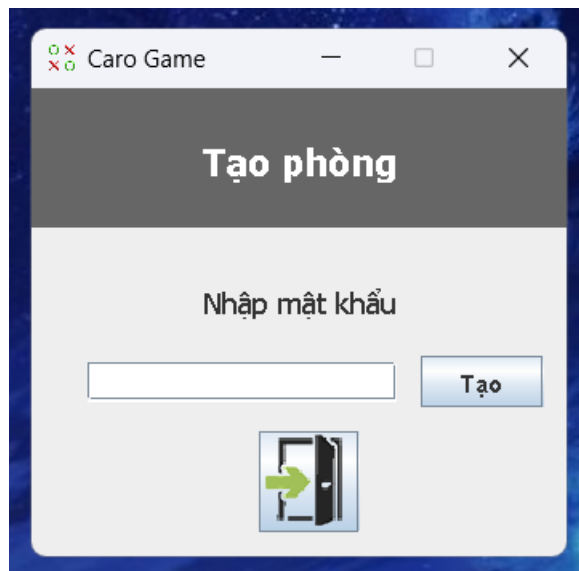
Hình5.7. Giao diện đấu máy

5.1.8 Giao diện vào phòng



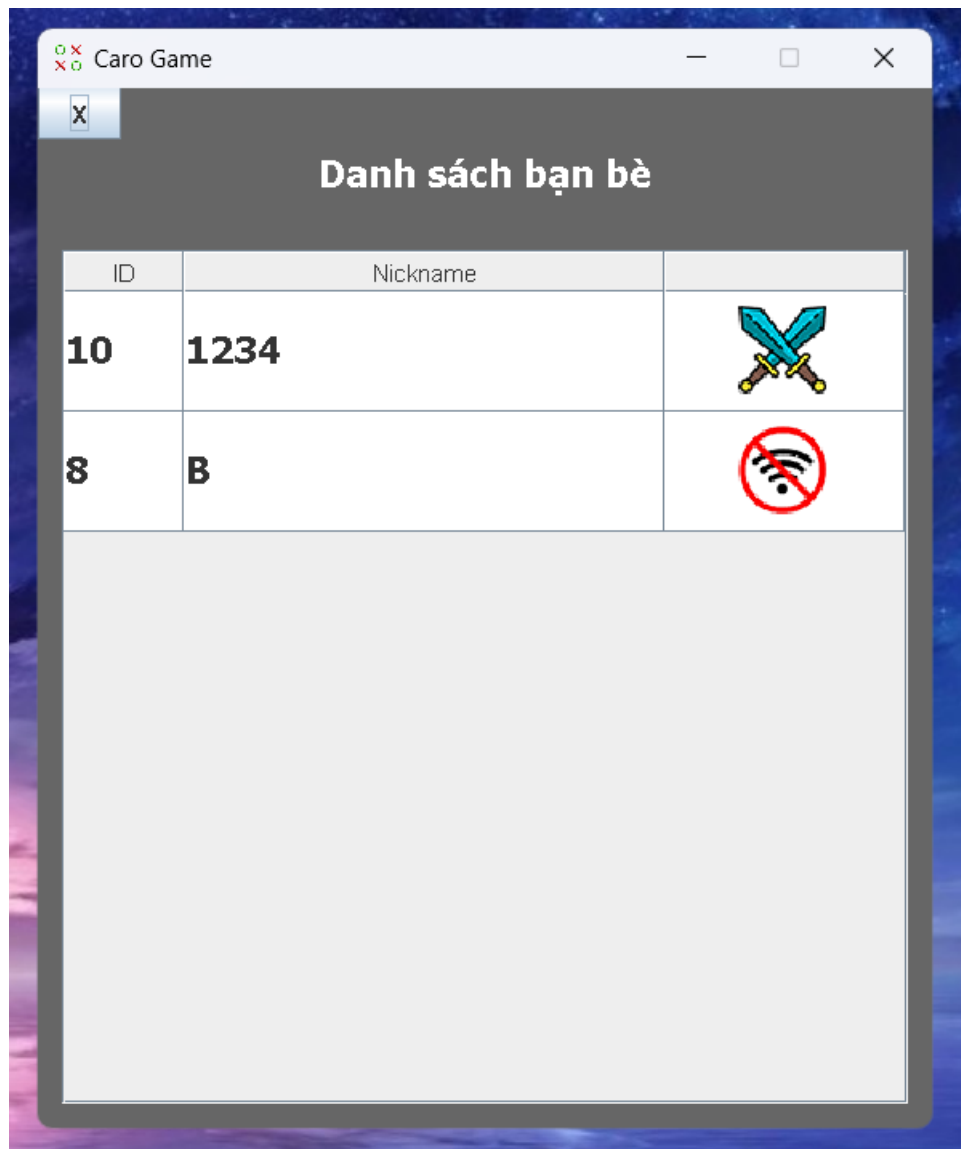
Hình5.8. Giao diện vào phòng

5.1.9 Giao diện tạo phòng



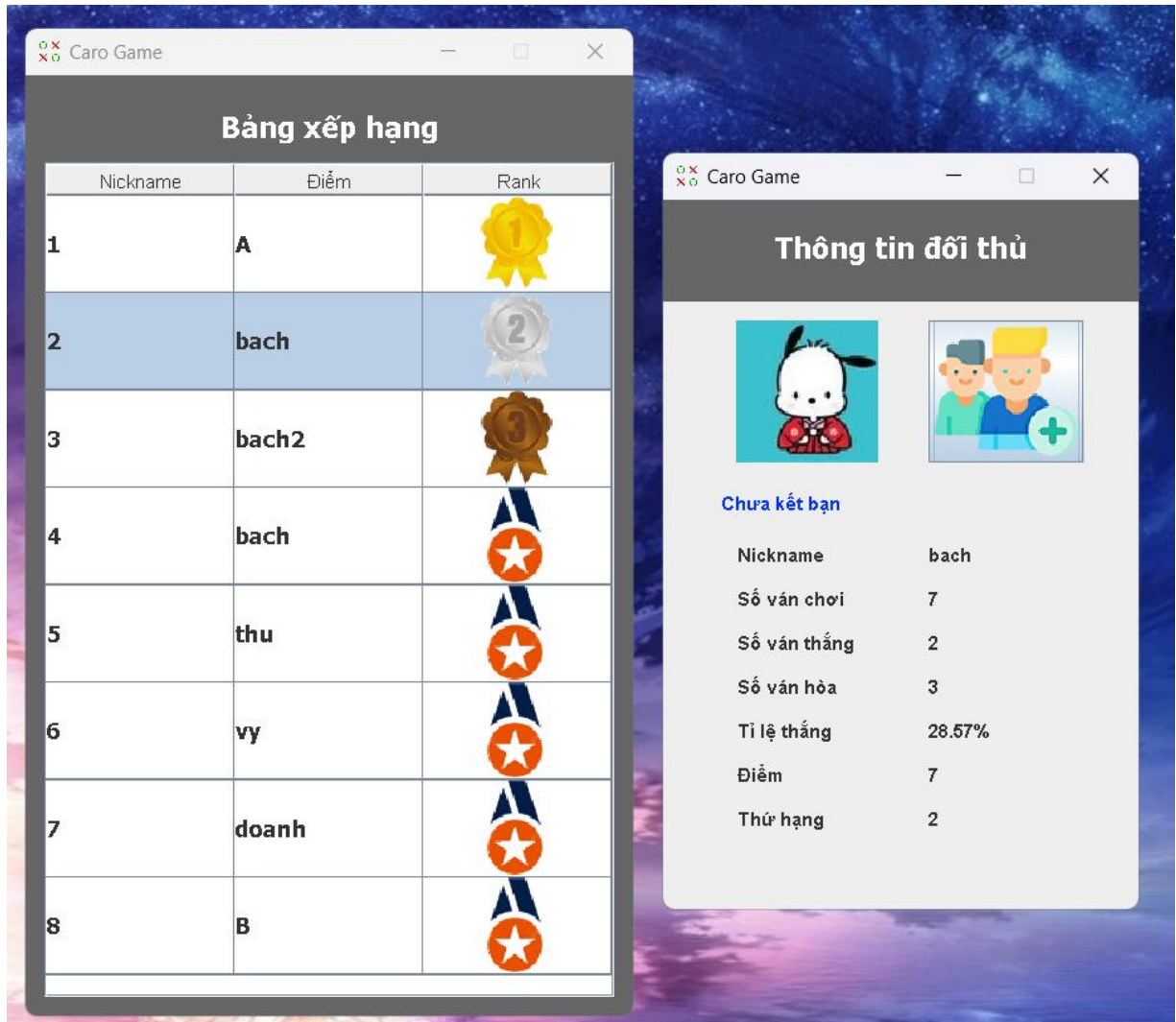
Hình 5.9. Giao diện tạo phòng

5.1.10 Giao diện danh sách bạn bè



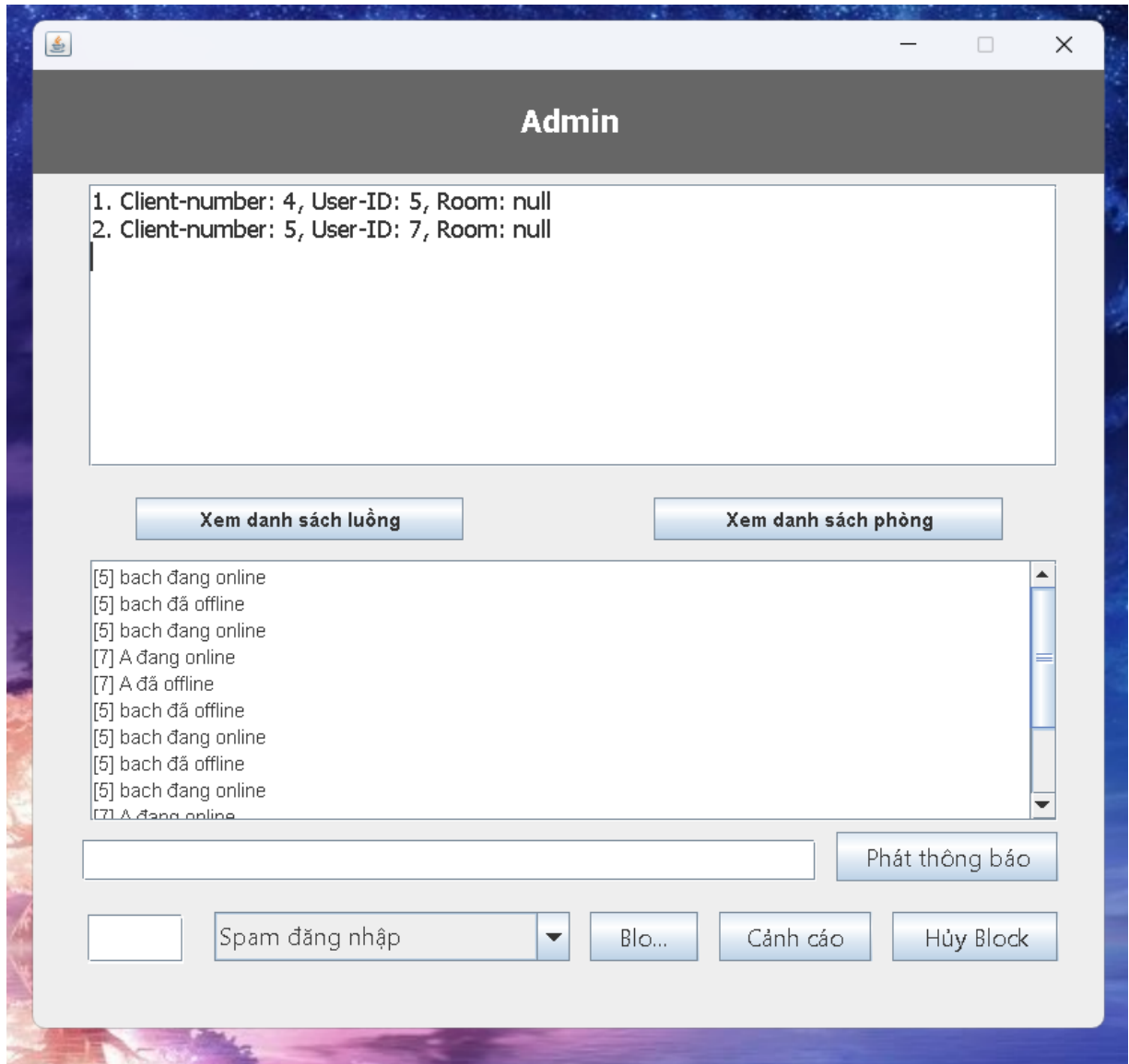
Hình 5.10. Giao diện danh sách bạn bè

5.1.11 Giao diện bảng xếp hạng và thông tin đối thủ



Hình 5.11. Giao diện bảng xếp hạng và thông tin đối thủ

5.1.12 Giao diện Admin



Hình5.12. Giao diện Admin

5.2 Đánh giá hệ thống

5.2.1 Hiệu năng

- **Tốc độ xử lý nhanh:** Hệ thống phản hồi nhanh, đảm bảo trải nghiệm người dùng mượt mà. Thời gian tính toán nước đi và phản hồi giao diện là tối ưu.
- **Thuật toán tối ưu:** Sử dụng thuật toán Minimax với cắt Alpha-Beta đã giúp hệ thống tìm kiếm nước đi tốt nhất trong thời gian ngắn.

- **Khả năng mở rộng tốt:** Hệ thống có khả năng xử lý nhiều người chơi cùng lúc mà không bị giảm hiệu lực.

5.2.2 Bảo mật

- **Bảo vệ dữ liệu người dùng:** Hệ thống bảo mật thông tin cá nhân và thông tin đăng nhập của người dùng.
- **Chống không lộ hiệu quả:** Đã có các biện pháp bảo vệ và ngăn chặn hành vi gian nan, như sử dụng bot.
- **Truy cập an toàn:** Hệ thống sử dụng kết nối HTTPS để bảo vệ việc truyền dữ liệu giữa máy khách và máy chủ.

5.2.3 Độ tin cậy

- **Định nghĩa hoạt động:** Định nghĩa hoạt động ổn định của hệ thống mà không gặp lỗi hay hoạt động đột ngột.
- **Cơ chế sao lưu và khôi phục:** Hệ thống có cơ chế sao lưu dữ liệu và khôi phục khi xảy ra sự cố, đảm bảo dữ liệu luôn an toàn.
- **Hỗ trợ kỹ thuật tốt:** Đội ngũ hỗ trợ kỹ thuật sẵn sàng giải quyết các vấn đề phát sinh phù hợp.

5.3 Kết luận

5.3.1 Hiệu năng

- **Tốc độ xử lý nhanh:** Hệ thống phản hồi nhanh, đảm bảo trải nghiệm người dùng mượt mà. Thời gian tính toán nước đi và phản hồi giao diện là tối ưu.
- **Thuật toán tối ưu:** Sử dụng thuật toán Minimax với cắt Alpha-Beta đã giúp hệ thống tìm kiếm nước đi tốt nhất trong thời gian ngắn.
- **Khả năng mở rộng tốt:** Hệ thống có khả năng xử lý nhiều người chơi cùng lúc mà không bị giảm hiệu lực.

5.3.2 Bảo mật

- **Bảo vệ dữ liệu người dùng:** Hệ thống bảo mật thông tin cá nhân và thông tin đăng nhập của người dùng.
- **Chống khổng lồ hiệu quả:** Đã có các biện pháp bảo vệ và ngăn chặn hành vi gian nan, như sử dụng bot.
- **Truy cập an toàn:** Hệ thống sử dụng kết nối HTTPS để bảo vệ việc truyền dữ liệu giữa máy khách và máy chủ.

5.3.4 Độ tin cậy

- **Định nghĩa hoạt động:** Định nghĩa hoạt động ổn định của hệ thống mà không gặp lỗi hay hoạt động đột ngột.
- **Cơ chế sao lưu và khôi phục:** Hệ thống có cơ chế sao lưu dữ liệu và khôi phục khi xảy ra sự cố, đảm bảo dữ liệu luôn an toàn.
- **Hỗ trợ kỹ thuật tốt:** Đội ngũ hỗ trợ kỹ thuật sẵn sàng giải quyết các vấn đề phát sinh phù hợp.

5.4 Hạn chế và hướng phát triển

5.4.1 Hạn chế của đề tài về cờ caro

- **Phạm vi nghiên cứu hạn chế**

Nhiều nghiên cứu tập trung vào các thuật toán cơ bản và chiến lược chơi đơn giản, không khám phá sâu hơn về các biến thể phức tạp hoặc các chiến thuật nâng cao.

- **Khả năng tính toán**

Đối với các trò chơi như cờ caro, việc tính toán tất cả các nước đi có thể trở nên phức tạp và tốn thời gian, đặc biệt khi kích thước bàn cờ tăng.

- **Giới hạn về dữ liệu**

Thiếu các cơ sở dữ liệu lớn về các trận đấu cờ caro để có thể phân tích và phát triển các chiến thuật và thuật toán hiệu quả hơn.

- **Phần cứng và phần mềm**

Yêu cầu về phần cứng mạnh mẽ để chạy các mô phỏng và thuật toán phức tạp.

Thiếu các phần mềm chuyên dụng được tối ưu hóa cho nghiên cứu và phát triển trong lĩnh vực cờ caro.

- **Khả năng áp dụng thực tế**

Khó khăn trong việc áp dụng các kết quả nghiên cứu từ mô phỏng máy tính vào thực tế chơi cờ do sự khác biệt giữa lý thuyết và thực tiễn.

5.4.2 Hướng phát triển trong tương lai

- **Cải tiến thuật toán**

Phát triển các thuật toán mới, tối ưu hóa các thuật toán hiện có để tăng hiệu suất và độ chính xác.

Áp dụng trí tuệ nhân tạo và học máy để tạo ra các mô hình chơi cờ caro thông minh hơn.

- **Phân tích dữ liệu lớn**

Xây dựng và khai thác các cơ sở dữ liệu lớn từ các trận đấu thực tế để phát triển các chiến lược chơi tối ưu.

Sử dụng phân tích dữ liệu lớn để xác định các mô hình và chiến thuật chơi hiệu quả.

- **Phát triển phần mềm chuyên dụng**

Tạo ra các công cụ và phần mềm chuyên dụng hỗ trợ việc nghiên cứu và phát triển trong lĩnh vực cờ caro.

Xây dựng các ứng dụng di động hoặc trang web cho phép người chơi thực hành và cải thiện kỹ năng của mình.

- **Hợp tác đa ngành**

Kết hợp với các lĩnh vực khác như tâm lý học, thần kinh học để hiểu rõ hơn về cách con người ra quyết định trong trò chơi.

Hợp tác với các chuyên gia từ các lĩnh vực khác để phát triển các chiến lược chơi mới và hiệu quả hơn.

- **Khả năng áp dụng trong giáo dục**

Sử dụng cờ caro như một công cụ giáo dục để phát triển tư duy logic, chiến lược và kỹ năng giải quyết vấn đề cho học sinh và sinh viên.

- **Tổ chức giải đấu và sự kiện**

Tổ chức các giải đấu cờ caro quốc tế để tạo sân chơi cho người chơi, đồng thời tạo cơ hội thu thập dữ liệu và phân tích chiến thuật.

- **Nghiên cứu giao diện người – máy**

Phát triển các giao diện người-máy tiên tiến để cải thiện trải nghiệm người dùng khi chơi cờ caro trên các thiết bị điện tử

Chương VI: TÀI LIỆU THAM KHẢO

1. Đề cương, các tài liệu về môn Lập trình mạng thuộc Khoa Công nghệ thông tin trường Đại học Ngoại Ngữ - Tin Học Tp.HCM.
2. <https://en.wikipedia.org/wiki/Minimax>
3. https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning
4. <https://techmaster.vn/posts/37756/huong-dan-tao-game-co-caro-bang-javascript>
5. <https://codelearn.io/sharing/lam-game-caro-don-gian-bang-java>
6. https://github.com/HoangTran0410/CaroOnline_SocketJava
7. <https://funix.edu.vn/chia-se-kien-thuc/lap-trinh-game-caro-bang-java/>
8. <https://123docz.net/document/626227-xay-dung-tro-choi-caro-online-tren-android.htm>
9. https://youtu.be/CAEI_J50B18?si=OLbibNloZYK9lfDe
10. <https://youtu.be/vLYu9upnMYQ?si=FHICuAWBLpyWneSX>