

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!

LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển mạnh mẽ, các ứng dụng di động ngày càng trở nên phổ biến và đóng vai trò quan trọng trong đời sống hàng ngày. Việc kết hợp giữa công nghệ và các trò chơi truyền thống như cờ vua đã tạo ra nhiều cơ hội để phát triển những ứng dụng giải trí vừa mang tính trí tuệ, vừa tiện lợi và hấp dẫn.

Đồ án “Ứng dụng ChessMate – Chơi cờ vua trên Android” được xây dựng nhằm cung cấp một nền tảng chơi cờ linh hoạt, hỗ trợ nhiều chế độ chơi như chơi với bạn bè, chơi với AI, chơi online, cùng với các tính năng quản lý tài khoản, kết bạn, và xem lịch sử trận đấu.

Đồ án được thực hiện bằng ngôn ngữ Kotlin, sử dụng Jetpack Compose để thiết kế giao diện, và Firestore làm nền tảng dữ liệu. Đồng thời, mô hình MVVM được áp dụng nhằm đảm bảo khả năng mở rộng, bảo trì và phát triển ứng dụng một cách dễ dàng.

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, em quyết định chọn đề tài này để vừa học hỏi, vừa ứng dụng 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.
- Lưu trữ và hiển thị lịch sử các trận đấu, xếp hạng người chơi.
- 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 (Minimax, Alpha-Beta Pruning).
- 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, xếp hạng người chơi, 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.

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
- Xếp hạng người chơi

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

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

Thông tin người chơi:

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

Lịch sử và xếp hạng:

- Lưu lại các ván cờ đã chơi
- Hiển thị lịch sử và chi tiết từng trận đấu
- Xếp hạng người chơi dựa trên số trận thắng/thua

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: chat, 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, xếp hạng.
- **Hệ thống:** Xử lý xác thực, lưu trữ dữ liệu Firestore, quản lý kết nối online, xử lý logic AI, thống 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
- Xếp hạng người chơi

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 như AI Minimax

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
- Xếp hạng

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 (có màu hoặc nhấn sáng)
- 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 sử dụng thuật toán Minimax kết hợp với Alpha-Beta pruning để xây dựng AI chơi cờ:

Minimax: Thuật toán tìm kiếm theo chiều sâu, mô phỏng các nước đi và phản ứng của đối thủ, từ đó chọn nước đi tối ưu.

Alpha-Beta pruning: Cải tiến của Minimax, giúp cắt bớt các nhánh không cần thiết, tăng tốc độ xử lý.

Độ sâu (depth) của thuật toán được điều chỉnh theo độ khó của người chơi lựa chọn.

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

3.6. Sơ đồ chức năng (Use Case Diagram)