**BỘ CÔNG THƯƠNG**



**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞

**BÁO CÁO MÔN LẬP TRÌNH DI ĐỘNG**

**ĐỀ TÀI: ỨNG DỤNG ĐẶT SÂN THỂ THAO**

Giáo viên hướng dẫn: Trần Xuân Thanh Phúc

Sinh viên thực hiện:

1. Nguyễn Tất Ngọc
2. Nguyễn Hồng Huy Bảo
3. Trần Chí Cương

TP. Hồ Chí Minh - 2025

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **HỌ TÊN** | **CÔNG VIỆC ĐƯỢC GIAO** | **ĐÁNH GIÁ** |
| 1 | 2001223103 | Nguyễn Tất Ngọc | Đăng nhập, Đăng ký, quản lý admin, danh sách và chi tiết sân, thông báo | 100% |
| 2 | 2001220303 | Nguyễn Hồng Huy Bảo | Đổi thông tin người dùng, tìm kiếm, quên mật khẩu, lọc theo loại sân | 100% |
| 3 | 2001220440 | Trần Chí Cương | Đặt sân, hủy sân, API Gmail xác nhận, CSDL Firebase, đánh giá | 100% |

# MỤC LỤC

[**BẢNG PHÂN CÔNG CÔNG VIỆC 2**](#_Toc200499831)

[**MỤC LỤC 3**](#_Toc200499832)

[**LỜI MỞ ĐẦU 4**](#_Toc200499833)

[**CHƯƠNG 1: GIỚI THIỆU 5**](#_Toc200499834)

[**1.1 Đặt vấn đề 5**](#_Toc200499835)

[**1.2 Mục tiêu đồ án 5**](#_Toc200499836)

[**1.3 Phạm vi nghiên cứu 6**](#_Toc200499837)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 7**](#_Toc200499838)

[**2.1 Flutter 7**](#_Toc200499839)

[**2.2 Firebase 7**](#_Toc200499840)

[**2.3 Provider – Quản lý trạng thái trong Flutter 8**](#_Toc200499841)

[**2.4 Mô hình MVC/MVVM (áp dụng gián tiếp) 8**](#_Toc200499842)

[**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 9**](#_Toc200499843)

[**3.1 Khảo sát yêu cầu hệ thống 9**](#_Toc200499844)

[**3.2 Sơ đồ UseCase 9**](#_Toc200499845)

[**3.3 Thiết kế cơ sở dữ liệu (Firestore) 10**](#_Toc200499846)

[**3.4 Thiết kế giao diện (UI) 11**](#_Toc200499847)

[**3.5 Kiến trúc hệ thống 20**](#_Toc200499848)

[**CHƯƠNG 4: CÀI ĐẶT VÀ KẾT QUẢ THỰC NGHIỆM 21**](#_Toc200499849)

[**4.1 Công cụ và môi trường phát triển 21**](#_Toc200499850)

[**4.2 Một số chức năng chính đã triển khai 21**](#_Toc200499851)

[**4.3.1 Đăng ký và đăng nhập người dùng 21**](#_Toc200499852)

[**4.3.2 Xem danh sách và chi tiết sân 21**](#_Toc200499853)

[**4.3.3 Đặt sân 21**](#_Toc200499854)

[**4.3.4 Quản lý sân (Admin) 22**](#_Toc200499855)

[**4.3.5 Đánh giá sân 22**](#_Toc200499856)

[**4.3 Kết quả thực nghiệm 22**](#_Toc200499857)

[**4.4.1 Giao diện người dùng 22**](#_Toc200499858)

[**4.4.2 Tính ổn định 22**](#_Toc200499859)

[**4.4.3 Kiểm thử 22**](#_Toc200499860)

[**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 23**](#_Toc200499861)

[**5.1 Kết luận 23**](#_Toc200499862)

[**5.2 Hạn chế 23**](#_Toc200499863)

[**5.3 Hướng phát triển 24**](#_Toc200499864)

[**TÀI LIỆU THAM KHẢO 25**](#_Toc200499865)

# LỜI MỞ ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, nhu cầu sử dụng các ứng dụng di động để hỗ trợ các hoạt động thường nhật ngày càng trở nên phổ biến. Một trong những nhu cầu thiết thực hiện nay là việc **đặt sân thể thao trực tuyến**, giúp người dùng tiết kiệm thời gian và thuận tiện hơn trong việc tìm kiếm, so sánh và đặt sân theo nhu cầu.

Nhận thấy tiềm năng ứng dụng thực tiễn của lĩnh vực này, nhóm chúng em đã lựa chọn đề tài “**Ứng dụng đặt sân thể thao sử dụng Flutter và Firebase**” làm đồ án học phần. Ứng dụng được phát triển bằng **Flutter** – một framework đa nền tảng do Google phát triển – kết hợp với **Firebase** để quản lý người dùng, lưu trữ dữ liệu và xử lý các chức năng liên quan đến đăng nhập, đặt sân, đánh giá, và quản lý thông tin sân.

Quá trình thực hiện đồ án không chỉ giúp chúng em vận dụng kiến thức đã học như lập trình hướng đối tượng, thiết kế giao diện người dùng, làm việc với cơ sở dữ liệu và quản lý trạng thái ứng dụng mà còn giúp trau dồi kỹ năng làm việc nhóm, tư duy thiết kế hệ thống và giải quyết vấn đề thực tiễn.

# CHƯƠNG 1: GIỚI THIỆU

## 1.1 Đặt vấn đề

Trong bối cảnh cuộc sống hiện đại ngày càng bận rộn, việc tìm kiếm và đặt chỗ cho các hoạt động thể thao như bóng đá, bóng chuyền, cầu lông,… theo cách truyền thống (gọi điện, đến trực tiếp) thường tốn thời gian và thiếu hiệu quả. Việc quản lý sân bãi, giờ thi đấu, đặt chỗ và thanh toán cũng gặp nhiều bất cập nếu không có sự hỗ trợ từ công nghệ.

Cùng với sự phát triển mạnh mẽ của công nghệ di động, việc xây dựng một **ứng dụng đặt sân thể thao trực tuyến** không chỉ đáp ứng nhu cầu thực tế của người dùng mà còn mang lại giải pháp hiệu quả cho các chủ sân trong việc quản lý hoạt động kinh doanh của mình.

Đề tài đồ án “**Ứng dụng đặt sân thể thao sử dụng Flutter và Firebase**” được thực hiện với mục tiêu tạo ra một ứng dụng thân thiện, tiện lợi cho người dùng trong việc đặt sân, đồng thời hỗ trợ chủ sân trong công tác quản lý, cập nhật sân bãi và theo dõi các lượt đặt sân.

## **1.2 Mục tiêu đồ án**

* Xây dựng ứng dụng di động đa nền tảng (Android/iOS) sử dụng **Flutter**.
* Tích hợp **Firebase** để:
  + Quản lý đăng nhập, đăng ký người dùng.
  + Lưu trữ dữ liệu sân, lịch đặt, đánh giá.
* Phát triển các chức năng chính như:
  + Tìm kiếm và xem thông tin sân thể thao.
  + Đặt lịch chơi thể thao theo khung giờ.
  + Quản lý người dùng (Admin/User).
  + Chức năng đánh giá, phản hồi về sân.
* Tạo giao diện trực quan, dễ sử dụng.
* Phân quyền người dùng: người quản trị (admin) và người dùng thường.

## 1.3 Phạm vi nghiên cứu

* **Nền tảng sử dụng**: Flutter cho phần giao diện người dùng (UI), Firebase cho backend (Authentication, Firestore, Storage).
* **Đối tượng sử dụng**:
  + **Người dùng thường**: có thể tìm kiếm sân, đặt sân, đánh giá.
  + **Người quản trị (Admin)**: quản lý danh sách sân, xem và xử lý các đơn đặt sân.
* **Thiết bị triển khai**: Ứng dụng chủ yếu chạy trên điện thoại di động (Android là chính, iOS nếu cần có thể mở rộng).
* **Chưa triển khai**:
  + Tích hợp thanh toán online.
  + Xác thực đa yếu tố hoặc bảo mật nâng cao.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Flutter

Flutter là một **framework phát triển ứng dụng giao diện người dùng đa nền tảng** (cross-platform UI toolkit) được phát triển bởi Google. Flutter cho phép lập trình viên xây dựng các ứng dụng chạy trên Android, iOS, web và desktop chỉ từ một **codebase duy nhất**.

**Các đặc điểm nổi bật của Flutter:**

* **Ngôn ngữ lập trình Dart**: Flutter sử dụng Dart – ngôn ngữ do Google phát triển – với cú pháp dễ hiểu, hỗ trợ lập trình hướng đối tượng.
* **Hot Reload**: Cho phép cập nhật giao diện ngay lập tức mà không cần chạy lại toàn bộ ứng dụng.
* **Widgets**: Mọi thành phần giao diện trong Flutter đều là widget, từ đơn giản (Text, Button) đến phức tạp (ListView, Scaffold).
* **Hiệu năng cao**: Flutter sử dụng bộ render riêng thay vì dùng native UI components, mang lại hiệu năng tốt.

## 2.2 Firebase

Firebase là nền tảng phát triển ứng dụng do Google cung cấp, hỗ trợ lập trình viên phát triển và mở rộng các ứng dụng di động/web một cách nhanh chóng. Firebase cung cấp nhiều dịch vụ tiện ích như:

* **Firebase Authentication**: Hệ thống xác thực người dùng thông qua email, Google, Facebook, v.v.
* **Cloud Firestore**: Cơ sở dữ liệu NoSQL realtime, phù hợp với ứng dụng có nhiều tương tác như đặt sân, đánh giá, quản lý người dùng.
* **Firebase Storage**: Lưu trữ hình ảnh sân, avatar người dùng...
* **Firebase Hosting & Cloud Functions** *(nếu dùng)*: Triển khai backend hoặc website tĩnh.

Firebase giúp giảm thời gian phát triển backend, dễ tích hợp vào Flutter thông qua các package như firebase\_core, cloud\_firestore, firebase\_auth, v.v.

## 2.3 Provider – Quản lý trạng thái trong Flutter

**Provider** là một trong những package phổ biến nhất trong Flutter dùng để quản lý trạng thái. Trong các ứng dụng có nhiều tương tác giữa các widget, việc chia sẻ và cập nhật trạng thái (dữ liệu) một cách hợp lý là rất quan trọng.

**Một số khái niệm:**

* **ChangeNotifier**: Một lớp trong Flutter giúp phát ra sự thay đổi khi có dữ liệu mới.
* **Consumer**: Widget dùng để lắng nghe và phản ứng khi dữ liệu thay đổi.
* **ProviderScope / MultiProvider**: Cấu trúc bọc quanh toàn ứng dụng để cung cấp trạng thái cho nhiều nơi.

Provider giúp tách logic xử lý dữ liệu khỏi UI, giữ cho mã nguồn rõ ràng, dễ bảo trì.

## 2.4 Mô hình MVC/MVVM (áp dụng gián tiếp)

Mặc dù Flutter không bắt buộc sử dụng một kiến trúc cụ thể, nhưng việc tổ chức mã nguồn theo mô hình như **MVC** (Model – View – Controller) hoặc **MVVM** (Model – View – ViewModel) giúp tăng khả năng mở rộng và tái sử dụng.

* **Model**: Chứa dữ liệu, ví dụ: UserModel, FieldModel, BookingModel.
* **View**: Các widget giao diện như LoginScreen, BookingScreen, v.v.
* **Provider (Controller/ViewModel)**: Quản lý trạng thái, điều phối luồng dữ liệu giữa Model và View.

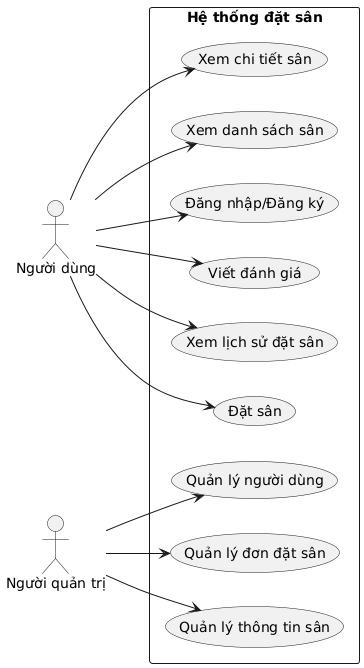
# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1 Khảo sát yêu cầu hệ thống

**Người dùng hệ thống gồm hai loại:**

* **Người dùng thông thường (User):**
  + Đăng ký, đăng nhập.
  + Xem danh sách các sân thể thao.
  + Đặt sân theo ngày giờ.
  + Đánh giá, nhận xét về sân sau khi sử dụng.
* **Người quản trị (Admin):**
  + Quản lý thông tin sân (thêm, sửa, xoá).
  + Xem danh sách các đơn đặt sân.
  + Quản lý người dùng và thời gian hoạt động của sân.

## 3.2 Sơ đồ UseCase



## 3.3 Thiết kế cơ sở dữ liệu (Firestore)

Dựa trên Firebase Firestore (NoSQL), dữ liệu được lưu theo dạng collection/document. Một số collections chính như:

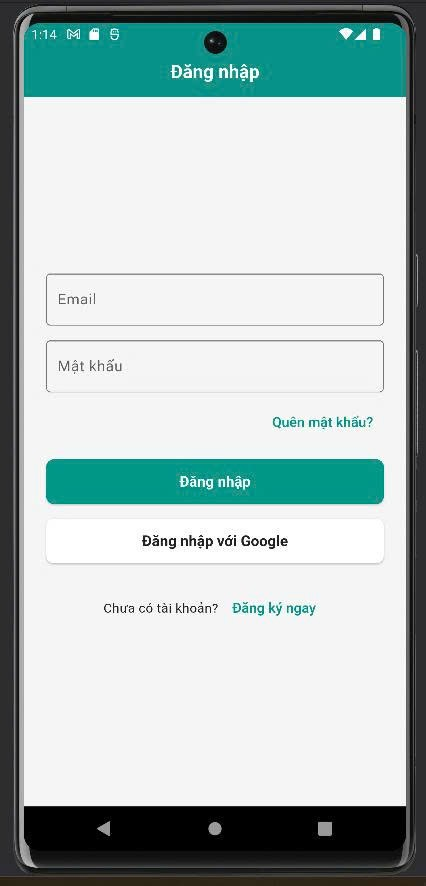
| **Collection** | **Mô tả** |
| --- | --- |
| **users** | Lưu thông tin người dùng: uid, email, họ tên, vai trò |
| **fields** | Lưu thông tin các sân: tên sân, loại sân, hình ảnh, địa chỉ, giá |
| **bookings** | Thông tin đặt sân: id người đặt, id sân, thời gian, trạng thái |
| **reviews** | Đánh giá: user\_id, field\_id, nội dung, số sao |
| **time\_slots** | Khung giờ hoạt động của từng sân |

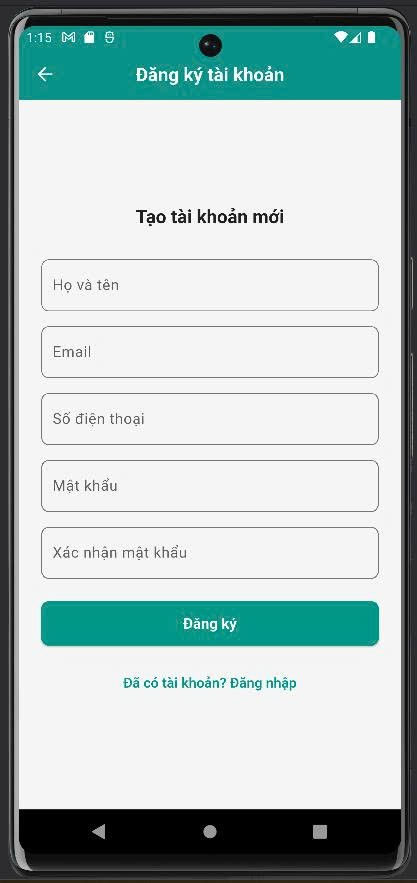
## 3.4 Thiết kế giao diện (UI)

Ứng dụng được thiết kế bằng Flutter với các màn hình chính:

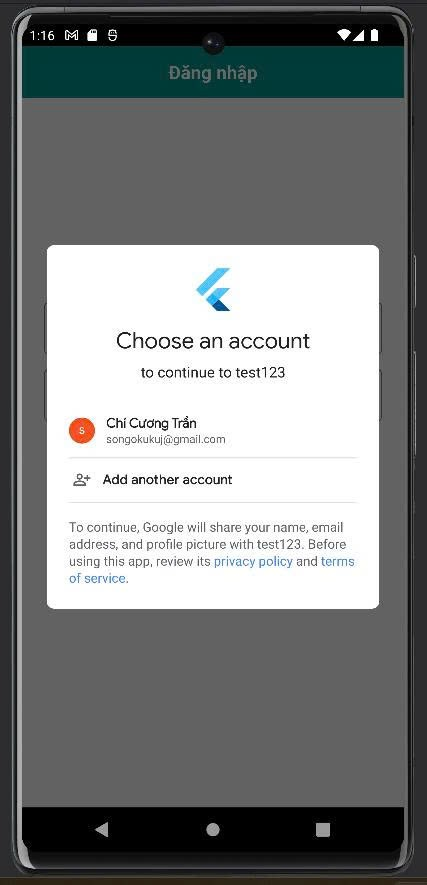
**Đối với người dùng:**

* Trang đăng nhập / đăng ký

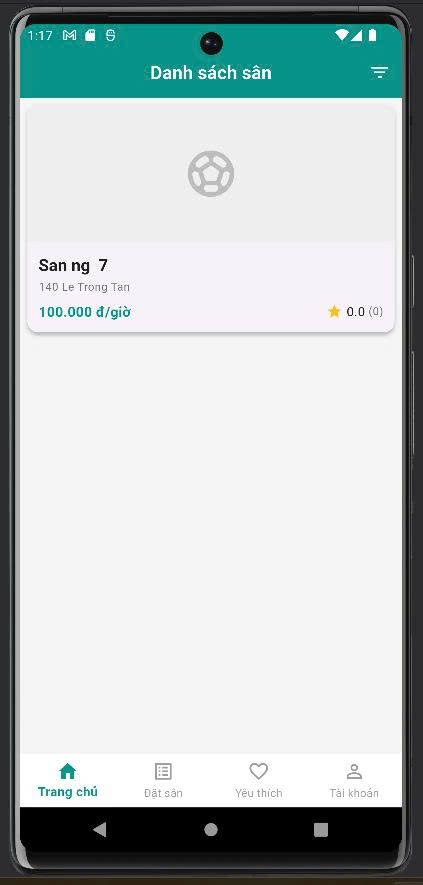




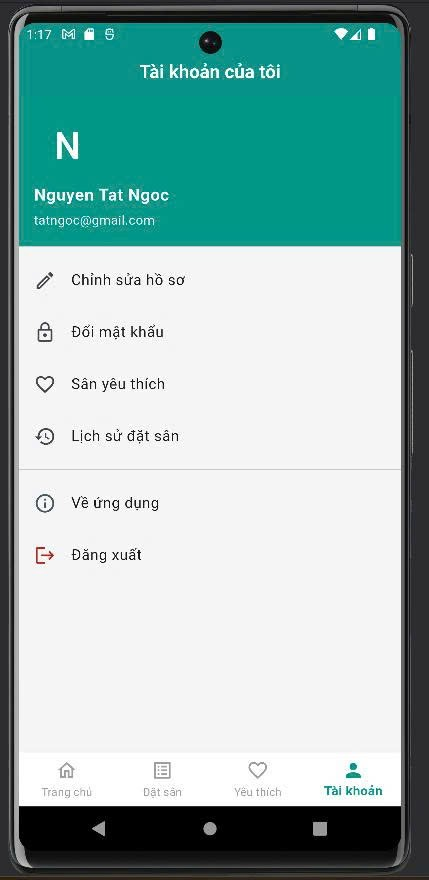


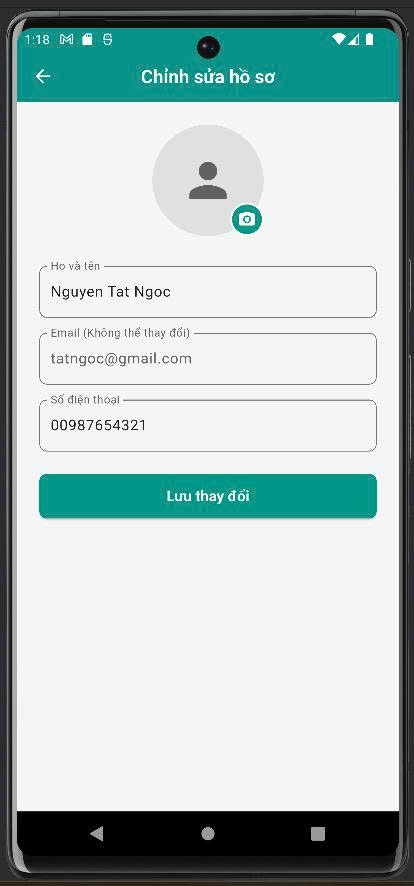


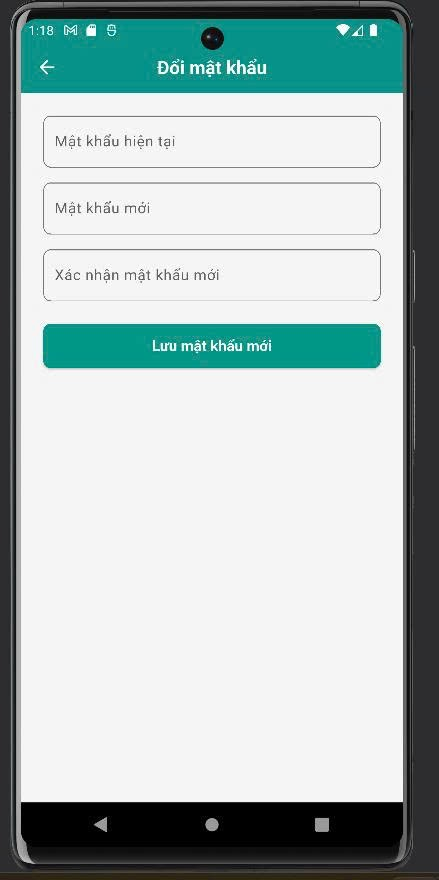
* Trang danh sách sân



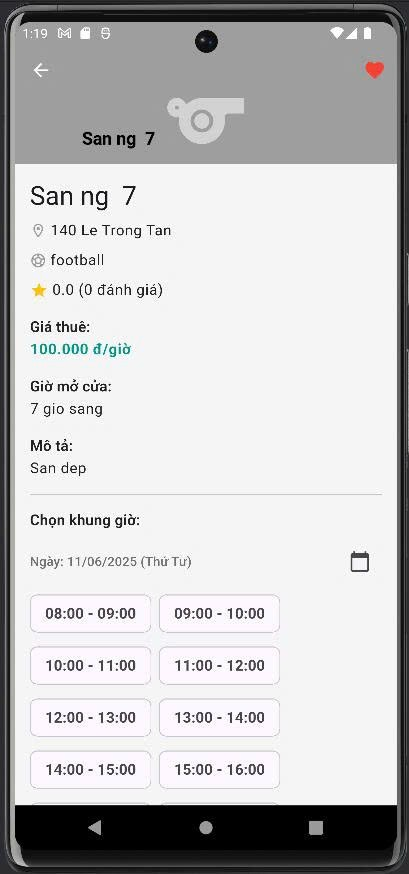
* Thông tin tài khoản

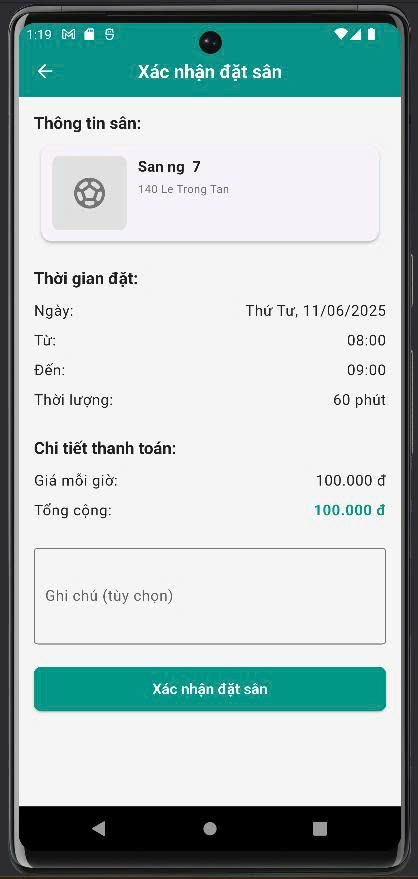




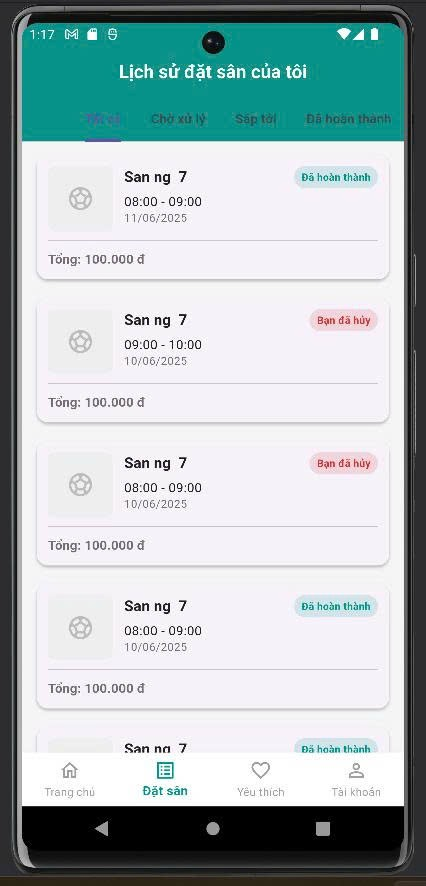


* Trang đặt sân

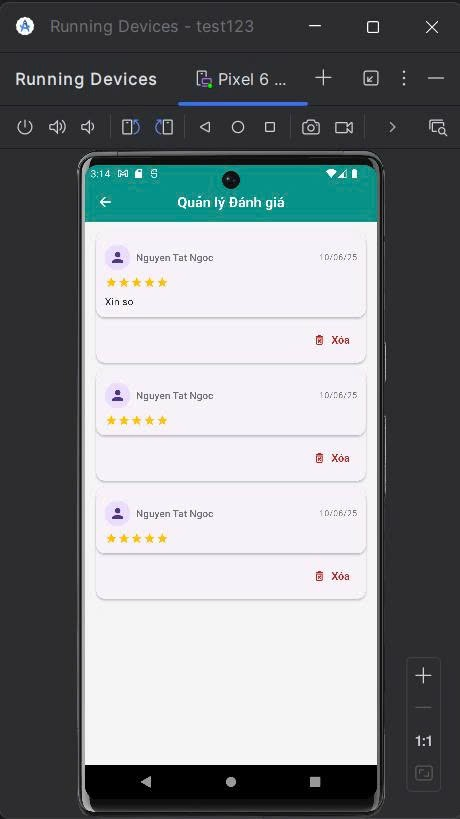




* Lịch sử đặt sân

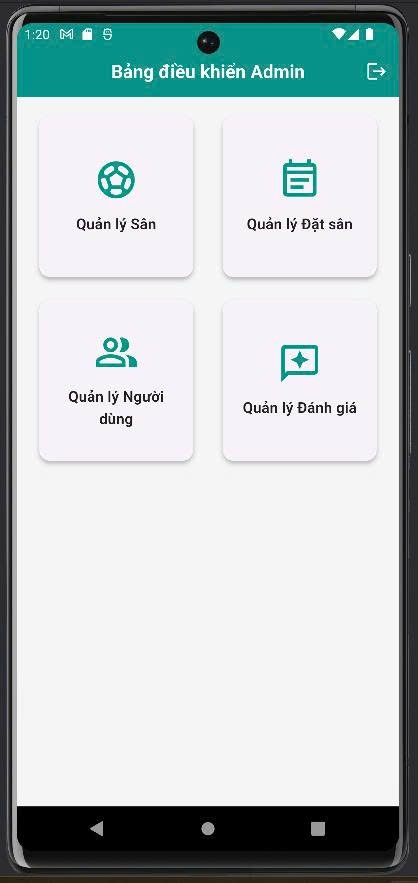


* Trang đánh giá

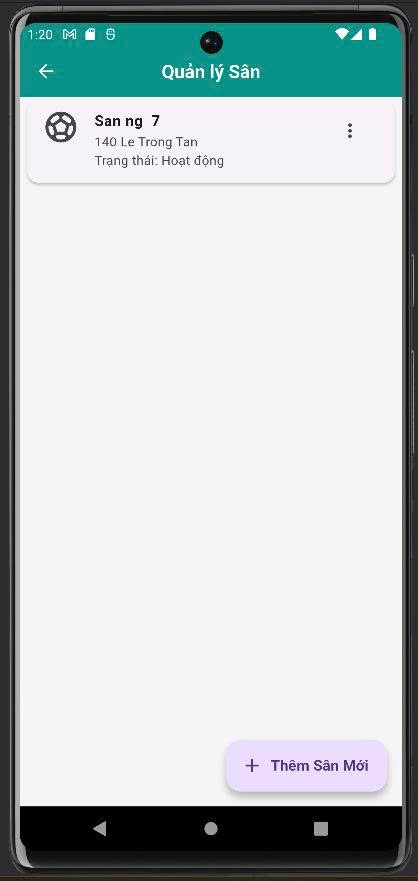


**Đối với admin:**

* Dashboard quản trị

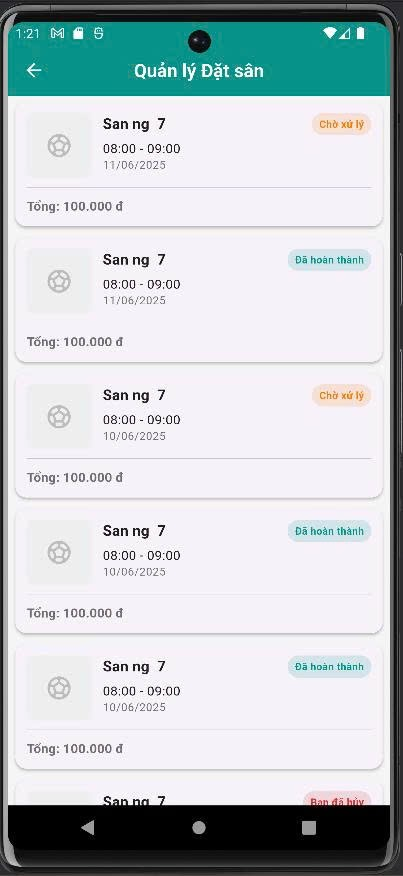


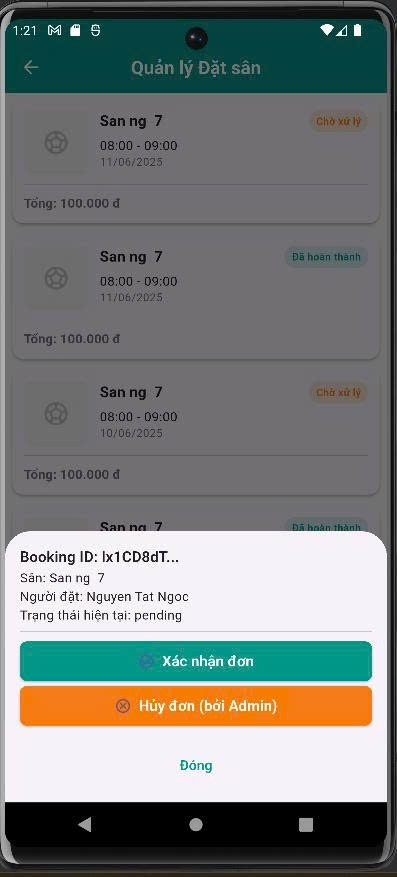
* Trang quản lý sân



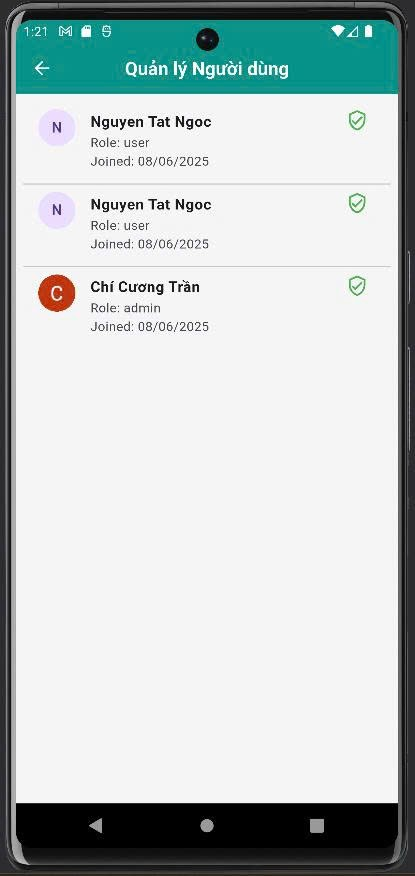


* Quản lý đơn đặt sân





* Quản lý người dùng



## 3.5 Kiến trúc hệ thống

Hệ thống được tổ chức theo mô hình **phân tầng**, gồm:

* **Giao diện người dùng (UI)**: Flutter Widgets.
* **Quản lý trạng thái (Provider)**: Xử lý luồng dữ liệu giữa UI và dữ liệu.
* **Dữ liệu (Model)**: Các class như FieldModel, BookingModel, UserModel, v.v.
* **Dịch vụ Firebase**: Đăng nhập, lưu trữ dữ liệu, tải ảnh, đồng bộ thời gian thực.

# CHƯƠNG 4: CÀI ĐẶT VÀ KẾT QUẢ THỰC NGHIỆM

## **4.1** Công cụ và môi trường phát triển

| **Thành phần** | **Công cụ/Thông tin** |
| --- | --- |
| Ngôn ngữ | Dart |
| Framework | Flutter SDK (>= 3.x) |
| IDE | Android Studio / Visual Studio Code |
| Backend | Firebase (Firestore, Auth, Storage) |
| Quản lý trạng thái | Provider |
| Thiết bị kiểm thử | Giả lập Android (Pixel), thiết bị thật (Samsung A12, v.v.) |

## **4.2** Một số chức năng chính đã triển khai

### 4.3.1 Đăng ký và đăng nhập người dùng

* Người dùng đăng nhập bằng email & password.
* Xác thực qua FirebaseAuth.
* Kiểm tra vai trò (admin/user) để điều hướng giao diện phù hợp.

### 4.3.2 Xem danh sách và chi tiết sân

* Hiển thị danh sách sân từ Firestore.
* Mỗi sân có tên, ảnh, giá, địa điểm, đánh giá trung bình.
* Xem chi tiết sân và chọn khung giờ để đặt.

### 4.3.3 Đặt sân

* Người dùng chọn khung giờ và ngày để đặt sân.
* Kiểm tra trùng lặp lịch đặt.
* Dữ liệu được lưu vào collection bookings.

### 4.3.4 Quản lý sân (Admin)

* Thêm, sửa, xóa sân.
* Tải ảnh sân lên Firebase Storage.
* Xem danh sách các đơn đặt sân.

### 4.3.5 Đánh giá sân

* Sau khi sử dụng, người dùng có thể đánh giá bằng sao và bình luận.
* Đánh giá được lưu vào Firestore và hiển thị công khai.

## **4.3** Kết quả thực nghiệm

### 4.4.1 Giao diện người dùng

* Giao diện thân thiện, tương thích điện thoại.
* Màu sắc hài hoà, dễ thao tác.
* Chia giao diện theo quyền: Admin / User.

### 4.4.2 Tính ổn định

* Ứng dụng hoạt động tốt trên các thiết bị Android thật và giả lập.
* Thời gian phản hồi nhanh nhờ Firebase realtime.

### 4.4.3 Kiểm thử

* Đã thực hiện kiểm thử chức năng: đăng nhập, đặt sân, sửa sân, xóa sân.
* Tình huống lỗi phổ biến đã được xử lý (ví dụ: nhập thiếu dữ liệu, đặt trùng khung giờ).

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## **5.1** Kết luận

Trong quá trình thực hiện đồ án *“Xây dựng ứng dụng đặt sân thể thao trên nền tảng Flutter sử dụng Firebase”*, nhóm đã hoàn thành các mục tiêu chính đề ra:

* Tìm hiểu và sử dụng thành thạo **Flutter framework** để xây dựng ứng dụng di động đa nền tảng.
* Kết nối và khai thác hiệu quả **Firebase** (Authentication, Firestore, Storage) cho phần backend.
* Cài đặt được các **chức năng cơ bản và quan trọng** của một hệ thống đặt sân:
  + Đăng ký/đăng nhập người dùng
  + Quản lý thông tin sân và đơn đặt sân
  + Đặt sân, đánh giá, phân quyền người dùng
* Ứng dụng hoạt động ổn định, có giao diện dễ sử dụng và khả năng mở rộng cao.

Thông qua đồ án, nhóm đã nâng cao kiến thức lập trình di động, kỹ năng làm việc nhóm, tổ chức code theo hướng mô-đun, và hiểu sâu hơn về quản lý trạng thái cũng như tích hợp dịch vụ đám mây vào ứng dụng thực tế.

5.2 Hạn chế

Mặc dù đã hoàn thiện được phần lớn các chức năng, đồ án vẫn còn một số hạn chế sau:

* Chưa hỗ trợ thanh toán trực tuyến.
* Thiếu tính năng thông báo (notification) khi đặt sân thành công.
* Giao diện chỉ tối ưu cho Android, chưa triển khai bản web hoặc iOS.
* Phân quyền admin được xử lý đơn giản, chưa có hệ thống phân vai nâng cao.

## **5.3 Hướng phát triển**

Trong tương lai, nhóm định hướng tiếp tục phát triển ứng dụng theo các hướng sau:

* **Tích hợp cổng thanh toán trực tuyến** (Momo, ZaloPay, v.v.).
* **Phát triển hệ thống thông báo real-time** bằng Firebase Cloud Messaging.
* **Hỗ trợ người quản lý sân** đăng ký sân riêng và tự quản lý lịch hoạt động.
* **Triển khai phiên bản Web và iOS** để tăng khả năng tiếp cận người dùng.
* **Tối ưu trải nghiệm người dùng** và hiệu suất ứng dụng trên thiết bị yếu.

# TÀI LIỆU THAM KHẢO

Google. *Flutter Documentation*. Truy cập tại: <https://flutter.dev/docs>

Google Firebase. *Firebase Documentation*. Truy cập tại: <https://firebase.google.com/docs>

Reso Coder. *Flutter & Firebase App Tutorial*. Truy cập tại: <https://resocoder.com/>

Udemy. *Flutter & Dart – The Complete Guide [2024 Edition]*. Tác giả: Maximilian Schwarzmüller.

YouTube - Net Ninja. *Flutter Tutorial for Beginners* Series.

Medium Articles. *How to Build a Booking App with Flutter and Firebase*.

Stack Overflow. *Hỏi đáp kỹ thuật trong quá trình xây dựng ứng dụng*.

Sách: “Lập trình Flutter cho người mới bắt đầu” – NXB Thống Kê, 2021.

Tài liệu giảng dạy môn “Lập trình thiết bị di động” – Khoa CNTT, Trường Đại học XYZ.