

TÀI LIỆU YÊU CẦU HỆ THỐNG (REQUIREMENTS DOCUMENT)

ỨNG DỤNG GAME BATTLESHIP TRỰC TUYẾN

Phiên bản: 1.0

Ngày: 29/10/2025

Ngôn ngữ lập trình: C/C++

Nền tảng: Linux/Ubuntu

1. GIỚI THIỆU

1.1 Mục đích tài liệu

Tài liệu này mô tả chi tiết các yêu cầu chức năng và phi chức năng cho hệ thống game Battleship (Hải chiến) trực tuyến theo lượt. Tài liệu được sử dụng làm cơ sở cho việc thiết kế, phát triển và kiểm thử hệ thống.

1.2 Phạm vi dự án

Xây dựng ứng dụng game Battleship trực tuyến cho phép hai người chơi đối đầu với nhau theo thời gian thực qua mạng. Hệ thống bao gồm:

- Server game quản lý kết nối, logic game và dữ liệu
- Client game với giao diện đồ họa để người chơi tương tác
- Hệ thống xếp hạng dựa trên điểm ELO
- Cơ chế lưu trữ và replay ván đấu

1.3 Mô tả game Battleship

Battleship là trò chơi chiến thuật 2 người chơi trên bàn cờ lưới 10x10:

- Mỗi người chơi bố trí một hạm đội gồm 5 tàu có kích thước khác nhau
- Người chơi lần lượt "bắn" vào ô trên bàn cờ đối phương
- Mục tiêu: Phá hủy toàn bộ hạm đội đối phương
- Người chơi thắng khi đánh chìm tất cả tàu của đối thủ

Danh sách tàu chuẩn:

- Aircraft Carrier (Tàu sân bay) - 5 ô
- Battleship (Tàu chiến) - 4 ô

- Cruiser (T||順chiến hạm) - 3 ô
 - Submarine (Tàu ngầm) - 3 ô
 - Destroyer (Tàu khu trục) - 2 ô
-

2. YÊU CẦU CHỨC NĂNG (FUNCTIONAL REQUIREMENTS)

2.1 Quản lý người dùng (User Management)

2.1.1 Đăng ký tài khoản (FR-001)

Mô tả: Cho phép người dùng mới tạo tài khoản

Yêu cầu chi tiết:

- Username: 4-20 ký tự, chỉ chứa chữ cái, số và dấu gạch dưới
- Password: Tối thiểu 6 ký tự
- Email: Định dạng hợp lệ (tùy chọn)
- Hiển thị tên (Display name): 2-30 ký tự
- Kiểm tra username không trùng lặp
- Mã hóa password trước khi lưu (SHA-256 hoặc bcrypt)
- Khởi tạo điểm ELO ban đầu: 1000

Input: Username, Password, Display name, Email (optional)

Output: Thông báo đăng ký thành công/thất bại với lý do

Điểm: 1 điểm (trong tổng 2 điểm quản lý tài khoản)

2.1.2 Đăng nhập (FR-002)

Mô tả: Xác thực người dùng và tạo phiên làm việc

Yêu cầu chi tiết:

- Xác thực username và password
- Kiểm tra tài khoản đã đăng nhập ở nơi khác (không cho đăng nhập đồng thời)
- Tạo session token duy nhất cho phiên làm việc
- Cập nhật trạng thái online
- Timeout phiên: 24 giờ hoặc khi đăng xuất

Input: Username, Password

Output: Session token và thông tin người dùng hoặc thông báo lỗi

Điểm: 2 điểm

2.1.3 Đăng xuất (FR-003)

Mô tả: Kết thúc phiên làm việc

Yêu cầu chi tiết:

- Xóa session token
- Cập nhật trạng thái offline
- Ngắt kết nối socket nếu có

Input: Session token

Output: Xác nhận đăng xuất

Điểm: Bao gồm trong FR-002

2.1.4 Quản lý profile (FR-004)

Mô tả: Xem và chỉnh sửa thông tin cá nhân

Yêu cầu chi tiết:

- Xem thông tin: Username, Display name, Email, ELO rating
- Thống kê: Tổng số trận, thắng, thua, hòa, tỷ lệ thắng
- Lịch sử xếp hạng cao nhất
- Cho phép đổi password
- Cho phép đổi display name và email

Input: Session token, thông tin cần cập nhật

Output: Thông tin profile hoặc xác nhận cập nhật

Điểm: 1 điểm (trong tổng 2 điểm quản lý tài khoản)

2.2 Danh sách người chơi và matchmaking

2.2.1 Hiển thị danh sách người chơi online (FR-005)

Mô tả: Hiển thị danh sách người chơi đang online và trạng thái của họ

Yêu cầu chi tiết:

- Danh sách người chơi online với các trạng thái:
 - Available (Sẵn sàng): Đang chờ trận đấu
 - In Game (Đang chơi): Đang trong ván đấu
 - Busy (Bận): Đang xem replay hoặc trong menu
- Hiển thị thông tin: Display name, ELO rating, trạng thái
- Sắp xếp theo ELO rating (mặc định) hoặc tên
- Tự động cập nhật khi có thay đổi (real-time update)
- Hiển thị chỉ báo khi có thách đấu đến

Input: Session token

Output: Danh sách người chơi với thông tin và trạng thái

Điểm: 2 điểm

2.2.2 Gửi lời thách đấu (FR-006)

Mô tả: Người chơi gửi lời mời thi đấu cho người khác

Yêu cầu chi tiết:

- Chỉ gửi được cho người chơi có trạng thái "Available"
- Có thể đính kèm cài đặt ván đấu:
 - Thời gian mỗi lượt (30s, 60s, 90s, unlimited)
 - Chế độ bố trí tàu (manual/random)
- Gửi thông báo real-time đến người được thách đấu
- Timeout: Lời thách đấu tự động hủy sau 60 giây nếu không có phản hồi
- Một người chỉ gửi được 1 lời thách đấu tại một thời điểm

Input: Session token, Target player ID, Game settings

Output: Challenge ID và trạng thái gửi

Điểm: 2 điểm

2.2.3 Nhận/Tùy chối lời thách đấu (FR-007)

Mô tả: Người chơi phản hồi lời thách đấu

Yêu cầu chi tiết:

- Hiển thị popup/notification khi nhận thách đấu
- Hiển thị thông tin người thách đấu và cài đặt trận
- Nút Accept/Decline
- Khi Accept: Chuyển sang màn hình bố trí tàu
- Khi Decline: Gửi thông báo từ chối cho người gửi
- Tự động từ chối nếu timeout

Input: Session token, Challenge ID, Response (Accept/Decline)

Output: Match ID (nếu accept) hoặc xác nhận từ chối

Điểm: 1 điểm

2.2.4 Matchmaking tự động (FR-008) - Chức năng nâng cao

Mô tả: Tự động ghép cặp người chơi có trình độ tương đương

Yêu cầu chi tiết:

- Nút "Quick Match" để tìm đối thủ
- Ghép cặp dựa trên ELO rating (chênh lệch < 200 ELO)
- Ưu tiên ghép người chơi có thời gian chờ lâu nhất
- Mở rộng phạm vi tìm kiếm sau mỗi 30 giây (+100 ELO)
- Hiển thị trạng thái đang tìm kiếm với animation
- Cho phép hủy tìm kiếm
- Thông báo khi tìm thấy đối thủ

Input: Session token, Preferred settings

Output: Match ID khi tìm thấy đối thủ

Điểm: 3 điểm (chức năng nâng cao)

2.3 Gameplay - Bố trí tàu

2.3.1 Bố trí tàu thủ công (FR-009)

Mô tả: Người chơi tự đặt vị trí các tàu trên bàn cờ

Yêu cầu chi tiết:

- Bàn cờ 10x10 với tọa độ A-J (hàng) và 1-10 (cột)
- Hiển thị danh sách 5 tàu cần đặt với kích thước
- Drag-and-drop hoặc click để đặt tàu
- Cho phép xoay tàu (ngang/dọc) bằng phím R hoặc nút
- Kiểm tra vị trí hợp lệ:
 - Tàu không được chồng lên nhau
 - Tàu không được vượt ra ngoài bàn cờ
 - Có thể cho phép hoặc không cho phép tàu kề nhau (tùy chọn)
- Nút "Random" để bố trí ngẫu nhiên
- Nút "Reset" để xóa và đặt lại
- Nút "Ready" khi hoàn thành bố trí
- Countdown timer: 90 giây để bố trí (nếu hết thời gian thì random)

Input: Ship ID, Position, Orientation

Output: Xác nhận vị trí hợp lệ/không hợp lệ

Điểm: 2 điểm (bao gồm trong giao diện đồ họa)

2.3.2 Bố trí tàu ngẫu nhiên (FR-010) - Chức năng nâng cao

Mô tả: Tự động bố trí ngẫu nhiên các tàu hợp lệ

Yêu cầu chi tiết:

- Thuật toán đặt ngẫu nhiên đảm bảo không trùng lặp
- Phân bố đều trên bàn cờ
- Tùy chọn: Cho phép tàu kề nhau hoặc giãn cách tối thiểu 1 ô

Input: Session token, Match ID

Output: Board configuration

Điểm: 1 điểm (chức năng nâng cao)

2.4 Gameplay - Lượt chơi

2.4.1 Xác định lượt chơi (FR-011)

Mô tả: Xác định người chơi đi trước

Yêu cầu chi tiết:

- Ngẫu nhiên chọn người chơi đi trước
- Hoặc người có ELO thấp hơn đi trước
- Thông báo rõ ràng cho cả 2 người chơi

Input: Match ID

Output: Player turn

Điểm: Bao gồm trong logic game

2.4.2 Thực hiện nước đi (FR-012)

Mô tả: Người chơi chọn ô để bắn

Yêu cầu chi tiết:

- Chỉ người chơi có lượt mới được chọn
- Click vào ô trên bàn cờ đối phương để bắn
- Hiển thị countdown timer cho lượt (nếu có giới hạn thời gian)
- Xác nhận nước đi trước khi gửi (optional)
- Gửi tọa độ đến server

Input: Session token, Match ID, Target coordinate (row, col)

Output: Shot result (Miss/Hit/Sunk)

Điểm: 1 điểm (trong tổng 2 điểm chuyển thông tin nước đi)

2.4.3 Kiểm tra tính hợp lệ nước đi (FR-013)

Mô tả: Server kiểm tra nước đi có hợp lệ không

Yêu cầu chi tiết:

- Kiểm tra đúng lượt của người chơi
- Kiểm tra tọa độ trong phạm vi (0-9, 0-9)
- Kiểm tra ô chưa được bắn trước đó
- Kiểm tra không timeout (nếu có giới hạn thời gian)
- Từ chối nước đi không hợp lệ với thông báo lỗi cụ thể

Input: Match state, Player ID, Coordinate

Output: Valid/Invalid với lý do

Điểm: 1 điểm

2.4.4 Xử lý kết quả nước đi (FR-014)

Mô tả: Xác định kết quả của nước bắn

Yêu cầu chi tiết:

- Kiểm tra tọa độ có trùng với vị trí tàu không:
 - **Miss:** Không trúng tàu nào
 - **Hit:** Trúng một phần của tàu
 - **Sunk:** Trúng và đánh chìm hoàn toàn một tàu
- Cập nhật trạng thái bàn cờ
- Ghi nhận vào lịch sử nước đi
- Gửi kết quả cho cả 2 người chơi với thông tin chi tiết:
 - Tọa độ

- Kết quả (Miss/Hit/Sunk)
- Tên tàu bị chìm (nếu Sunk)
- Số lượng tàu còn lại
- Hiệu ứng animation tương ứng trên UI

Input: Board state, Coordinate

Output: Shot result với details

Điểm: 1 điểm (trong tổng 2 điểm chuyển thông tin nước đi)

2.4.5 Chuyển lượt (FR-015)

Mô tả: Chuyển quyền điều khiển sang người chơi khác

Yêu cầu chi tiết:

- Sau mỗi nước đi Miss: Chuyển lượt cho đối thủ
- Sau mỗi nước Hit/Sunk: Người chơi hiện tại tiếp tục (luật tùy chọn, có thể thay đổi)
- Cập nhật UI cho cả 2 người chơi
- Reset countdown timer
- Disable/Enable bàn cờ tương ứng

Input: Match state

Output: Current player turn

Điểm: Bao gồm trong logic game

2.5 Kết thúc trận đấu

2.5.1 Xác định kết quả (FR-016)

Mô tả: Kiểm tra điều kiện thắng/thua/hòa

Yêu cầu chi tiết:

- **Thắng:** Khi đánh chìm tất cả 5 tàu của đối phương
- **Thua:** Khi tắt cả tàu của mình bị chìm
- **Hòa:**
 - Cả 2 người chơi đồng ý hòa
 - Timeout không ai thực hiện nước đi trong X phút
 - Cả 2 người chơi disconnect (xử lý đặc biệt)
- Hiển thị màn hình kết quả với:
 - Người thắng/thua/hòa
 - Bàn cờ cuối cùng của cả 2 bên
 - Thông kê: Số nước đi, độ chính xác, thời gian chơi
 - Điểm ELO thay đổi

Input: Match state

Output: Game result (Win/Lose/Draw) với statistics

Điểm: 1 điểm

2.5.2 Tính điểm ELO (FR-017)

Mô tả: Cập nhật ELO rating sau trận đấu

Yêu cầu chi tiết:

- Sử dụng công thức ELO chuẩn:
 - Expected Score: $E_A = 1 / (1 + 10^{((R_B - R_A)/400)})$
 - New Rating: $R'_A = R_A + K * (S_A - E_A)$
 - K-factor: 32 cho người chơi mới (<30 trận), 24 cho người chơi bình thường, 16 cho người chơi cao thủ (>2400 ELO)
- Tính cho cả thắng, thua và hòa:
 - Thắng: $S = 1$
 - Hòa: $S = 0.5$
 - Thua: $S = 0$
- Hiển thị thay đổi ELO rõ ràng (+25, -18, etc.)
- Không cho phép ELO < 0 (tối thiểu = 0)
- Lưu lịch sử thay đổi ELO

Input: Player A ELO, Player B ELO, Match result

Output: New ELO for both players

Điểm: 2 điểm

2.5.3 Lưu kết quả trận đấu (FR-018)

Mô tả: Ghi nhận thông tin trận đấu vào database

Yêu cầu chi tiết:

- Thông tin cần lưu:
 - Match ID (unique)
 - Player 1 ID, username, ELO trước/sau
 - Player 2 ID, username, ELO trước/sau
 - Winner/Loser/Draw
 - Ngày giờ bắt đầu và kết thúc
 - Tổng thời gian chơi
 - Số lượng nước đi của mỗi người
 - Settings (time limit, etc.)
- Cập nhật thống kê người chơi:
 - Total games
 - Wins/Losses/Draws
 - Win rate
 - Highest ELO
 - Current win/loss streak

Input: Match data

Output: Confirmation

Điểm: 1 điểm (trong tổng 2 điểm lưu thông tin)

2.6 Lưu trữ và Replay

2.6.1 Ghi log nước đi (FR-019)

Mô tả: Lưu trữ từng nước đi trong trận

Yêu cầu chi tiết:

- Format log cho mỗi nước đi:
 - Move number
 - Timestamp
 - Player ID
 - Coordinate (row, col)
 - Result (Miss/Hit/Sunk)
 - Ship name (nếu Sunk)
 - Time taken for the move
- Lưu theo format JSON hoặc binary để dễ parse
- Nén log nếu cần thiết

Input: Match ID, Move data

Output: Confirmation

Điểm: 1 điểm

2.6.2 Lưu trữ trạng thái ván đấu (FR-020)

Mô tả: Lưu toàn bộ thông tin để có thể replay

Yêu cầu chi tiết:

- Lưu trạng thái ban đầu:
 - Vị trí tất cả tàu của cả 2 người chơi
 - Người đi trước
 - Settings
- Lưu danh sách tất cả các nước đi (moves log)
- Lưu trạng thái cuối cùng và kết quả
- File/database entry cho mỗi match

Input: Match ID, Full match data

Output: Confirmation và storage location

Điểm: 1 điểm (trong tổng 2 điểm lưu thông tin)

2.6.3 Danh sách lịch sử trận đấu (FR-021)

Mô tả: Xem danh sách các trận đã chơi

Yêu cầu chi tiết:

- Hiển thị danh sách trận đấu với thông tin:
 - Ngày giờ
 - Đối thủ
 - Kết quả (W/L/D)
 - ELO thay đổi

- Duration
- Filter theo:
 - Thời gian (hôm nay, tuần này, tháng này, tất cả)
 - Kết quả (tất cả/thắng/thua/hòa)
 - Đối thủ cụ thể
- Sắp xếp theo ngày (mới nhất trước)
- Phân trang (10-20 trận/trang)
- Click để xem chi tiết hoặc replay

Input: Session token, Filter options

Output: List of matches

Điểm: Bao gồm trong chức năng xem thống kê

2.6.4 Replay trận đấu (FR-022)

Mô tả: Xem lại trận đấu đã chơi

Yêu cầu chi tiết:

- Load dữ liệu trận đấu từ storage
- Hiển thị 2 bàn cờ (của cả 2 người chơi)
- Điều khiển:
 - Play/Pause
 - Next move / Previous move
 - Speed control (0.5x, 1x, 2x, 4x)
 - Jump to move (slider hoặc input)
 - Jump to beginning/end
- Hiển thị thông tin nước đi hiện tại:
 - Move number
 - Player
 - Coordinate
 - Result
 - Time taken
- Highlight nước đi đang xem
- Hiển thị thống kê tổng quan bên cạnh

Input: Match ID

Output: Interactive replay interface

Điểm: 2 điểm

2.7 Tương tác trong trận đấu

2.7.1 Chat trong trận (FR-023) - Chức năng nâng cao

Mô tả: Gửi tin nhắn cho đối thủ trong khi chơi

Yêu cầu chi tiết:

- Chat box đơn giản

- Giới hạn độ dài tin nhắn (200 ký tự)
- Filter từ ngữ không phù hợp (optional)
- Hiển thị timestamp và tên người gửi
- Lưu lại chat log (optional)

Input: Session token, Match ID, Message

Output: Confirmation

Điểm: 2 điểm (chức năng nâng cao)

2.7.2 Xin tạm dừng (FR-024)

Mô tả: Tạm dừng trận đấu

Yêu cầu chi tiết:

- Nút "Pause" trong game
- Gửi yêu cầu đến đối thủ
- Đối thủ có thể Accept/Decline
- Nếu Accept: Dừng countdown timer, freeze game state
- Timeout tự động resume sau X phút (5-10 phút)
- Số lần pause giới hạn (mỗi người 2 lần/trận)

Input: Session token, Match ID

Output: Pause state

Điểm: 0.5 điểm (trong tổng 1 điểm xin ngừng/mời hòa)

2.7.3 Xin hòa (FR-025)

Mô tả: Đề nghị kết thúc trận với kết quả hòa

Yêu cầu chi tiết:

- Nút "Offer Draw" trong game
- Gửi yêu cầu đến đối thủ với popup
- Đối thủ Accept/Decline
- Nếu Accept: Kết thúc trận với kết quả hòa, tính ELO hòa
- Nếu Decline: Tiếp tục trận đấu
- Timeout tự động Decline sau 30 giây

Input: Session token, Match ID

Output: Draw offer response

Điểm: 0.5 điểm (trong tổng 1 điểm xin ngừng/mời hòa)

2.7.4 Chịu thua (FR-026)

Mô tả: Đầu hàng và kết thúc trận đấu

Yêu cầu chi tiết:

- Nút "Resign" trong game

- Hiển thị confirmation dialog
- Kết thúc ngay lập tức, không cần sự đồng ý của đối thủ
- Ghi nhận là thua, trừ ELO tương ứng
- Thông báo đối thủ

Input: Session token, Match ID

Output: Match end with resignation

Điểm: Bao gồm trong FR-025

2.7.5 Đề nghị ván đấu mới (FR-027)

Mô tả: Mời đối thủ vừa chơi đấu lại

Yêu cầu chi tiết:

- Nút "Rematch" sau khi kết thúc trận
- Gửi lời mời tự động với cùng settings
- Đối thủ Accept/Decline
- Nếu Accept: Tạo match mới và chuyển sang màn hình bố trí tàu
- Timeout sau 60 giây

Input: Session token, Previous match ID

Output: New match ID hoặc decline notification

Điểm: 1 điểm

2.8 Xếp hạng và Thống kê

2.8.1 Bảng xếp hạng (FR-028) - Chức năng nâng cao

Mô tả: Hiển thị top người chơi theo ELO

Yêu cầu chi tiết:

- Bảng xếp hạng toàn server
- Hiển thị top 100 người chơi
- Thông tin: Rank, Username, ELO, Win rate, Total games
- Highlight vị trí của người dùng hiện tại
- Filter theo:
 - Tất cả thời gian
 - Tháng này
 - Tuần này
- Tự động cập nhật sau mỗi trận đấu

Input: Session token, Filter options

Output: Leaderboard data

Điểm: 2 điểm (chức năng nâng cao)

2.8.2 Thống kê cá nhân (FR-029)

Mô tả: Xem thống kê chi tiết cá nhân

Yêu cầu chi tiết:

- Overall stats:
 - Total games played
 - Win/Loss/Draw count và rate
 - Current ELO và highest ELO
 - Current streak (win/loss)
 - Longest win streak
 - Average game duration
- Performance stats:
 - Accuracy (% shots hit)
 - Average shots to win
 - Favorite target areas (heatmap - nâng cao)
 - Most destroyed ship type
- Time-based stats:
 - Games per day/week/month
 - Win rate trend (graph - nâng cao)
 - ELO history (graph - nâng cao)

Input: Session token

Output: Statistics data

Điểm: Bao gồm trong các chức năng khác

2.8.3 So sánh với người chơi khác (FR-030) - Chức năng nâng cao

Mô tả: So sánh thống kê với người chơi cụ thể

Yêu cầu chi tiết:

- Chọn người chơi để so sánh
- Hiển thị stats song song
- Head-to-head record (nếu đã từng đấu)
- ELO comparison
- Win rate comparison

Input: Session token, Target player ID

Output: Comparison data

Điểm: 2 điểm (chức năng nâng cao)

2.9 Chức năng nâng cao khác

2.9.1 Custom board setup (FR-033) - Nâng cao

Mô tả: Tùy chỉnh cấu hình bàn cờ

Yêu cầu chi tiết:

- Thay đổi kích thước bàn cờ (8x8, 10x10, 12x12)
- Thay đổi số lượng và kích thước tàu

- Lưu và load presets
- Áp dụng cho custom matches

Input: Board configuration

Output: Confirmation

Điểm: 2-3 điểm

2.9.2 Game variants (FR-034) - Nâng cao

Mô tả: Các biến thể của Battleship

Yêu cầu chi tiết:

- **Salvo mode:** Mỗi lượt bắn nhiều phát (bằng số tàu còn lại)
- **Fog of war:** Không hiển thị Miss/Hit trước đó

Input: Game variant type

Output: Modified game rules

Điểm: 3-5 điểm mỗi variant

3. YÊU CẦU PHI CHỨC NĂNG (NON-FUNCTIONAL REQUIREMENTS)

3.1 Hiệu năng (Performance)

NFR-001: Response Time

- Đăng nhập/đăng ký: < 1 giây
- Load danh sách người chơi: < 500ms
- Gửi/nhận nước đi: < 200ms (latency thấp)
- Load replay: < 2 giây

NFR-002: Throughput

- Server hỗ trợ tối thiểu 100 người chơi đồng thời
- Xử lý tối thiểu 50 trận đấu đồng thời
- Hỗ trợ 1000+ queries/second

NFR-003: Scalability

- Kiến trúc cho phép scale theo chiều ngang (thêm server)
- Database design cho phép millions of records

3.2 Bảo mật (Security)

NFR-004: Authentication & Authorization

- Session-based authentication với token
- Token expiration: 24 giờ
- Password hashing: SHA-256 hoặc bcrypt với salt
- Không lưu plaintext password

NFR-005: Data Protection

- Truyền dữ liệu mã hóa (khuyến nghị TLS/SSL nếu có thể)
- Validate tất cả input từ client
- Prevent SQL injection (nếu dùng SQL)
- Chống cheating: Validate moves server-side

NFR-006: Privacy

- Không chia sẻ thông tin cá nhân ngoài username và stats
- Email không hiển thị công khai

3.3 Độ tin cậy (Reliability)

NFR-007: Availability

- Uptime mục tiêu: 95%+
- Xử lý graceful shutdown

NFR-008: Error Handling

- Xử lý timeout kết nối
- Xử lý disconnect giữa trận (auto-resign sau X giây)
- Recovery từ crash (save game state)
- Logging đầy đủ để debug

NFR-009: Data Integrity

- ACID compliance cho database transactions (nếu dùng SQL)
- Backup định kỳ database
- Không mất dữ liệu ELO và match history

3.4 Khả năng sử dụng (Usability)

NFR-010: User Interface

- Giao diện đồ họa với GTK hoặc SDL/OpenGL
- Responsive, smooth animation
- Clear visual feedback cho mọi hành động
- Intuitive controls

NFR-011: User Experience

- Onboarding flow rõ ràng cho người dùng mới
- Tutorial mode (optional)

- Helpful error messages
- Confirmation dialogs cho các hành động quan trọng

NFR-012: Accessibility

- Font size hợp lý, dễ đọc
- Color scheme không gây khó chịu
- Keyboard shortcuts (optional)

3.5 Khả năng bảo trì (Maintainability)

NFR-013: Code Quality

- Code structure rõ ràng, module hóa
- Comments đầy đủ
- Coding conventions nhất quán
- Version control (Git)

NFR-014: Documentation

- README với hướng dẫn compile và run
- API documentation
- Database schema documentation
- This requirements document

NFR-015: Testing

- Unit tests cho core logic
- Integration tests cho client-server communication
- Test cases cho các scenario quan trọng

3.6 Khả năng tương thích (Compatibility)

NFR-016: Platform

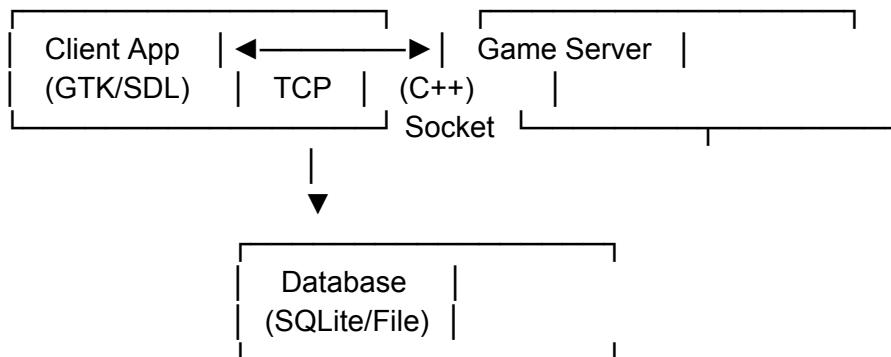
- Hỗ trợ Linux/Ubuntu (bắt buộc)
- Compile với GCC/G++
- Không phụ thuộc Windows-specific libraries

NFR-017: Standards

- Tuân thủ C++11/C++14/C++17 standards
- POSIX-compliant socket programming
- Standard network protocols (TCP/IP)

4. KIẾN TRÚC HỆ THỐNG

4.1 Tổng quan kiến trúc



4.2 Client Architecture

Modules:

1. **Network Module:** Quản lý socket connection, gửi/nhận messages
2. **UI Module:** Rendering, event handling (GTK hoặc SDL)
3. **Game Logic Module:** Validate local actions, update game state
4. **Audio Module (optional):** Sound effects
5. **Settings Module:** Lưu user preferences

Threads:

- Main thread: UI rendering và event loop
- Network thread: Lắng nghe messages từ server

4.3 Server Architecture

Modules:

1. **Network Module:**
 - Socket management (đa luồng hoặc epoll/select)
 - Cài đặt cơ chế vào/ra socket: 2 điểm
 - Message parsing và routing
 - Xử lý truyền dòng: 1 điểm
2. **Authentication Module:**
 - Đăng ký và quản lý tài khoản: 2 điểm
 - Đăng nhập và quản lý phiên: 2 điểm
 - Session management
3. **Matchmaking Module:**
 - Cung cấp danh sách người chơi: 2 điểm
 - Chuyển lời thách đấu: 2 điểm
 - Từ chối/Nhận lời thách đấu: 1 điểm

- Auto-matching logic

4. Game Engine:

- Chuyển thông tin nước đi: 2 điểm
- Kiểm tra tính hợp lệ nước đi: 1 điểm
- Xác định kết quả ván cờ: 1 điểm
- Ghi log: 1 điểm
- Chuyển kết quả và log ván cờ: 2 điểm
- Game state management
- Rules enforcement
- Win/draw detection

5. ELO Rating Module:

- Xây dựng hệ thống tính điểm: 2 điểm
- Calculate rating changes
- Update rankings

6. Database Module:

- Lưu thông tin ván đấu và replay: 2 điểm
- User data persistence
- Match history storage
- Statistics aggregation

7. Additional Features:

- Xin ngừng ván cờ/mời hòa: 1 điểm
- Xin đấu lại người chơi: 1 điểm
- Chức năng nâng cao: 2-10 điểm

Threads/Processes:

- Listener thread: Accept connections
- Worker threads/processes: Handle client requests
- Game threads: Manage active matches
- Database thread: Async DB operations

4.4 Communication Protocol

4.4.1 Message Format

Sử dụng format JSON hoặc custom binary protocol:

JSON Example:

```
{  
  "type": "MOVE",  
  "token": "abc123...",  
  "matchId": "m12345",  
  "data": {  
    "row": 5,  
    "col": 7  
  }  
}
```

```
    "col": 7
  }
}
```

Binary Protocol:

- Hiệu quả hơn nhưng phức tạp hơn
- Header: Message Type (1 byte), Length (4 bytes)
- Payload: Serialized data

4.4.2 Message Types

Client → Server:

- **AUTH_REGISTER**: Đăng ký
- **AUTH_LOGIN**: Đăng nhập
- **AUTH_LOGOUT**: Đăng xuất
- **PLAYER_LIST_REQUEST**: Yêu cầu danh sách người chơi
- **CHALLENGE_SEND**: Gửi lời thách đấu
- **CHALLENGE_RESPONSE**: Phản hồi thách đấu
- **MATCH_READY**: Đã sẵn sàng (sau khi bố trí tàu)
- **MOVE**: Thực hiện nước đi
- **PAUSE_REQUEST**: Xin tạm dừng
- **DRAW_OFFER**: Xin hòa
- **RESIGN**: Chịu thua
- **REMATCH_REQUEST**: Xin đấu lại
- **REPLAY_REQUEST**: Xem replay
- **CHAT_MESSAGE**: Gửi chat

Server → Client:

- **AUTH_RESPONSE**: Kết quả đăng nhập/dăng ký
- **PLAYER_LIST**: Danh sách người chơi
- **CHALLENGE RECEIVED**: Nhận thách đấu
- **MATCH_START**: Bắt đầu trận đấu
- **MATCH_STATE**: Cập nhật trạng thái trận
- **TURN_UPDATE**: Thông báo lượt chơi
- **MOVE_RESULT**: Kết quả nước đi
- **MATCH_END**: Kết thúc trận với kết quả
- **ELO_UPDATE**: Cập nhật điểm ELO
- **ERROR**: Thông báo lỗi
- **NOTIFICATION**: Thông báo chung

4.5 Database Schema

Users Table

users

└── user_id (PK, INTEGER, AUTO_INCREMENT)
└── username (UNIQUE, VARCHAR(20))
└── password_hash (VARCHAR(64))
└── display_name (VARCHAR(30))
└── email (VARCHAR(100), NULLABLE)
└── elo_rating (INTEGER, DEFAULT 1000)
└── highest_elo (INTEGER, DEFAULT 1000)
└── total_games (INTEGER, DEFAULT 0)
└── wins (INTEGER, DEFAULT 0)
└── losses (INTEGER, DEFAULT 0)
└── draws (INTEGER, DEFAULT 0)
└── current_streak (INTEGER, DEFAULT 0)
└── longest_win_streak (INTEGER, DEFAULT 0)
└── created_at (TIMESTAMP)
└── last_login (TIMESTAMP)

Matches Table

matches

└── match_id (PK, INTEGER, AUTO_INCREMENT)
└── player1_id (FK → users)
└── player2_id (FK → users)
└── player1 Elo before (INTEGER)
└── player1 Elo after (INTEGER)
└── player2 Elo before (INTEGER)
└── player2 Elo after (INTEGER)
└── winner_id (FK → users, NULLABLE)
└── result (ENUM: 'win', 'draw')
└── start_time (TIMESTAMP)
└── end_time (TIMESTAMP)
└── total_moves (INTEGER)
└── player1_moves (INTEGER)
└── player2_moves (INTEGER)
└── time_limit_per_turn (INTEGER, seconds)
└── game_variant (VARCHAR(20), DEFAULT 'standard')

Match_Moves Table (Log)

match_moves

└── move_id (PK, INTEGER, AUTO_INCREMENT)
└── match_id (FK → matches)
└── move_number (INTEGER)
└── player_id (FK → users)
└── target_row (INTEGER)
└── target_col (INTEGER)
└── result (ENUM: 'miss', 'hit', 'sunk')

```
    └── ship_sunk (VARCHAR(20), NULLABLE)
    └── timestamp (TIMESTAMP)
    └── time_taken (INTEGER, milliseconds)
```

Match_Bords Table

```
match_boards
└── board_id (PK, INTEGER, AUTO_INCREMENT)
└── match_id (FK → matches)
└── player_id (FK → users)
└── ship_positions (TEXT/BLOB, serialized data)
// Format: JSON or binary encoding of ship positions
```

Sessions Table

```
sessions
└── session_id (PK, INTEGER, AUTO_INCREMENT)
└── user_id (FK → users)
└── token (VARCHAR(64), UNIQUE)
└── created_at (TIMESTAMP)
└── expires_at (TIMESTAMP)
└── last_activity (TIMESTAMP)
```

5. USE CASES

5.1 Use Case: Đăng ký tài khoản mới

Actor: Người dùng mới

Preconditions: Ứng dụng được mở, người dùng ở màn hình chính

Main Flow:

1. Người dùng click nút "Register"
2. Hệ thống hiển thị form đăng ký
3. Người dùng nhập username, password, display name
4. Người dùng click "Submit"
5. Hệ thống validate dữ liệu
6. Hệ thống tạo tài khoản với ELO = 1000
7. Hệ thống hiển thị thông báo thành công
8. Người dùng được chuyển về màn hình đăng nhập

Alternative Flows:

- 5a. Username đã tồn tại: Hiển thị lỗi "Username already exists"
- 5b. Password quá ngắn: Hiển thị lỗi "Password must be at least 6 characters"

- 5c. Lỗi kết nối: Hiển thị "Connection error. Please try again."

Postconditions: Tài khoản mới được tạo trong database

5.2 Use Case: Tìm và thách đấu người chơi

Actor: Người chơi đã đăng nhập

Preconditions: Người chơi đã đăng nhập, ở lobby

Main Flow:

1. Hệ thống hiển thị danh sách người chơi online
2. Người chơi chọn đối thủ từ danh sách
3. Người chơi click "Challenge"
4. Hệ thống hiển thị dialog cài đặt (time limit, etc.)
5. Người chơi xác nhận
6. Hệ thống gửi lời thách đấu đến đối thủ
7. Đối thủ nhận thông báo và chấp nhận
8. Hệ thống tạo match mới
9. Cả 2 người chơi chuyển sang màn hình bố trí tàu

Alternative Flows:

- 7a. Đối thủ từ chối: Thông báo "Challenge declined"
- 7b. Timeout: Thông báo "Challenge expired"
- 2a. Người chơi click "Quick Match": Hệ thống tự động tìm đối thủ phù hợp

Postconditions: Match được tạo, cả 2 người chơi trong trạng thái "In Game"

5.3 Use Case: Chơi một trận Battleship

Actor: Người chơi 1, Người chơi 2

Preconditions: Match đã được tạo

Main Flow:

1. Cả 2 người chơi bố trí tàu trên bàn cờ của mình
2. Cả 2 click "Ready"
3. Hệ thống random người đi trước (giả sử Player 1)
4. Player 1 chọn ô để bắn
5. Hệ thống kiểm tra và trả về kết quả (Miss/Hit/Sunk)
6. Hệ thống cập nhật UI cho cả 2 người
7. Lượt chuyển sang Player 2
8. Lặp lại bước 4-7 cho đến khi:
 - Một người chơi đánh chìm tất cả tàu → Người kia thắng
 - Một người chơi resign → Người kia thắng
 - Cả 2 đồng ý hòa → Kết quả hòa

9. Hệ thống tính điểm ELO mới
10. Hiển thị màn hình kết quả với stats

Alternative Flows:

- 4a. Player 1 mất kết nối: Chờ 60s, nếu không reconnect thì auto-resign
- 5a. Nước đi không hợp lệ: Hiển thị lỗi, yêu cầu chọn lại
- 8a. Player xin pause: Đổi thủ chấp nhận → Pause game

Postconditions:

- Match kết thúc với kết quả rõ ràng
- ELO được cập nhật
- Match được lưu vào database

5.4 Use Case: Xem lại trận đấu (Replay)

Actor: Người chơi

Preconditions: Người chơi đã đăng nhập, có ít nhất 1 trận đã chơi

Main Flow:

1. Người chơi vào mục "Match History"
2. Hệ thống hiển thị danh sách các trận đã chơi
3. Người chơi chọn trận muốn xem
4. Người chơi click "Replay"
5. Hệ thống load dữ liệu trận đấu
6. Hệ thống hiển thị 2 bàn cờ và control bar
7. Người chơi điều khiển replay (play/pause/next/prev)
8. Hệ thống hiển thị từng nước đi theo thứ tự

Alternative Flows:

- 5a. Dữ liệu bị lỗi/mất: Hiển thị "Replay data unavailable"

Postconditions: Không có thay đổi dữ liệu

6. PHÂN BỐ ĐIỂM CHI TIẾT

6.1 Điểm bắt buộc (Core - 29 điểm)

Chức năng	Điểm	Ghi chú
Xử lý truyền dòng	1	Stream processing

Cơ chế vào/ra socket trên server	2	Socket management
Đăng ký và quản lý tài khoản	2	Registration + management
Đăng nhập và quản lý phiên	2	Login + session
Cung cấp danh sách người chơi	2	Player list with status
Chuyển lời thách đấu	2	Challenge system
Từ chối/Nhận lời thách đấu	1	Challenge response
Chuyển thông tin nước đi	2	Move transmission
Kiểm tra tính hợp lệ nước đi	1	Move validation
Xác định kết quả ván cờ	1	Win/draw detection
Ghi log	1	Move logging
Chuyển kết quả và log ván cờ	2	Result + log transmission
Lưu thông tin ván đấu và replay	2	Match storage
Xây dựng hệ thống tính điểm	2	ELO rating system
Xin ngừng ván cờ/mời hòa	1	Pause/draw offers
Xin đấu lại người chơi	1	Rematch
Giao diện đồ họa	3	GUI implementation
TỔNG CORE	29	Điểm tối thiểu

6.2 Điểm nâng cao (Optional - lên đến 10 điểm)

Chức năng	Điểm	Ghi chú
Auto-matchmaking	3	Tự động ghép cặp theo ELO
In-game chat	2	Chat trong trận
Custom board setup	2-3	Tùy chỉnh bàn cờ
Game variants (Salvo, etc.)	3-5	Các biến thể game

6.3 Khuyến nghị phân bổ

Ưu tiên:

1. Hoàn thiện các chức năng core trước
 2. Đảm bảo game chơi mượt mà, ít bug
 3. Thêm tính năng nâng cao dễ nhất trước
 4. UI/UX đẹp và professional
-

7. KẾ HOẠCH PHÁT TRIỂN

7.1 Phase 1: Foundation (2 tuần)

Week 1:

- Setup môi trường phát triển (Linux, Git, IDE)
- Thiết kế database schema
- Implement basic socket server/client
- Implement authentication (register/login)

Week 2:

- Implement session management
- Player list display
- Basic matchmaking (challenge system)
- Challenge accept/decline

Deliverable: Có thể đăng ký, đăng nhập, thấy người chơi khác và gửi/nhận thách đấu

7.2 Phase 2: Core Gameplay (2-3 tuần)

Week 3:

- Implement game board logic
- Ship placement (manual + random)
- Move validation
- Hit/miss detection

Week 4:

- Turn management
- Win/draw detection
- Basic game UI (board rendering)
- Move animation

Week 5:

- Polish gameplay
- Bug fixes

- Timeout handling
- Disconnect handling

Deliverable: Có thể chơi một trận Battleship hoàn chỉnh

7.3 Phase 3: ELO & Persistence (1-2 tuần)

Week 6:

- Implement ELO calculation
- Match history storage
- Move logging
- Replay functionality

Week 7:

- Statistics tracking
- Profile page
- Replay UI
- Testing

Deliverable: Hệ thống xếp hạng hoạt động, có thể xem lại trận đấu

7.4 Phase 4: Polish & Advanced Features (1-2 tuần)

Week 8:

- UI improvements
- Add animations and sounds
- Implement chosen advanced features
- Comprehensive testing

Week 9:

- Bug fixes
- Performance optimization
- Documentation
- Final testing

Deliverable: Sản phẩm hoàn chỉnh, professional

7.5 Tổng thời gian: 8-10 tuần

8. CÔNG NGHỆ VÀ TOOLS

8.1 Ngôn ngữ và Frameworks

Backend (Server):

- C++ (C++11 trở lên)
- POSIX Sockets (TCP)
- pthread cho multithreading
- SQLite3 hoặc file-based storage

Frontend (Client):

- C++ (C++11 trở lên)
- GTK+ 3.0 hoặc SDL2 cho GUI
- Cairo (với GTK) cho drawing
- POSIX Sockets cho networking

Libraries hữu ích:

- JSON parsing: nlohmann/json hoặc RapidJSON
- Crypto: OpenSSL (cho SHA-256)
- Testing: Google Test (optional)

8.2 Development Tools

- **IDE:** Visual Studio Code, CLion, hoặc Vim/Emacs
- **Compiler:** GCC/G++ 7.0+
- **Build System:** Makefile hoặc CMake
- **Version Control:** Git + GitHub/GitLab
- **Database Tool:** DB Browser for SQLite
- **Debugging:** GDB, Valgrind

8.3 Testing Tools

- **Unit Testing:** Google Test
- **Network Testing:** netcat, Wireshark
- **Load Testing:** Custom scripts với multiple clients

9. RỦI RO VÀ GIẢI PHÁP

9.1 Rủi ro kỹ thuật

Rủi ro	Mức độ	Giải pháp
Deadlock/race conditions trong multithreading	Cao	Sử dụng mutex cẩn thận, review code kỹ
Network latency	Trung bình	Optimize protocol, sử dụng binary format

Cheating (client-side validation)	Cao	Luôn validate trên server
Scalability issues	Trung bình	Design tốt từ đầu, sử dụng thread pool
Data corruption	Trung bình	Transactions, regular backups

9.2 Rủi ro dự án

Rủi ro	Mức độ	Giải pháp
Underestimate effort	Cao	Ưu tiên core features trước
Scope creep	Trung bình	Tập trung vào requirements, từ chối tính năng không cần thiết
Team communication	Trung bình	Daily standups, Git workflow rõ ràng
Technical debt	Trung bình	Code review, refactoring sớm

10. TIÊU CHÍ CHẤP NHẬN (ACCEPTANCE CRITERIA)

10.1 Chức năng

- [] Người dùng có thể đăng ký và đăng nhập thành công
- [] Hiển thị danh sách người chơi online với trạng thái real-time
- [] Gửi và nhận lời thách đấu hoạt động đúng
- [] Có thể bố trí tàu và bắt đầu trận đấu
- [] Mỗi nước đi được validate đúng luật
- [] Kết quả trận đấu (thắng/thua/hòa) chính xác
- [] ELO được tính và cập nhật đúng
- [] Có thể xem lại trận đấu (replay)
- [] Pause/resign/draw offers hoạt động đúng
- [] Rematch hoạt động

10.2 Phi chức năng

- [] Server hỗ trợ ít nhất 20 concurrent users
- [] Response time < 500ms cho hầu hết operations
- [] UI mượt mà, không lag
- [] Không crash trong normal operations
- [] Data được lưu persistent
- [] Code được organize tốt với comments

10.3 Documentation

- [] README với hướng dẫn compile và run
 - [] Requirements document (tài liệu này)
 - [] API/Protocol documentation
 - [] Database schema documented
-

11. GLOSSARY (Thuật ngữ)

Thuật ngữ	Định nghĩa
ELO Rating	Hệ thống xếp hạng được phát triển bởi Arpad Elo, dùng để đánh giá kỹ năng người chơi tương đối
Session Token	Chuỗi ký tự duy nhất để xác thực người dùng trong phiên làm việc
Matchmaking	Quá trình ghép cặp người chơi với nhau
Lobby	Khu vực chờ nơi người chơi có thể xem danh sách người khác và tìm trận đấu
Rematch	Trận đấu lại giữa 2 người chơi vừa chơi với nhau
Replay	Xem lại trận đấu đã chơi
Hit	Bắn trúng một phần của tàu
Miss	Bắn trượt
Sunk	Đánh chìm hoàn toàn một tàu
Socket	Endpoint cho giao tiếp mạng hai chiều
Thread-safe	Code an toàn khi được thực thi bởi nhiều threads đồng thời
Race condition	Bug xảy ra khi timing của events ảnh hưởng đến tính đúng đắn của code

12. REFERENCES (Tài liệu tham khảo)

12.1 Game Rules

- [Battleship Game Rules - Hasbro](#)
- [Battleship - Wikipedia](#)

12.2 ELO Rating System

- [ELO Rating System - Wikipedia](#)
- [Understanding ELO Ratings](#)

12.3 Network Programming

- [Beej's Guide to Network Programming](#)
- [POSIX Sockets Tutorial](#)

12.4 GUI Development

- [GTK+ 3 Documentation](#)
- [SDL2 Documentation](#)

12.5 C++ Best Practices

- [C++ Core Guidelines](#)
 - [Effective Modern C++ by Scott Meyers](#)
-

13. APPENDICES

Appendix A: Message Protocol Examples

Example 1: Login Request

```
{  
  "type": "AUTH_LOGIN",  
  "username": "player1",  
  "password": "hashed_password"  
}
```

Example 2: Move Request

```
{  
  "type": "MOVE",  
  "token": "abc123xyz",  
  "matchId": "m12345",  
  "data": {  
    "row": 5,  
    "col": 7  
  }  
}
```

Example 3: Move Result

```
{  
  "type": "MOVE_RESULT",  
  "matchId": "m12345",  
  "move": {  
    "player": "player1",  
    "row": 5,  
    "col": 7,  
    "result": "hit",  
    "shipHit": "cruiser"  
  },  
  "nextTurn": "player1"  
}
```

Appendix B: Ship Configuration

```
struct Ship {  
    string name;  
    int length;  
    vector<pair<int,int>> positions; // coordinates  
    int hits; // number of hits taken  
    bool isSunk;  
};  
  
// Standard ships  
const vector<Ship> STANDARD_SHIPS = {  
    {"Aircraft Carrier", 5},  
    {"Battleship", 4},  
    {"Cruiser", 3},  
    {"Submarine", 3},  
    {"Destroyer", 2}  
};
```

Appendix C: ELO Calculation Code Example

```
// Calculate new ELO rating  
pair<int, int> calculateNewELO(int eloA, int eloB, double scoreA) {  
    const double K = 32.0; // K-factor  
  
    // Expected scores  
    double expectedA = 1.0 / (1.0 + pow(10, (eloB - eloA) / 400.0));  
    double expectedB = 1.0 - expectedA;  
  
    // Actual scores (scoreA = 1 for win, 0.5 for draw, 0 for loss)  
    double scoreB = 1.0 - scoreA;
```

```
// New ratings  
int newEloA = eloA + (int)(K * (scoreA - expectedA));  
int newEloB = eloB + (int)(K * (scoreB - expectedB));  
  
return {newEloA, newEloB};  
}
```

14. CHANGE LOG

Version	Date	Author	Changes
1.0	29/10/2025	Team	Initial requirements document