**1. Thiết Lập Môi Trường Phát Triển**

Để bắt đầu phát triển ứng dụng, cần chuẩn bị môi trường làm việc phù hợp:

* **Cài đặt .NET Core 3.1**: Tải và cài đặt .NET Core 3.1 SDK từ trang chính thức của Microsoft. Đây là nền tảng cần thiết để phát triển ứng dụng server và client. Sau khi cài đặt, kiểm tra phiên bản bằng lệnh dotnet --version trong terminal.
* **Cài đặt VSCode**: Tải Visual Studio Code (VSCode) và cài đặt làm môi trường phát triển chính. Thêm các extension hỗ trợ như:
  + **C# for Visual Studio Code**: Để hỗ trợ lập trình C#.
  + Các extension khác như GitLens (nếu dùng Git) hoặc Terminal tích hợp.

**2. Tạo Dự Án Server**

Ứng dụng cờ vua sẽ sử dụng kiến trúc client-server, với server được xây dựng bằng gRPC:

* **Tạo dự án gRPC**: Mở terminal và chạy lệnh dotnet new grpc -o ChessServer để tạo một dự án gRPC mới có tên "ChessServer".
* **Thêm gói NuGet**: Trong thư mục dự án ChessServer, chạy lệnh dotnet add package ChessDotNet để tích hợp thư viện ChessDotNet, hỗ trợ logic cờ vua như quản lý trạng thái bàn cờ và kiểm tra nước đi.
* **Định nghĩa dịch vụ gRPC**:
  + Tạo file .proto (ví dụ: chess.proto) trong thư mục Protos.
  + Định nghĩa các dịch vụ như Connect, StartGame, MakeMove, Resign, và GetGameState.
  + Định nghĩa các thông điệp (message) như vị trí quân cờ, lượt đi, và trạng thái ván cờ.

**3. Triển Khai Logic Server**

Server sẽ quản lý toàn bộ logic của ván cờ và giao tiếp với client:

* **Quản lý kết nối**:
  + Khi hai client kết nối, server gán màu quân cờ (trắng hoặc đen) cho từng client.
  + Lưu trữ thông tin kết nối của từng client.
* **Quản lý trạng thái ván cờ**:
  + Sử dụng lớp ChessGame từ ChessDotNet để khởi tạo và theo dõi trạng thái ván cờ (vị trí quân cờ, lượt đi).
* **Xác nhận nước đi**:
  + Khi client gửi yêu cầu di chuyển, server gọi ChessGame.MakeMove để kiểm tra nước đi hợp lệ. Nếu hợp lệ, cập nhật trạng thái ván cờ.
* **Phát hiện kết thúc ván cờ**:
  + Dùng các phương thức như IsCheckmate (chiếu hết) và IsStalemate (hòa) từ ChessDotNet để kiểm tra trạng thái kết thúc.
* **Streaming cập nhật trạng thái**:
  + Sử dụng tính năng streaming của gRPC để gửi trạng thái ván cờ (dạng FEN hoặc tương tự) đến cả hai client sau mỗi nước đi hoặc sự kiện.

**4. Tạo Dự Án Client**

Client sẽ là ứng dụng giao diện người dùng, được xây dựng bằng WPF:

* **Tạo dự án WPF**: Chạy lệnh dotnet new wpf -o ChessClient để tạo dự án client có tên "ChessClient".
* **Thêm gói NuGet**: Trong thư mục ChessClient, thêm các gói:
  + Grpc.Net.Client: Để kết nối với server gRPC.
  + Google.Protobuf: Để xử lý dữ liệu gRPC.
  + ChessDotNet: Để hiển thị trạng thái bàn cờ.
* **Thiết kế giao diện**: Trong WPF, xây dựng giao diện gồm:
  + **TextBox**: Để người dùng nhập địa chỉ IP của server.
  + **Nút Connect**: Kết nối đến server.
  + **Grid 8x8**: Hiển thị bàn cờ cờ vua.
  + **Nút New Game**: Bắt đầu ván mới.
  + **Nút Resign**: Đầu hàng.
  + **TextBlock**: Hiển thị thông báo trạng thái (ví dụ: "Đến lượt bạn", "Chiếu hết").

**5. Triển Khai Logic Client**

Logic client tập trung vào giao tiếp với server và hiển thị thông tin:

* **Kết nối đến server**:
  + Khi người dùng nhập IP và nhấn "Connect", thiết lập kênh gRPC đến server và nhận thông tin màu quân (trắng/đen).
* **Hiển thị bàn cờ**:
  + Phân tích chuỗi FEN (Forsyth-Edwards Notation) từ server để vẽ vị trí quân cờ lên Grid 8x8.
* **Xử lý tương tác người dùng**:
  + Khi đến lượt, cho phép người chơi nhấp chọn quân cờ và ô đích, sau đó gửi yêu cầu di chuyển qua gRPC.
* **Xử lý nâng quân**:
  + Khi quân tốt (pawn) đến hàng cuối, hiển thị hộp thoại để chọn nâng cấp thành Queen, Rook, Bishop hoặc Knight, rồi gửi lựa chọn đến server.
* **Nhận cập nhật trạng thái**:
  + Đăng ký stream từ server để nhận cập nhật trạng thái ván cờ và làm mới giao diện ngay lập tức.

**6. Xử Lý Kết Thúc Ván Cờ**

Đảm bảo client phản ánh đúng trạng thái khi ván cờ kết thúc:

* **Nhận kết quả**:
  + Khi server gửi thông báo kết thúc (chiếu hết, hòa, hoặc đầu hàng), hiển thị thông báo trên TextBlock (ví dụ: "Trắng thắng!", "Hòa!").
* **Cho phép bắt đầu ván mới**:
  + Sau khi ván kết thúc, kích hoạt nút "New Game" để gửi yêu cầu bắt đầu ván mới đến server.

**7. Kiểm Thử Và Debug**

Đảm bảo ứng dụng hoạt động ổn định:

* **Kiểm thử chức năng**:
  + Kiểm tra kết nối giữa client và server.
  + Thử các kịch bản như di chuyển hợp lệ/không hợp lệ, đầu hàng, và kết thúc ván cờ.
* **Debug**:
  + Sử dụng công cụ debug trong VSCode để tìm và sửa lỗi, đặc biệt là trong giao tiếp gRPC và logic cờ vua.

**8. Tài Liệu Hướng Dẫn**

Cung cấp tài liệu để người dùng dễ dàng sử dụng:

* **Hướng dẫn cài đặt**:
  + Mô tả cách cài .NET Core, chạy server và client trên máy tính.
* **Hướng dẫn sử dụng**:
  + Hướng dẫn cách nhập IP, kết nối, chơi cờ, và sử dụng các nút chức năng.

**9. Đóng Gói Và Triển Khai**

Chuẩn bị ứng dụng để triển khai thực tế:

* **Đóng gói ứng dụng**:
  + Sử dụng lệnh dotnet publish để tạo file thực thi cho cả server và client.
* **Triển khai**:
  + Cung cấp hướng dẫn chạy server trên một máy và kết nối client qua mạng LAN.