**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**-------🙡🙣 🕮 🙡🙣-------**



**BÀI BÁO CÁO VỀ LẬP TRÌNH GAME CỜ SODUKU**

**Giảng viên hướng dẫn :** Mai Ngọc Châu

**Môn :** Lập trình mạng

**Nhóm thực hiện :** nhóm 7

**Sinh viên thực hiện:**

|  |  |
| --- | --- |
| Họ tên sinh viên | Mã số sinh viên |
| 1. Đoàn Phan Vĩnh Phú 2. Nguyễn Quang Tiến 3. Đào Minh Tâm 4. Phan Quyết Tâm Phú 5. Huỳnh Bá Thành 6. Huỳnh Thị Minh Thơ |  |

*Tp.Hồ Chí Minh, ngày …. tháng ….. năm 2025*

# **LỜI MỞ ĐẦU**

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, các ứng dụng mạng ngày càng đóng vai trò quan trọng trong đời sống và học tập. Môn học Lập trình mạng không chỉ giúp sinh viên nắm vững kiến thức lý thuyết về giao tiếp giữa các thiết bị qua mạng, mà còn tạo cơ hội áp dụng vào các dự án thực tế.

Xuất phát từ mong muốn kết hợp giữa kiến thức lập trình socket và tư duy logic trong trò chơi trí tuệ, nhóm chúng em đã lựa chọn đề tài “Lập trình game cờ Sodoku” làm đồ án môn học. Dự án mô phỏng trò chơi Sudoku quen thuộc, đồng thời mở rộng thêm tính năng thách đấu giữa hai người chơi thông qua mạng Client – Server, giúp tăng tính tương tác và thử thách.Thông qua quá trình thực hiện, nhóm chúng em đã củng cố được kỹ năng xây dựng mô hình mạng, xử lý luồng dữ liệu, và tổ chức giao tiếp hai chiều giữa client và server. Bên cạnh đó, dự án cũng giúp chúng em rèn luyện tư duy lập trình, khả năng làm việc nhóm, và vận dụng kiến thức học được vào việc phát triển một ứng dụng hoàn chỉnh.

Chúng em xin chân thành cảm ơn Thầy Mai Ngọc Châu đã hướng dẫn tận tình, giúp chúng em hiểu rõ hơn về lập trình mạng và ứng dụng vào thực tế. Rất mong nhận được sự góp ý của thầy để chúng em có thể hoàn thiện hơn trong các dự án sau.

**CHƯƠNG 1. TỔNG QUAN VÀ MỤC TIÊU NGHIÊN CỨU VỀ GAME SODUKU**

* 1. *Giới thiệu chung*

Trong thế giới trò chơi trí tuệ, Sudoku là một biểu tượng của tư duy logic và sự kiên nhẫn. Ra đời tại Mỹ vào thập niên 1970 dưới tên gọi “Number Place”, do Howard Gans sáng tạo, trò chơi nhanh chóng được yêu thích trên toàn cầu nhờ luật chơi đơn giản nhưng đầy thử thách. Nhiệm vụ của người chơi là điền các con số từ 1 đến 9 vào bảng 9x9 sao cho mỗi hàng, mỗi cột và mỗi khối 3x3 đều chứa đủ các số từ 1 đến 9 mà không bị trùng lặp.

Đề tài “Lập trình game cờ Sudoku” được thực hiện nhằm tái hiện trò chơi này dưới dạng ứng dụng mạng trực tuyến (online), hoạt động theo mô hình Client–Server. Thông qua đó, hai người chơi có thể kết nối đến cùng một máy chủ (Server) để thi đấu trực tiếp trong thời gian thực.

Không chỉ dừng lại ở việc mô phỏng trò chơi, hệ thống còn được bổ sung tính năng đếm thời gian suy nghĩ, giúp tạo nên môi trường cạnh tranh công bằng, thúc đẩy khả năng phản xạ và tư duy chiến lược của người chơi. Mục tiêu cuối cùng là xây dựng một trò chơi vừa mang tính giải trí, vừa mang giá trị học tập và rèn luyện trí não.

*1.2.Mục tiêu đề tài*

Đề tài hướng đến việc nghiên cứu và phát triển ứng dụng trò chơi Sudoku theo mô hình Client–Server, tập trung vào khả năng tương tác trực tuyến giữa hai người chơi. Mục tiêu chính bao gồm:

* *Xây dựng hệ thống Client–Server ổn định:* Cho phép nhiều client kết nối đồng thời đến server.
* *Phát triển cơ chế thách đấu:* Người chơi có thể gửi và nhận lời mời, tạo phòng đấu khi cả hai đồng ý.
* *Quản lý và lưu trữ lịch sử trận đấu:* Ghi lại thông tin người chơi, thời gian suy nghĩ, và kết quả từng trận.
* *Cải thiện trải nghiệm người dùng:* Hệ thống đếm thời gian suy nghĩ, thống kê kết quả, và giao diện thân thiện.
* *Hỗ trợ khả năng mở rộng:* Cho phép phát triển thêm các tính năng như bảng xếp hạng, trò chuyện, hoặc nhiều phòng chơi.

**CHƯƠNG 2. PHÂN TÍCH, THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG**

*2.1. Công nghệ và môi trường phát triển*

*2.1.1. Ngôn ngữ lập trình và Framework chính*

|  |  |  |
| --- | --- | --- |
| Thành phần | Công nghệ | Vai trò / Lý do lựa chọn |
| Server (Backend) | Python 3.10+ và FastAPI | * Lắng nghe nhiều kết nối đồng thời thông qua WebSocket. * Quản lý người chơi online, xử lý thách đấu, sinh bảng Sudoku, kiểm tra kết quả. * Lưu lịch sử trận đấu bằng SQLite hoặc PostgreSQL. |
| Client (Frontend) | JavaScript (ES6+) và ReactJS | * Xây dựng giao diện động, component hóa bảng Sudoku. * Gửi/nhận dữ liệu với Server, hiển thị real-time, ReactJS phù hợp giao diện động. |
| Giao diện | HTML5, CSS3, Tailwind CSS / Bootstrap | * Xây dựng bố cục và giao diện người chơi trực quan, thân thiện, responsive. |

*2.1.2. Công nghệ và Thư viện sử dụng*

* **Phần Server (Backend):**
  + *Uvicorn:* Web server chạy FastAPI ở chế độ bất đồng bộ.
  + *SQLAlchemy:* ORM quản lý truy cập cơ sở dữ liệu.
  + *WebSocket (websockets hoặc fastapi[websockets]):* Giao tiếp liên tục, real-time giữa client và server.
  + *Pydantic:* Kiểm tra dữ liệu vào/ra (request/response).
  + *SQLite (Phát triển) / PostgreSQL (Triển khai):* Cơ sở dữ liệu lưu lịch sử trận đấu.
* **Phần Client (Frontend):**
  + *Axios / Fetch API:* Gửi request REST (đăng nhập, lấy lịch sử).
  + *WebSocket API:* Giao tiếp trực tiếp khi chơi thách đấu.

a) Môi trường phát triển:

* *Hệ điều hành:* Windows 10/11, macOS, hoặc Ubuntu 20.04+.
* *IDE:* Visual Studio Code, PyCharm.
* *Công cụ quản lý gói:* pip, virtualenv (Python); npm hoặc yarn (React).
* *Công cụ kiểm thử:* Postman, Thunder Client.
* *Quản lý mã nguồn:* Git, GitHub.

b)Môi trường triển khai:

* *Server:* Được triển khai trên các nền tảng như Render, Railway, hoặc PythonAnywhere. Có thể đóng gói bằng Docker để đảm bảo tính nhất quán.
* *Client:* Được build và triển khai tĩnh trên Vercel, Netlify hoặc GitHub Pages.
* *Cơ sở dữ liệu:* SQLite dùng trong giai đoạn phát triển; PostgreSQL dùng khi triển khai thực tế.

c) Lý do lựa chọn công nghệ:

* *FastAPI:* Framework hiện đại, hiệu năng cao, dễ tích hợp WebSocket.
* *ReactJS:* Tạo giao diện linh hoạt, cập nhật real-time.
* *WebSocket:* Cho phép giao tiếp hai chiều, giúp trò chơi vận hành trơn tru theo thời gian thực.
* *PostgreSQL/SQLite:* Quản lý dữ liệu dễ dàng, bền vững và dễ mở rộng.

*2.2. Yêu cầu hệ thống*

a) Yêu cầu chức năng:

* *Kết nối Client–Server:* Cho phép nhiều client cùng truy cập qua địa chỉ IP và cổng định sẵn.
* *Đăng nhập và định danh người chơi:* Mỗi người nhập tên (username); server quản lý danh sách online.
* *Chức năng thách đấu:* Gửi, nhận và chấp nhận lời mời thách đấu. Khi đồng ý, server tạo phòng và gửi cùng bảng Sudoku.
* *Gameplay:* Hai người chơi cùng giải bảng Sudoku, mỗi thao tác gửi đến server để xác nhận hợp lệ.
* *Kết thúc trận:* Server kiểm tra kết quả, xác định người thắng dựa vào độ chính xác và thời gian suy nghĩ.
* *Lưu lịch sử:* Server lưu kết quả, thời gian, người thắng, phục vụ thống kê và phân tích.

b) Yêu cầu phi chức năng:

* *Hiệu năng:* Độ trễ phản hồi dưới 1 giây, server xử lý 5–10 client cùng lúc.
* *Bảo mật:* Mã hóa dữ liệu, xác thực token, tránh truy cập trái phép.
* *Ổn định:* Xử lý tốt trường hợp client ngắt kết nối đột ngột.
* *Mở rộng:* Dễ thêm tính năng như nhiều phòng chơi, chat, bảng xếp hạng.
* *Thân thiện người dùng:* Giao diện trực quan, có đồng hồ và thông báo rõ ràng.

*2.3. Các module chính*

Hệ thống hoạt động real-time qua WebSocket, gồm 6 thành phần chính:

1. *WebSocket Gateway:*
   * Tiếp nhận kết nối client qua **/ws?token=**....
   * Xác thực JWT, chuyển tiếp thông điệp như ( move,challenge, kết quả ) đến Room Manager.
2. *Authentication / API:*
   * Các REST endpoint**: /login, /history, /profile**.
   * Quản lý xác thực người dùng, truy xuất lịch sử trận đấu.
3. *Matchmaker / Room Manager:*
   * Quản lý danh sách người chơi online. Tạo match\_id và MatchState (chứa board, timers, players) khi có hai người chấp nhận thách đấu.
   * Tạo phòng, sinh bảng Sudoku, và quản lý trạng thái trận.
4. *Game Engine:*
   * Sinh bảng Sudoku (theo độ khó).
   * Kiểm tra hợp lệ nước đi, cập nhật trạng thái, tính điểm, xác định người thắng.
5. *Timer Manager:*
   * Quản lý thời gian mỗi lượt, gửi thông báo timeout, đảm bảo công bằng.
6. *Persistence Layer:*
   * Lưu dữ liệu vào cơ sở dữ liệu (users, matches, moves).
   * Hỗ trợ SQLite trong giai đoạn phát triển, PostgreSQL khi triển khai.

*2.4. Thiết kế cơ sở dữ liệu và bảo mật*

**Cấu trúc cơ sở dữ liệu chính:**

|  |  |  |
| --- | --- | --- |
| **Bảng** | **Thuộc tính chính** | **Mô tả** |
| Users | id, username, created\_at | Lưu thông tin người chơi |
| matches | id, player\_a\_id, player\_b\_id, start\_time, end\_time, winner\_id, board | Lưu thông tin trận đấu |
| moves | id, match\_id, player\_id, row, col, value, think\_ms | Lưu các nước đi và thời gian suy nghĩ |

**Bảo mật hệ thống:**

* Giao tiếp qua **WSS (TLS)** để mã hóa dữ liệu.
* Xác thực bằng **JWT Token** cho mọi kết nối.
* Hash mật khẩu bằng **bcrypt**.
* Kiểm tra dữ liệu nghiêm ngặt bằng **Pydantic**.
* Giới hạn tần suất gửi yêu cầu để ngăn tấn công từ chối dịch vụ.

*2.5. Mô tả luồng hoạt động*

**A. Tạo trận (Challenge Flow)**

1. Client A → Server: { "type": "challenge\_request", "from": "A", "to": "B" }
2. Server → B
3. B → Server: { "type": "challenge\_response", "accept": true }
4. Server tạo match\_id, sinh board: { "type": "game\_start", "match\_id": "uuid", "board": "base64/JSON", "turn": "A", "per\_move\_ms": 60000 }

**🔸 B. Đi nước & đồng bộ (Move Flow)**

1. Player → Server: { "type": "move", "match\_id": "uuid", "player": "A", "r": 0, "c": 1, "v": 5 }
2. Server:

* Kiểm tra lượt hợp lệ.
* Validate move bằng Game Engine.
* Cập nhật board, lưu move.
* Broadcast move\_result + next\_turn.

**🔸 C. Hoàn thành / Kết thúc**

Player nhấn complete.

Server validate toàn bộ board.

Nếu hợp lệ → declare winner, lưu kết quả, gửi game\_end.

**Điểm nổi bật của hệ thống:**

* **Server là nguồn chân lý (Authoritative Server):** Mọi luật chơi và kết quả đều được xác định ở phía server.
* **Timer chạy phía server:** Đảm bảo tính công bằng tuyệt đối.
* **Dễ mở rộng:** Có thể phát triển thêm nhiều phòng đấu, chat và bảng xếp hạng.

*2.6. Kiểm thử, ghi log và triển khai*

* **Kiểm thử:**
  + *Unit test:* Game Engine, timer logic.
  + *Integration test:* WS flow (thách đấu, di chuyển, timeout).
* **Logging:**
  + Ghi log JSON: match start/end, lỗi, thời gian xử lý.
* **Triển khai:**
  + *Dev:* Uvicorn + SQLite.
  + *Prod:* Nginx load balancing, Redis, PostgreSQL, Docker + CI/CD.

*2.7. Sơ đồ mô tả hệ thống game Soduku*

