

BỘ GIAO THÔNG VẬN TẢI
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN
—o0o—



BÁO CÁO BÀI TẬP LỚN

Đề tài:

LẬP TRÌNH ỨNG DỤNG TRÒ CHƠI CỜ VUA TRÊN NỀN TẢNG ANDROID

Học phần: LẬP TRÌNH THIẾT BỊ DI ĐỘNG

Mã học phần: 010112103401

SVTH: Nguyễn Quốc Tùng
Nguyễn Nhật Lâm
Nguyễn Phi Khanh

CBHD: Thầy Trương Quang Tuấn

TP. Hồ Chí Minh - 2025

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu Trường Đại học Giao thông Vận tải đã tạo điều kiện thuận lợi, giúp chúng em có được môi trường học tập và nghiên cứu lý tưởng nhất. Trong quá trình thực hiện đồ án Lập trình thiết bị di động, em xin chân thành gửi lời cảm ơn đến Thầy Trương Quang Tuấn – người đã tận tình giảng dạy và hướng dẫn em trong suốt quá trình học tập và thực hiện đề tài này.

Thầy không chỉ cung cấp những kiến thức chuyên môn quý báu mà còn luôn khuyến khích, tạo điều kiện để em rèn luyện khả năng tư duy, sáng tạo và ứng dụng thực tiễn.

Mặc dù đã rất cố gắng, nhưng do kiến thức còn hạn chế và thời gian nghiên cứu có giới hạn, đồ án vẫn còn nhiều thiếu sót. Chúng em mong thầy có thể bỏ qua những sai sót đó và dành cho chúng em những góp ý chân thành. Những đóng góp quý báu của thầy sẽ là kinh nghiệm vô cùng quý giá, giúp chúng em tự tin hơn trong việc hoàn thiện các dự án tiếp theo.

Một lần nữa, em xin chân thành cảm ơn!

BẢNG PHÂN CÔNG CÔNG VIỆC CHO CÁC THÀNH VIÊN

MSSV	Họ và tên	Nhiệm vụ
2251120259	Nguyễn Quốc Tùng	...
2251120222	Nguyễn Nhật Lâm	...
2251120217	Nguyễn Phi Khanh	...

Mục lục

LỜI MỞ ĐẦU	i
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	1
1.1. Lý do chọn đề tài	1
1.2. Mục tiêu của đồ án	1
1.3. Phạm vi nghiên cứu	1
1.4. Phương pháp nghiên cứu	2
CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG	2
2.1. Khảo sát hiện trạng	2
2.1.1. Đối tượng người dùng chính	2
2.1.1. Nhu cầu và hành vi	3
2.1.3. So sánh với các sản phẩm hiện có	3
2.2. Phân tích yêu cầu hệ thống	3
2.2.1. Yêu cầu chức năng	3
2.2.2. Yêu cầu phi chức năng	4
2.3. Sơ đồ use case hệ thống	5
2.4. Kiến trúc hệ thống	5
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG	5
3.1. Kiến trúc tổng thể	5
3.2. Thiết kế giao diện người dùng (UI)	6
3.2.1. Màn hình chính	6
3.2.2. Màn hình đăng nhập / đăng ký	6
3.2.3. Màn hình chơi cờ	6
3.2.4. Màn hình bạn bè và tìm kiếm	7
3.2.5. Màn hình hồ sơ cá nhân	7
3.2.6. Màn hình lịch sử và chi tiết trận đấu	7
3.3. Thiết kế cơ sở dữ liệu	7
3.4. Thiết kế thuật toán chơi với AI	9
3.4.1. Xác định tất cả các nước đi hợp lệ	9
3.4.2. Đánh giá và lựa chọn nước đi tốt nhất	10
3.4.3. Thực hiện nước đi của AI	11
3.4.4. Nhận xét về thuật toán	11
3.5. Luồng hoạt động chính	11
3.5.1. Luồng chơi online	11
3.5.2. Luồng thêm bạn	11
CHƯƠNG 4: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN	11

4.1. Kết quả đạt được	11
4.2. Hạn chế	13
4.3. Hướng phát triển	14
4.4. Kết quả học tập đạt được	15
KẾT LUẬN	16
TÀI LIỆU THAM KHẢO	17

LỜI MỞ ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 và sự bùng nổ của các thiết bị di động thông minh, ứng dụng di động đã trở thành một phần không thể thiếu trong đời sống hiện đại. Chúng không chỉ phục vụ nhu cầu giao tiếp, làm việc, học tập mà còn mang đến những trải nghiệm giải trí đa dạng và phong phú. Sự giao thoa giữa công nghệ và các trò chơi truyền thống, vốn được coi là biểu tượng của trí tuệ và chiến lược, đã mở ra một chân trời mới cho việc phát triển các ứng dụng giải trí mang tính giáo dục và thư giãn cao. Cờ vua, một trò chơi có lịch sử lâu đời và được biết đến với khả năng rèn luyện tư duy logic, khả năng phân tích và tầm nhìn chiến lược, là một ví dụ điển hình.

Xuất phát từ nhu cầu ngày càng tăng của người dùng về một nền tảng chơi cờ vua tiện lợi, đa năng và có thể truy cập mọi lúc mọi nơi, đề án “Ứng dụng ChessMate – Chơi cờ vua trên Android” đã được hình thành và triển khai. Mục tiêu chính của đề án là xây dựng một ứng dụng di động hoàn chỉnh, cung cấp một trải nghiệm chơi cờ vua mượt mà và hấp dẫn trên hệ điều hành Android. Ứng dụng không chỉ giới hạn ở việc tái hiện bàn cờ và các quân cờ một cách trực quan, mà còn hướng đến việc hỗ trợ đa dạng các chế độ chơi, đáp ứng nhu cầu của nhiều đối tượng người dùng khác nhau, từ những người mới bắt đầu làm quen với cờ vua cho đến những người chơi có kinh nghiệm muốn nâng cao trình độ và giao lưu với cộng đồng. Các chế độ chơi được tích hợp bao gồm khả năng so tài trí tuệ với bạn bè trên cùng một thiết bị, thử thách bản thân với trí tuệ nhân tạo ở nhiều cấp độ khác nhau, và kết nối với cộng đồng người chơi cờ vua trực tuyến để tham gia vào những ván đấu kịch tính và học hỏi kinh nghiệm.

Để đảm bảo tính hiện đại, hiệu suất và khả năng mở rộng của ứng dụng, đề án được xây dựng dựa trên các công nghệ tiên tiến và các kiến trúc phần mềm phổ biến trong phát triển ứng dụng Android. Ngôn ngữ lập trình Kotlin, với những ưu điểm vượt trội về cú pháp, tính an toàn và khả năng tương tác tốt với Java, đã được lựa chọn làm ngôn ngữ phát triển chính. Giao diện người dùng của ứng dụng được thiết kế bằng Jetpack Compose, một framework UI hiện đại và khai báo của Google, cho phép xây dựng giao diện một cách nhanh chóng, trực quan và có khả năng tái sử dụng cao. Nền tảng dữ liệu đám mây Firestore đã được tích hợp để quản lý dữ liệu người dùng, thông tin tài khoản, lịch sử trận đấu và các dữ liệu liên quan khác một cách hiệu quả và an toàn. Bên cạnh đó, việc áp dụng mô hình kiến trúc MVVM (Model-View-ViewModel) giúp tách biệt các mối quan tâm trong ứng dụng, tăng cường khả năng kiểm thử, bảo trì và mở rộng ứng dụng trong tương lai, đồng thời đảm bảo một luồng dữ liệu rõ ràng và dễ quản lý giữa giao diện người dùng và logic nghiệp vụ của ứng dụng.

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

Cờ vua là một trò chơi trí tuệ lâu đời, đòi hỏi sự tư duy, tính toán chiến lược và khả năng phản xạ. Việc phát triển một ứng dụng chơi cờ vua trên nền tảng di động không chỉ là cơ hội để vận dụng các kiến thức lập trình đã học mà còn nhằm tạo ra một công cụ giải trí hữu ích, giúp người dùng có thể chơi và học hỏi mọi lúc, mọi nơi.

Ngoài ra, xu hướng kết nối người dùng thông qua mạng Internet và các dịch vụ đám mây như Firebase đã mở ra nhiều tiềm năng cho các ứng dụng tương tác thời gian thực như cờ vua trực tuyến. Do đó, nhóm em quyết định chọn đề tài này để vừa học hỏi, vừa ứng dụng các kiến thức đã học vào thực tế.

1.2. Mục tiêu của đồ án

- Xây dựng một ứng dụng cờ vua hoàn chỉnh, giao diện thân thiện, dễ sử dụng.
- Cung cấp các chế độ chơi: chơi với bạn (trên cùng thiết bị), chơi với AI, và chơi online với bạn bè.
- Tích hợp đăng nhập Google, đăng ký, quên mật khẩu thông qua Firebase Authentication.
- Cho phép người dùng kết bạn, tìm kiếm bạn bè, xem và chỉnh sửa thông tin cá nhân.
- Có tính năng chat với bạn bè.
- Có thể chat trực tiếp với nhau trong trận đấu khi ghép online.
- Lưu trữ và hiển thị lịch sử các trận đấu.
- Áp dụng mô hình kiến trúc MVVM nhằm đảm bảo tính linh hoạt và dễ bảo trì.

1.3. Phạm vi nghiên cứu

Nội dung và phạm vi của đồ án bao gồm:

- Thiết kế và phát triển ứng dụng di động chạy trên nền tảng Android bằng Kotlin.
- Xây dựng các chức năng chính: đăng nhập, đăng ký, chơi cờ với nhiều chế độ khác nhau.
- Tích hợp Google Sign-In, Firebase Authentication và Firestore.
- Thiết kế giao diện người dùng với Jetpack Compose.
- Triển khai các thuật toán AI cơ bản để chơi cờ với máy.
- Sử dụng mô hình MVVM để quản lý luồng dữ liệu và giao diện.

Không bao gồm các nền tảng khác ngoài Android (như iOS hay web), cũng như các tính năng nâng cao như phân tích ván cờ chuyên sâu hay hệ thống trò chuyện thời gian thực.

1.4. Phương pháp nghiên cứu

Mô tả các phương pháp và kỹ thuật được sử dụng trong quá trình thực hiện đồ án.

- Nghiên cứu tài liệu: Tìm hiểu các công nghệ sử dụng như Jetpack Compose, Firebase, Firestore, Google Authentication, cùng với các thuật toán AI cơ bản cho cờ vua.
- Phân tích và thiết kế hệ thống: Ứng dụng mô hình MVVM để phân tách các thành phần và xử lý luồng dữ liệu hiệu quả.
- Xây dựng và triển khai: Viết mã bằng Kotlin, thiết kế giao diện bằng Compose, và triển khai các tính năng theo từng bước.
- Kiểm thử và đánh giá: Kiểm tra tính năng và hiệu suất của ứng dụng trên nhiều thiết bị Android khác nhau.
- Tham khảo ứng dụng thực tế: Phân tích các ứng dụng chơi cờ vua phổ biến để học hỏi giao diện và trải nghiệm người dùng.

CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG

2.1. Khảo sát hiện trạng

Hiện nay, trên thị trường có nhiều ứng dụng chơi cờ vua như: Chess.com, lichess.org, Play Magnus,... Các ứng dụng này đều cung cấp các tính năng đa dạng như: chơi với AI, chơi trực tuyến, phân tích ván cờ,... Tuy nhiên, phần lớn các ứng dụng đều được phát triển bởi đội ngũ lớn và mang tính thương mại cao.

Ứng dụng ChessMate được phát triển với mục tiêu phục vụ đa dạng người dùng, từ người mới bắt đầu học chơi cờ vua đến những người chơi có kinh nghiệm muốn luyện tập hoặc thi đấu online. Việc hiểu rõ nhu cầu và hành vi của người dùng là yếu tố then chốt để định hướng thiết kế chức năng, giao diện cũng như trải nghiệm người dùng tổng thể.

2.1.1. Đối tượng người dùng chính

Người mới học chơi cờ: Cần một giao diện đơn giản, trực quan, có AI ở mức độ cơ bản để luyện tập mà không bị “ngợp”.

Người chơi cờ trung bình: Muốn có đối thủ ở các cấp độ khác nhau, chơi với bạn bè online và theo dõi lịch sử các ván cờ để cải thiện kỹ năng.

Người chơi có kinh nghiệm: Quan tâm đến AI mạnh, độ phản hồi nhanh, ít lỗi, có thể phân tích lại ván cờ sau khi chơi.

2.1.2. Nhu cầu và hành vi

Người dùng thường không muốn tạo tài khoản phức tạp, vì vậy việc hỗ trợ đăng nhập bằng Google là một điểm cộng lớn.

Họ thường sử dụng ứng dụng khi có thời gian rảnh, vì thế khả năng vào ván nhanh chóng và dễ dàng là cực kỳ quan trọng.

Người dùng có nhu cầu chơi lại các ván đã chơi, hoặc chia sẻ thành tích với bạn bè, do đó việc lưu lịch sử và cho phép xem lại các nước đi là rất hữu ích.

Nhu cầu chơi cờ với bạn bè cụ thể cũng rất lớn – điều này giúp tăng tính kết nối và giữ chân người dùng để họ tiếp tục sử dụng ứng dụng.

2.1.3. So sánh với các sản phẩm hiện có

Hiện nay các nền tảng như Lichess và Chess.com có giao diện và chức năng rất đầy đủ nhưng thường đòi hỏi đăng ký, tạo tài khoản. ChessMate hướng tới sự đơn giản, gọn nhẹ, dễ dùng, đồng thời cung cấp đủ các tính năng cốt lõi cho trải nghiệm chơi cờ chất lượng.

Với đề tài ChessMate, mục tiêu hướng đến là xây dựng một ứng dụng đơn giản, dễ sử dụng, tích hợp các tính năng cốt lõi như:

- Chơi với bạn (trên cùng thiết bị)
- Chơi với AI
- Chơi online
- Đăng nhập, đăng ký, khôi phục mật khẩu
- Kết bạn, tìm kiếm người chơi
- Lưu lịch sử trận đấu
- Xem và chỉnh sửa thông tin cá nhân

Ứng dụng này hướng đến người dùng phổ thông yêu thích cờ vua và mong muốn trải nghiệm trên thiết bị di động Android.

2.2. Phân tích yêu cầu hệ thống

2.2.1. Yêu cầu chức năng

Tài khoản người dùng:

- Đăng ký, đăng nhập bằng email và mật khẩu
- Đăng nhập bằng Google (Firebase Authentication)
- Quên mật khẩu và gửi email đặt lại mật khẩu

Tính năng bạn bè:

- Tìm kiếm người chơi theo tên người dùng
- Gửi và chấp nhận lời mời kết bạn
- Xem danh sách bạn bè
- Xem thông tin cá nhân của bạn bè
- Chat với bạn bè

Chơi cờ:

- Chế độ chơi với bạn: hai người cùng chơi trên cùng thiết bị
- Chế độ chơi với AI: sử dụng thuật toán để tự động xử lý nước đi
- Chế độ ghép trận online: tìm và ghép người chơi để thi đấu trực tuyến
- Chat trực tiếp với đối thủ khi chọn chế độ ghép trực tiếp

Thông tin người chơi:

- Xem và chỉnh sửa thông tin cá nhân (tên, ảnh đại diện,...)
- Xem thông tin người chơi khác

Lịch sử:

- Lưu lại các ván cờ đã chơi
- Hiện thị lịch sử và chi tiết từng trận đấu

2.2.2. Yêu cầu phi chức năng

- **Tính ổn định:** Ứng dụng hoạt động ổn định trên các thiết bị Android phổ biến.
- **Tính bảo mật:** Sử dụng Firebase Authentication đảm bảo an toàn cho thông tin người dùng.
- **Tính phản hồi:** Giao diện phản hồi nhanh, thao tác mượt mà, không gây giật lag.
- **Khả năng mở rộng:** Có thể dễ dàng thêm các tính năng như: phân tích nước đi, chơi nhiều bàn cùng lúc,...

2.3. Sơ đồ use case hệ thống

Tác nhân (Actor):

- **Người dùng:** Đăng ký, đăng nhập, chơi cờ, kết bạn, xem hồ sơ, chỉnh sửa thông tin, xem lịch sử trận đấu.
- **Hệ thống:** Xử lý xác thực, lưu trữ dữ liệu Firestore, quản lý kết nối online, xử lý logic AI, thông kê trận đấu.

Các use case chính:

- Đăng nhập/đăng ký tài khoản
- Chơi với AI
- Chơi với bạn
- Ghép trận online
- Tìm kiếm và kết bạn
- Xem thông tin cá nhân và chỉnh sửa
- Xem lịch sử các trận đấu

2.4. Kiến trúc hệ thống

Hệ thống được xây dựng dựa trên kiến trúc MVVM (Model – View – ViewModel) nhằm phân tách rõ ràng các thành phần, giúp dễ quản lý và bảo trì:

- **Model:** Xử lý dữ liệu (Firestore, Authentication, thuật toán AI,...)
- **View:** Giao diện người dùng (được xây dựng bằng Jetpack Compose)
- **ViewModel:** Trung gian điều phối logic giữa Model và View, giúp cập nhật dữ liệu và trạng thái UI theo thời gian thực.

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1. Kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình MVVM (Model – View – ViewModel). Đây là mô hình phổ biến trong lập trình Android hiện đại, đặc biệt phù hợp với Jetpack Compose – công cụ UI được Google khuyến nghị cho các ứng dụng Android thế hệ mới.

Model: Chứa các lớp và thành phần xử lý dữ liệu, bao gồm:

- Các lớp dữ liệu (Data Class) như User, Game, FriendRequest, MatchHistory,...
- Các repository để truy xuất dữ liệu từ Firebase Firestore và Firebase Authentication
- Các thuật toán logic.

ViewModel: Trung gian giữa View và Model. Nó chứa logic xử lý, điều hướng dữ liệu từ Model đến View. ViewModel cũng quản lý trạng thái UI thông qua State, LiveData, hoặc StateFlow.

View (UI): Giao diện người dùng được xây dựng bằng Jetpack Compose, phản hồi dữ liệu trạng thái từ ViewModel để hiển thị thông tin phù hợp.

3.2. Thiết kế giao diện người dùng (UI)

3.2.1. Màn hình chính

Gồm các lựa chọn:

- Chơi với bạn
- Chơi với AI
- Chơi online
- Xem lịch sử
- Hồ sơ cá nhân

3.2.2. Màn hình đăng nhập / đăng ký

Người dùng có thể:

- Nhập email và mật khẩu để đăng nhập
- Đăng nhập bằng Google thông qua Firebase Auth
- Chọn "Quên mật khẩu" để khôi phục mật khẩu qua email

3.2.3. Màn hình chơi cờ

- Giao diện bàn cờ 8x8, với các quân cờ có thể tương tác trực tiếp
- Thể hiện nước đi hợp lệ khi chọn một quân cờ
- Có thể phân biệt lượt chơi của người dùng và đối thủ

- Ghi nhận và hiển thị lịch sử các nước đi

3.2.4. Màn hình bạn bè và tìm kiếm

- Danh sách bạn bè kèm ảnh đại diện và tên
- Tìm kiếm bạn bè theo tên người dùng
- Gửi lời mời kết bạn và phản hồi lời mời

3.2.5. Màn hình hồ sơ cá nhân

Hiển thị thông tin: ảnh đại diện, tên, email, số trận thắng/thua, hạng hiện tại. Cho phép người dùng cập nhật thông tin cá nhân

3.2.6. Màn hình lịch sử và chi tiết trận đấu

- Danh sách các trận đã chơi (ngày, đối thủ, kết quả)
- Xem lại toàn bộ ván đấu dưới dạng bảng nước đi hoặc mô phỏng lại bàn cờ

3.3. Thiết kế cơ sở dữ liệu

Hệ thống sử dụng Firebase Firestore – một cơ sở dữ liệu NoSQL dạng tài liệu, linh hoạt, thời gian thực.

Các collection chính:

- **users:** lưu thông tin người dùng

```
createdAt: "09/04/2025"  
description: "Không có mô tả"  
email: "man123@chessmate.com"  
name: "Nhu"  
score: 110  
userId: "d7WtnUcbMSStE9tW31c3wifTtkb2"  
username: "man123"
```

- **friends:** quản lý mối quan hệ bạn bè

```
user1: "Ep8lmzq69BaDyTvQByHTE9dndrk2"  
user2: "ch6rxEqGUeYaT3MaYpBoe2ZE0Xn1"
```

- **friends:** quản lý yêu cầu kết bạn

```
fromName: "Nguyễn Phi Khanh"  
fromUserId: "ZKeZ8Hmfn4b6ln341DGDf2bsnJ12"  
status: "pending"  
timestamp: 10 avril 2025 à 17:22:59 UTC+7  
toUserId: "d7WtnUcbMSStE9tW31c3wifTtkb2"
```

- **matches:** lưu thông tin các trận đấu

```

val matchId = db.collection(collectionPath: "matches").document().id
val initialBoard = createInitialBoard()
val matchData = hashMapOf(
    "matchId" to matchId,
    "player1" to userId,
    "player2" to opponentId,
    "board" to boardToFlatMap(initialBoard),
    "currentTurn" to PieceColor.WHITE.toString(),
    "whiteTime" to 600,
    "blackTime" to 600,
    "status" to "ongoing",
    "winner" to null,
    "drawRequest" to null,
    "lastMove" to null,
    "moveHistory" to emptyList<String>(),
    "fiftyMoveCounter" to 0
)

```

- **rankings:** danh sách xếp hạng người chơi, có thể cập nhật định kỳ hoặc theo sự kiện

3.4. Thiết kế thuật toán chơi với AI

Ứng dụng ChessMate tích hợp một chiến lược trí tuệ nhân tạo (AI) cơ bản để người dùng có thể chơi cờ vua với máy. Thuật toán này được triển khai trong phương thức `findBestMove()` của lớp `ChessViewModel`. Mục tiêu chính của thuật toán là tự động đưa ra nước đi cho quân Đen dựa trên trạng thái hiện tại của bàn cờ, ưu tiên việc bắt quân của đối phương. Dưới đây là phân tích chi tiết về cách thuật toán này hoạt động:

3.4.1. Xác định tất cả các nước đi hợp lệ

Bước đầu tiên trong quá trình ra quyết định của AI là xác định tất cả các nước đi hợp lệ mà quân Đen có thể thực hiện trong lượt hiện tại. Điều này được thực hiện bởi phương thức `getAllMoves(PieceColor.BLACK)`. Phương thức này hoạt động theo các bước sau:

- Duyệt qua tất cả 64 ô trên bàn cờ, từ hàng 0 đến 7 và cột 0 đến 7.
- Tại mỗi ô, thuật toán kiểm tra xem có quân cờ nào đang đứng tại ô đó hay không và liệu quân cờ đó có phải là quân Đen hay không (có màu `PieceColor.BLACK`).
- Nếu tìm thấy một quân Đen, thuật toán sẽ gọi phương thức `game.selectPiece(row, col)` với vị trí hiện tại của quân cờ. Phương thức này (trong lớp `ChessGame`) sẽ trả về một danh sách các đối tượng `Move`, mỗi đối tượng chứa thông tin về một nước đi hợp lệ từ vị trí đã chọn, bao gồm cả vị trí đích (`position`).
- Đối với mỗi nước đi hợp lệ (`Move`), thuật toán sẽ tạo một cặp `Pair<Position, Position>`, với phần tử đầu tiên là vị trí hiện tại của quân cờ (vị trí đi) và phần tử thứ hai là vị trí đích

của nước đi (`move.position`).

- (v) Tất cả các cặp nước đi hợp lệ này sẽ được thêm vào một danh sách (`allMoves`).
- (vi) Cuối cùng, phương thức `getAllMoves()` trả về danh sách đầy đủ tất cả các nước đi hợp lệ mà quân Đen có thể thực hiện.

3.4.2. Đánh giá và lựa chọn nước đi tốt nhất

Sau khi đã có danh sách tất cả các nước đi hợp lệ, thuật toán sẽ tiến hành đánh giá từng nước đi để chọn ra nước đi tốt nhất dựa trên tiêu chí ưu tiên bắt quân của đối phương (quân Trắng). Quá trình này được thực hiện trong phương thức `findBestMove()`:

- (i) Khởi tạo biến theo dõi nước đi tốt nhất: Thuật toán khởi tạo một biến `bestMove` để lưu trữ nước đi tốt nhất tìm được (ban đầu là `null`) và một biến `bestCaptureValue` để lưu trữ giá trị của quân cờ bị bắt trong nước đi tốt nhất (ban đầu là `-1`).
- (ii) Duyệt qua tất cả các nước đi hợp lệ: Thuật toán lặp qua từng cặp nước đi (`move`) trong danh sách `allMoves`. Mỗi cặp `move` chứa vị trí đi (`from`) và vị trí đến (`to`).
- (iii) Kiểm tra khả năng bắt quân: Đối với mỗi nước đi, thuật toán kiểm tra xem tại vị trí đích (`to.row, to.col`) có quân cờ nào của đối phương (quân Trắng, có màu `PieceColor.WHITE`) đang đứng hay không bằng cách gọi phương thức `game.getPieceAt(to.row, to.col)`.
- (iv) Tính giá trị của quân bị bắt (nếu có): Nếu một quân Trắng bị bắt, thuật toán sẽ xác định loại quân đó (`piece.type`) và gán cho nó một giá trị điểm số nhất định:
 - Tốt (`PieceType.PAWN`): 1 điểm
 - Mã (`PieceType.KNIGHT`): 3 điểm
 - Tượng (`PieceType.BISHOP`): 3 điểm
 - Xe (`PieceType.ROOK`): 5 điểm
 - Hậu (`PieceType.QUEEN`): 9 điểm
 - Vua (`PieceType.KING`): 100 điểm
- (v) ****Cập nhật nước đi tốt nhất:**** Nếu giá trị của quân bị bắt (`captureValue`) lớn hơn giá trị tốt nhất đã tìm thấy (`bestCaptureValue`), thuật toán sẽ cập nhật `bestCaptureValue` và lưu trữ nước đi hiện tại (`move`) vào biến `bestMove`.
- (vi) Lựa chọn ngẫu nhiên nếu không có nước đi bắt quân: Nếu sau khi duyệt tất cả các nước đi mà `bestMove` vẫn là `null`, thuật toán sẽ chọn một nước đi ngẫu nhiên từ danh sách `allMoves`.

- (vii) Trả về nước đi tốt nhất: Cuối cùng, phương thức `findBestMove()` trả về cặp `Pair<Position, Position>` đại diện cho nước đi được chọn.

3.4.3. Thực hiện nước đi của AI

Khi phương thức `findBestMove()` trả về một nước đi, phương thức `triggerAIMove()` sẽ thực hiện nước đi đó trên bàn cờ, cập nhật trạng thái trò chơi và xử lý trường hợp phong cấp tốt (nếu có).

3.4.4. Nhận xét về thuật toán

Thuật toán AI được sử dụng trong ChessMate là một chiến lược tham lam cơ bản, tập trung vào việc bắt quân đối phương. Mặc dù đơn giản, nó cung cấp một cơ chế để AI thực hiện các nước đi hợp lệ. Các cải tiến trong tương lai có thể bao gồm việc tích hợp các thuật toán phức tạp hơn để nâng cao khả năng chiến lược của AI.

3.5. Luồng hoạt động chính

3.5.1. Luồng chơi online

- Người dùng bấm "Chơi Online"
- Hệ thống kiểm tra phòng đang chờ
- Nếu có, ghép người chơi vào phòng đó; nếu không, tạo phòng mới
- Hai người chơi thi đấu theo lượt
- Khi ván đấu kết thúc, hệ thống cập nhật kết quả, lịch sử trận đấu, và điểm xếp hạng

3.5.2. Luồng thêm bạn

- Người dùng tìm kiếm tên người chơi
- Gửi lời mời kết bạn
- Người nhận xác nhận lời mời
- Hệ thống cập nhật mối quan hệ trong collection friends

CHƯƠNG 4: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết quả đạt được

Sau một quá trình tìm tòi, nghiên cứu, thiết kế và triển khai, ứng dụng ChessMate đã được xây dựng thành công và hoàn thiện các chức năng chính như sau:

Chức năng xác thực người dùng:

- Đăng nhập bằng tài khoản email và mật khẩu thông qua Firebase Authentication.
- Đăng nhập nhanh bằng tài khoản Google (Google Sign-In).
- Đăng ký tài khoản mới và xác minh đầu vào cơ bản.
- Chức năng “Quên mật khẩu” giúp người dùng khôi phục mật khẩu thông qua email.

Chức năng chơi cờ:

- Chơi với bạn bè: Hai người cùng chơi trực tiếp trên một thiết bị.
- Chơi với AI: Ứng dụng tích hợp thuật toán để xây dựng AI có khả năng chơi cờ với nhiều mức độ khó khác nhau.
- Chơi online: Người dùng có thể ghép trận ngẫu nhiên và chơi cờ trực tuyến với người chơi khác.
- Chức năng chat với đối thủ khi chơi ghép trực tuyến.

Chức năng mạng xã hội:

- Gửi lời mời kết bạn, chấp nhận, từ chối lời mời.
- Danh sách bạn bè có thể xem thông tin, gửi lời mời chơi.
- Tìm kiếm người chơi theo tên.
- Chức năng chat với bạn bè.

Chức năng quản lý người dùng:

- Xem và chỉnh sửa thông tin cá nhân (tên hiển thị, mô tả...).
- Xem thông tin chi tiết của bạn bè và đối thủ đã từng chơi cùng.

Lịch sử và thống kê:

- Lưu trữ lịch sử các ván đấu, bao gồm thời gian, kết quả, và các nước đi.
- Hiển thị lại bàn cờ và diễn biến trận đấu.

Mô hình kiến trúc MVVM: Ứng dụng được xây dựng theo mô hình MVVM (Model - View - ViewModel) giúp phân tách rõ ràng giữa giao diện, logic xử lý và dữ liệu, tăng khả năng bảo trì, mở rộng và tái sử dụng code.

Công nghệ sử dụng:

- **Ngôn ngữ lập trình:** Kotlin
- **Giao diện:** Jetpack Compose – giúp viết UI bằng code, dễ tùy biến và hiện đại.
- **Cơ sở dữ liệu:** Firestore – lưu trữ thông tin người dùng, bạn bè, lịch sử trận đấu.
- **Xác thực:** Firebase Authentication

4.2. Hạn chế

Dù ứng dụng đã đạt được nhiều kết quả tích cực, vẫn còn một số điểm hạn chế cần cải thiện:

- Cần cải thiện hệ thống đánh giá AI để hỗ trợ nhiều cấp độ khó khăn hơn:
 - Hệ thống đánh giá AI hiện tại mới chỉ được xây dựng dựa trên một thuật toán cơ bản, tập trung chủ yếu vào việc tối ưu hóa lợi thế vật chất bằng cách ưu tiên các nước đi bắt quân của đối phương. Mặc dù cách tiếp cận này cho phép AI đưa ra các nước đi hợp lệ và tạo ra một đối thủ ở mức độ nhập môn, nhưng nó tồn tại nhiều hạn chế.
 - Thuật toán hiện tại thiếu khả năng xem xét các yếu tố chiến lược phức tạp và sâu sắc hơn, vốn là nền tảng của một ván cờ vua ở trình độ cao. Các yếu tố như kiểm soát trung tâm bàn cờ, sự phát triển hài hòa của các quân cờ, mức độ an toàn của Vua, cấu trúc tốt vững chắc, và khả năng tạo ra các mối đe dọa chiếu hết tiềm năng hiện chưa được đánh giá một cách đầy đủ.
 - Do sự đơn giản trong logic đánh giá, AI hiện tại dễ dàng bị đánh bại bởi những người chơi có kinh nghiệm, những người có khả năng nhìn nhận ván cờ ở một tầm chiến lược cao hơn và có thể khai thác những điểm yếu trong cách AI lựa chọn nước đi. AI thiếu khả năng lập kế hoạch dài hạn và không có khả năng đánh giá toàn diện các vị trí phức tạp trên bàn cờ.
- Tính năng chơi online đôi khi bị chậm do phụ thuộc vào tốc độ kết nối mạng và quá trình đồng bộ Firestore.
- Giao diện chưa tối ưu hoàn toàn cho tất cả kích thước màn hình.
- Chưa có tính năng sửa ảnh avatar.
- Chưa có tính năng xếp hạng người chơi dựa trên số trận thắng/thua và thời gian hoạt động.
- Bổ sung hiệu ứng âm thanh tương tác trong quá trình chơi và di chuyển quân cờ:
 - Tích hợp các hiệu ứng âm thanh sống động và phù hợp với từng hành động trong trò chơi, chẳng hạn như âm thanh khi người chơi thực hiện một nước đi, âm thanh khi quân cờ bị bắt, hoặc các hiệu ứng âm thanh đặc biệt cho các sự kiện quan trọng như chiếu tướng hoặc hết cờ.

- Cung cấp tùy chọn cho người dùng điều chỉnh âm lượng hoặc tắt/bật hiệu ứng âm thanh theo sở thích cá nhân, tạo sự thoải mái và tùy biến trong trải nghiệm chơi.
- Nghiên cứu và lựa chọn các hiệu ứng âm thanh chất lượng cao, đảm bảo tính thẩm mỹ và không gây xao nhãng cho người chơi, góp phần tăng thêm sự hấp dẫn và tính chân thực cho ứng dụng.

4.3. Hướng phát triển

Trong thời gian tới, ứng dụng ChessMate có thể được nâng cấp với các tính năng mới và tối ưu như sau:

Cải tiến hiệu năng:

- Tối ưu tốc độ load dữ liệu từ Firestore.
- Sử dụng Cloud Functions để xử lý một số logic phía server.
- Sử dụng Firebase Storage để lưu trữ ảnh .

Tính năng mở rộng:

- Hệ thống thành tích và huy hiệu cho người chơi:
 - Phát triển một hệ thống thành tích đa dạng, ghi nhận những cột mốc quan trọng và những chiến thắng ấn tượng của người chơi trong quá trình sử dụng ứng dụng.
 - Thiết kế các huy hiệu độc đáo và có giá trị tượng trưng khác nhau, trao thưởng cho người chơi khi họ đạt được các thành tích nhất định, khuyến khích sự tham gia tích cực và tạo động lực để người chơi tiếp tục rèn luyện và nâng cao kỹ năng.
 - Cho phép người chơi hiển thị các thành tích và huy hiệu đã đạt được trên hồ sơ cá nhân, tạo điểm nhấn và thể hiện đẳng cấp của họ trong cộng đồng người chơi ChessMate.
- Thêm nhiều cấp độ AI và nâng cao trí tuệ nhân tạo bằng các kỹ thuật Machine Learning:
 - Mở rộng số lượng cấp độ khó của AI, từ cơ bản đến nâng cao, nhằm đáp ứng nhu cầu luyện tập của người chơi ở mọi trình độ.
 - Thay thế thuật toán AI hiện tại bằng các thuật toán tìm kiếm cây trò chơi tiên tiến như **Minimax** và **Alpha-Beta Pruning** để cải thiện khả năng dự đoán và lựa chọn nước đi của AI. Các thuật toán này sẽ cho phép AI "nhìn xa" hơn trong các tình huống trên bàn cờ và đưa ra các quyết định chiến lược phức tạp hơn.
 - Xây dựng một **Hàm đánh giá (Evaluation Function)** tinh vi, có khả năng định lượng giá trị của một trạng thái bàn cờ dựa trên nhiều yếu tố chiến lược quan trọng

như giá trị vật chất, kiểm soát trung tâm, sự phát triển quân, cấu trúc tốt, an toàn của Vua và tính di động của quân. Việc tối ưu hóa hàm đánh giá sẽ giúp AI đưa ra những nhận định chính xác hơn về tình hình ván cờ.

- Nghiên cứu và ứng dụng các kỹ thuật **Học máy (Machine Learning)**, đặc biệt là việc huấn luyện **mạng nơ-ron sâu (Deep Neural Network)** trên một lượng lớn dữ liệu các ván cờ chất lượng cao. Mục tiêu là giúp AI học hỏi các mẫu hình chiến lược phức tạp và đưa ra những nước đi tối ưu dựa trên kinh nghiệm đã học được, vượt xa khả năng được lập trình thủ công. Việc tích hợp học máy có tiềm năng tạo ra một AI không chỉ mạnh mẽ mà còn có khả năng thích ứng với phong cách chơi của từng người dùng.

Tích hợp dịch vụ bên ngoài:

- Tích hợp Firebase Cloud Messaging để gửi thông báo khi có lời mời kết bạn, thách đấu...
- Đồng bộ tài khoản và xếp hạng qua Google Play Games.

4.4. Kết quả học tập đạt được

Thông qua quá trình thực hiện đồ án, nhóm đã học được và củng cố nhiều kiến thức, bao gồm:

- Thành thạo hơn trong sử dụng Kotlin và Jetpack Compose để phát triển ứng dụng Android hiện đại.
- Hiểu rõ về mô hình kiến trúc MVVM trong thực tế.
- Biết cách tích hợp Firebase (Authentication, Firestore, Storage) vào ứng dụng Android.
- Có thêm kinh nghiệm về thuật toán AI và xử lý logic cờ vua.
- Rèn luyện kỹ năng làm việc nhóm, chia nhỏ task và tổ chức tiến độ hợp lý.

KẾT LUẬN

Sau một khoảng thời gian tìm hiểu, nghiên cứu và thực hiện, nhóm đã hoàn thành ứng dụng ChessMate – một ứng dụng chơi cờ vua hiện đại, tiện lợi trên nền tảng Android. Ứng dụng được xây dựng bằng ngôn ngữ Kotlin, sử dụng Jetpack Compose cho giao diện và Firebase làm nền tảng backend, đáp ứng đầy đủ các chức năng như: chơi với bạn, chơi với AI, ghép trận online, quản lý tài khoản, bạn bè, và lưu lịch sử trận đấu.

Thông qua đồ án này, nhóm không chỉ củng cố kiến thức về lập trình Android và mô hình kiến trúc MVVM mà còn học được cách giải quyết vấn đề thực tế, thiết kế logic hợp lý và phối hợp làm việc nhóm hiệu quả. Đặc biệt, việc xây dựng AI để chơi cờ là một thử thách thú vị và giúp nhóm hiểu sâu hơn về thuật toán Minimax và cách tối ưu nó.

Tuy còn một số hạn chế cần cải thiện, nhưng kết quả hiện tại là bước đệm vững chắc để nhóm có thể tiếp tục nâng cấp và phát triển ứng dụng trong tương lai. Đây cũng là một trải nghiệm học tập rất đáng nhớ và hữu ích trong hành trình trở thành lập trình viên chuyên nghiệp.

Nhóm xin chân thành cảm ơn thầy Trương Quang Tuấn đã tận tình hướng dẫn, góp ý trong suốt quá trình thực hiện đồ án này.

TÀI LIỆU THAM KHẢO

Tài liệu

- [1] Android Developers. *Kotlin on Android*. <https://developer.android.com/kotlin> (Truy cập ngày 15 tháng 3 năm 2025).
- [2] Android Developers. *Jetpack Compose*. <https://developer.android.com/jetpack/compose> (Truy cập ngày 18 tháng 3 năm 2025).
- [3] Firebase Documentation. <https://firebase.google.com/docs> (Truy cập ngày 25 tháng 3 năm 2025).
- [4] Google Developers. *Guide to app architecture*. <https://developer.android.com/topic/architecture> (Truy cập ngày 01 tháng 4 năm 2025).
- [5] FIDE. *Laws of Chess*. <https://handbook.fide.com/chapter/E012023> (Truy cập ngày 10 tháng 4 năm 2025).