**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

****

**<Nhóm 2 – Hà Duy Cát Tường>**

2051012130 - Hà Duy Cát Tường

2051052062 - Bảo Khiêm

2051012071 - Huỳnh Trọng Nghĩa

2051012115 - Phạm Đức Tiến

2051012060 - Lê Hữu Mạnh

**Lập trình mobile**

**Giảng viên: Lê Ngọc Hiếu**

**Danh sách phân công**

|  |  |
| --- | --- |
| Hà Duy Cát Tường | Google Place API, thêm address bằng map, mô tả nghiệp vụ |
| Bảo Khiêm | Mô tả nghiệp vụ, chức năng, UI ở fragment home, hướng dẫn sử dụng |
| Phạm Đức Tiến | Thêm, sửa, xoá địa điểm, edit profile, xử lý login, camera |
| Huỳnh Trọng Nghĩa | Splash, theme UI, UI sign in, sign up, navigation bottom menu |
| Lê Hữu Mạnh | Chức năng sign up, thêm, sửa mục yêu thích |

**I/ MÔ TẢ NGHIỆP VỤ**

1/ Quản lý một ứng dụng giới thiệu nơi du lịch là một việc rất quan trọng, nhất là vào thời đại công nghệ phát triển, chúng ta cần có một trung gian quản lý các thông tin về các địa điểm du lịch để làm cầu nối giữa người dùng và bên các địa điểm du lịch, cần sự hợp tác tới từ người dùng, các doanh nghiệp du lịch và ban quản lý địa điểm du lịch đó. Khi cập nhật một địa điểm du lịch lên hệ thống ứng dụng, chúng ta cần phải xác thực lại thông tin người quản lý, vận hành nơi đó.

- Tiện ích và đa chức năng người dùng.

Về việc sử dụng hệ thống ứng dụng thì người dùng chính là các khách hàng tải và có nhu cầu sử dụng ứng dụng. Mỗi người dùng (user) sau khi tải ứng dụng thì sẽ được yêu cầu đăng ký và đăng nhập để có thể xem chi tiết bên trong ứng dụng. Ứng dụng ngoài ra còn hỗ trợ việc người dùng có thể thay đổi thông tin cá nhân trong tài khoản sau 30 ngày kể từ lần sửa gần nhất và hỗ trợ việc thay đổi mật khẩu nếu người dùng quên.

- Hỗ trợ công nghệ thân thiện với người dùng và định vị chính xác.

Sau khi đã tạo và đăng nhập vào tài khoản thì người dùng có thể xem các bài review từ các người dùng khác, comment, đánh giá và viết một bài review địa điểm du lịch riêng cho mình, với việc hỗ trợ của google map, người dùng có chọn địa chỉ và hiển thị địa điểm du lịch một các nhanh chóng. Với thanh tìm kiếm, ở đây người dùng có thể tìm ra danh sách địa điểm du lịch và các bài viết liên quan đến địa điểm đó.

II. Mô tả bài toán:

Người dùng với có các chức năng chính như Đăng Ký tài khoản, đăng nhập tài khoản, Quản lý tài khoản (bao gồm chức năng quên mật khẩu) , đăng xuất tài khoản, tìm kiếm địa điểm, bài viết, Thêm bài viết mới (bài review sẽ được công khai và ai cũng có thể thấy được), thêm địa điểm mới (trong chức năng mẹ là thêm bài viết mới), sửa bài viết, xóa bài viết, xem danh sách bài viết đã thích. Lên kế hoạch du lịch, Sửa và xóa kế hoạch du lịch (chỉ cá nhân người dùng thấy)

Mỗi tài khoản cá nhân gồm có username, password, nickname, ảnh avatar, giới tính.

Mỗi Bài do user viết có thông tin về tựa đề (title), địa điểm, thời gian, giá tiền và mô tả chi tiết, hình ảnh bài viết.

Comment là một phần lưu trữ thêm trong các bài viết gồm username và nội dụng của người đã comment.

Địa điểm là một phần lưu trữ của một bài viết, bao gồm các thông tin như tên địa điểm, tọa độ vị trí (lấy từ google map), hình ảnh tượng trưng.

Một kế hoạch du lịch do user viết có các thông tin như tên kế hoạch, địa điểm, thời gian, thông tin chi tiết kế hoạch đó.

**II/ GIỚI THIỆU CÁC TOOL ĐƯỢC SỬ DỤNG TRONG BÀI VÀ CƠ SỞ LÍ THUYẾT**

1. Giới thiệu về Android Studio

Android Studio: Môi trường Phát triển Ứng dụng Android Tối Ưu và Hiện Đại

Trong thời kỳ kỹ thuật số và cách mạng công nghệ, việc xây dựng ứng dụng di động không chỉ đòi hỏi kiến thức về lập trình mà còn cần một môi trường phát triển tối ưu để giúp nhà phát triển thực hiện các ý tưởng và dự án một cách hiệu quả. Android Studio, một sản phẩm của Google, đã trở thành sự lựa chọn hàng đầu cho các nhà phát triển ứng dụng Android, mang đến một loạt các công cụ và tính năng mạnh mẽ để hỗ trợ việc phát triển và kiểm thử ứng dụng một cách dễ dàng và hiệu quả.

- Giới thiệu về Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) mạnh mẽ và chính thức dành riêng cho việc xây dựng ứng dụng di động trên nền tảng Android. Được xây dựng trên cơ sở của IntelliJ IDEA, Android Studio mang đến cho các nhà phát triển một môi trường linh hoạt và tiện ích để tạo ra các ứng dụng Android đa dạng và sáng tạo.

- Tính năng Quan trọng

Thiết kế Giao diện Trực quan: Android Studio cung cấp công cụ Visual Layout Editor cho phép bạn thiết kế giao diện người dùng của ứng dụng bằng cách kéo và thả các thành phần. Bạn có thể xem trước giao diện trên nhiều loại thiết bị và màn hình khác nhau, giúp đảm bảo rằng ứng dụng của bạn hoạt động tốt trên mọi nền tảng.

Emulator Tích hợp và Thiết bị Ảo: Android Studio cung cấp các Emulator tích hợp, cho phép bạn kiểm tra và thử nghiệm ứng dụng trên nhiều thiết bị Android và phiên bản hệ điều hành khác nhau. Bạn cũng có thể tạo ra các thiết bị ảo để kiểm tra ứng dụng một cách nhanh chóng.

Hỗ trợ Ngôn ngữ Kotlin và Java: Android Studio hỗ trợ cả hai ngôn ngữ lập trình phổ biến trên nền tảng Android là Kotlin và Java. Việc này giúp bạn lựa chọn ngôn ngữ mà bạn thích và cảm thấy thoải mái, đồng thời giúp tăng hiệu suất và sáng tạo trong quá trình phát triển.

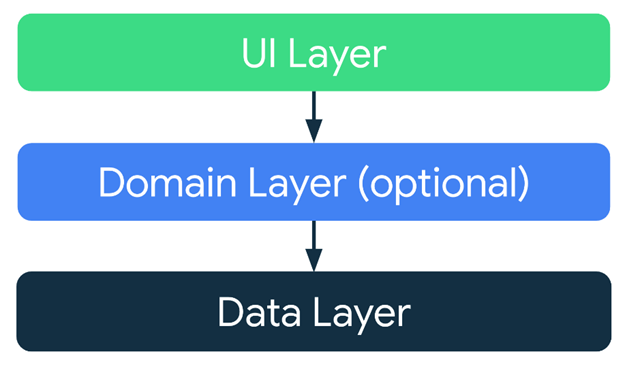
Quản lý Dự án Tốt hơn: Android Studio cung cấp công cụ quản lý dự án mạnh mẽ, giúp bạn kiểm soát mã nguồn và phiên bản một cách dễ dàng. Công cụ tích hợp Git giúp bạn quản lý mã nguồn và làm việc nhóm một cách hiệu quả.

Kiểm Thử và Phân Tích: Android Studio được tích hợp với các công cụ kiểm thử và phân tích mạnh mẽ, giúp bạn tìm ra lỗi, hiệu chỉnh hiệu suất và đảm bảo rằng ứng dụng hoạt động mượt mà và đáp ứng nhu cầu của người dùng.

Dự án Kiến Thức Cộng Đồng: Android Studio kết nối bạn với cộng đồng phát triển Android thông qua dự án Android Open Source Project (AOSP). Điều này giúp bạn có thể truy cập mã nguồn, tài liệu và tài nguyên từ cộng đồng, từ đó nâng cao kiến thức và tài năng của mình.

- Tạo Nền Tảng cho Ứng dụng Android

Android Studio không chỉ là một công cụ phát triển, mà còn là một nền tảng mạnh mẽ cho sự sáng tạo. Với khả năng thiết kế, phân tích, và kiểm thử mạnh mẽ, bạn có thể phát triển ứng dụng Android phức tạp và chất lượng cao, đồng thời tạo ra trải nghiệm tốt cho người dùng. Điều này đồng nghĩa với việc bạn có thể tham gia vào cuộc cách mạng số hóa và đóng góp vào cộng đồng phát triển ứng dụng Android toàn cầu.



- Cấu trúc cơ bản của một ứng dụng

Cấu trúc cơ bản của một chương trình trong Android Studio tuân theo kiến trúc và quy ước của việc phát triển ứng dụng Android. Dưới đây là cấu trúc cơ bản của một dự án trong Android Studio:

- Thư mục "app": Đây là nơi chứa mã nguồn của ứng dụng của bạn. Các tệp quan trọng thường được đặt trong thư mục con của "app".

- Thư mục "java": Chứa mã nguồn Java của ứng dụng. Mã nguồn của các hoạt động (Activities), lớp Adapter, các lớp xử lý sự kiện, và nhiều thứ khác sẽ được đặt trong các gói tương ứng.

- Thư mục "res": Chứa các tài nguyên giao diện người dùng và tài nguyên khác.

- Thư mục "layout": Chứa các tệp XML định dạng giao diện người dùng, ví dụ như tệp layout cho các hoạt động.

- Thư mục "drawable": Chứa hình ảnh và tập tin đồ họa.

- Thư mục "values": Chứa các tệp XML chứa tài nguyên như màu sắc, chuỗi, kích thước v.v.

- Thư mục "manifests": Chứa tệp AndroidManifest.xml, tệp này chứa thông tin cấu hình quan trọng về ứng dụng, bao gồm tên ứng dụng, phiên bản, quyền hạn và các hoạt động chính.

- Thư mục "java" và "res" khác: Thư mục "java" và "res" có thể chứa nhiều thư mục con khác, dựa trên cách bạn tổ chức dự án của mình.

- Thư mục "gradle": Chứa tệp "build.gradle" cho dự án và mô-đun ứng dụng. Tệp này quản lý việc cài đặt và cấu hình các thư viện và plugin của dự án.

- Tệp "gradlew" và "gradlew.bat": Đây là các tệp thực thi dòng lệnh cho hệ thống Gradle. Chúng cho phép bạn thực hiện các nhiệm vụ build và quản lý dự án bằng cách sử dụng dòng lệnh.

- Tệp "settings.gradle": Chứa cấu hình toàn cục cho các mô-đun của dự án.

- Tệp "local.properties": Chứa thông tin cấu hình cục bộ như đường dẫn đến Android SDK trên máy tính của bạn.

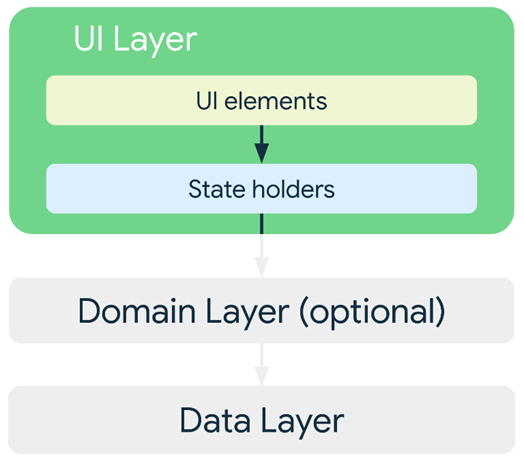
- Thư mục "build": Chứa tất cả các tệp được tạo ra từ quá trình build, bao gồm mã nguồn biên dịch, tệp tài nguyên và nhiều thứ khác.

- Lớp giao diện người dùng

Vai trò của lớp giao diện người dùng (hoặc *lớp bản trình bày*) là hiển thị dữ liệu ứng dụng trên màn hình. Bất cứ khi nào dữ liệu thay đổi, do sự tương tác của người dùng (chẳng hạn như nhấn một nút) hoặc đầu vào bên ngoài (chẳng hạn như phản hồi mạng), giao diện người dùng sẽ cập nhật để phản ánh các thay đổi đó.

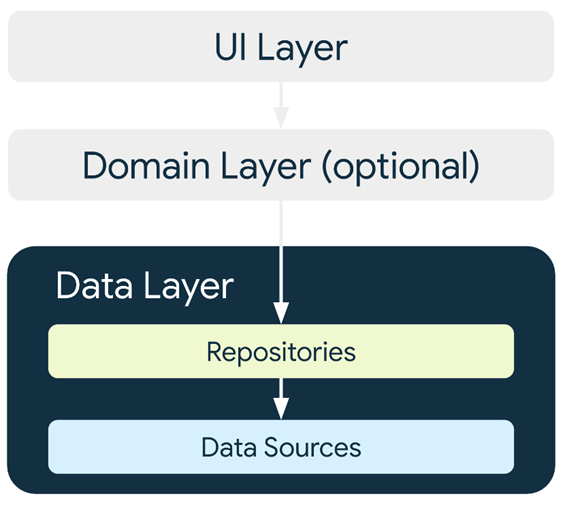
Lớp giao diện người dùng gồm hai nội dung:

* Các thành phần trên giao diện người dùng hiển thị dữ liệu trên màn hình. Bạn tạo các phần tử này bằng cách sử dụng các hàm View (Thành phần hiển thị) hoặc [Jetpack Compose](https://developer.android.com/jetpack/compose?hl=vi).
* Các chủ thể trạng thái (chẳng hạn như các lớp [ViewModel](https://developer.android.com/topic/libraries/architecture/viewmodel?hl=vi)) chứa dữ liệu, hiển thị thông tin đó tới giao diện người dùng và xử lý logic.
* Lớp dữ liệu
* Lớp dữ liệu của ứng dụng chứa logic nghiệp vụ. Logic nghiệp vụ là yếu tố tạo ra giá trị cho ứng dụng — logic này được tạo ra từ các quy tắc xác định cách ứng dụng tạo, lưu trữ và thay đổi dữ liệu.



- Lớp dữ liệu:

Lớp dữ liệu được tạo thành từ các kho lưu trữ, mỗi kho dữ liệu có thể chứa từ 0 đến nhiều nguồn dữ liệu. Bạn nên tạo một lớp kho lưu trữ cho từng loại dữ liệu khác nhau mà bạn xử lý trong ứng dụng. Ví dụ: bạn có thể tạo một lớp MoviesRepository cho dữ liệu liên quan đến phim hoặc một lớp PaymentsRepository cho dữ liệu liên quan đến các khoản thanh toán.



Các lớp kho lưu trữ chịu trách nhiệm về:

Hiển thị dữ liệu cho phần còn lại của ứng dụng.

Tập trung các thay đổi vào dữ liệu.

Giải quyết xung đột giữa nhiều nguồn dữ liệu.

Tóm tắt các nguồn dữ liệu từ phần còn lại của ứng dụng.

Chứa logic nghiệp vụ.

Mỗi lớp nguồn dữ liệu nên có trách nhiệm làm việc với chỉ một nguồn dữ liệu duy nhất, có thể là một tệp, nguồn mạng hoặc cơ sở dữ liệu cục bộ. Các lớp nguồn dữ liệu là cầu nối giữa ứng dụng và hệ thống để thao tác dữ liệu.

### - Lớp miền

Lớp miền là lớp không bắt buộc nằm giữa giao diện người dùng và các lớp dữ liệu.

Lớp miền chịu trách nhiệm về việc tổng hợp các logic nghiệp vụ phức tạp, hoặc logic nghiệp vụ đơn giản được sử dụng lại trong nhiều ViewModel. Lớp này là không bắt buộc vì không phải ứng dụng nào cũng có những yêu cầu này. Bạn chỉ nên sử dụng thuộc tính này khi cần, ví dụ: để xử lý độ phức tạp hoặc ưa chuộng khả năng tái sử dụng.

Các lớp này thường được gọi là trường hợp sử dụng (use case) hoặctrình tương tác (interactor). Mỗi trường hợp sử dụng phải có trách nhiệm đối với một chức năng. Ví dụ: ứng dụng có thể có một lớp GetTimeZoneUseCase nếu nhiều ViewModel dựa vào múi giờ để hiển thị thông báo thích hợp trên màn hình.

Cấu trúc này sẽ cung cấp một khung làm việc có tổ chức để phát triển ứng dụng Android trong Android Studio. Quy mô của dự án và cách bạn tổ chức thư mục có thể thay đổi tùy thuộc vào yêu cầu cụ thể của ứng dụng của bạn.

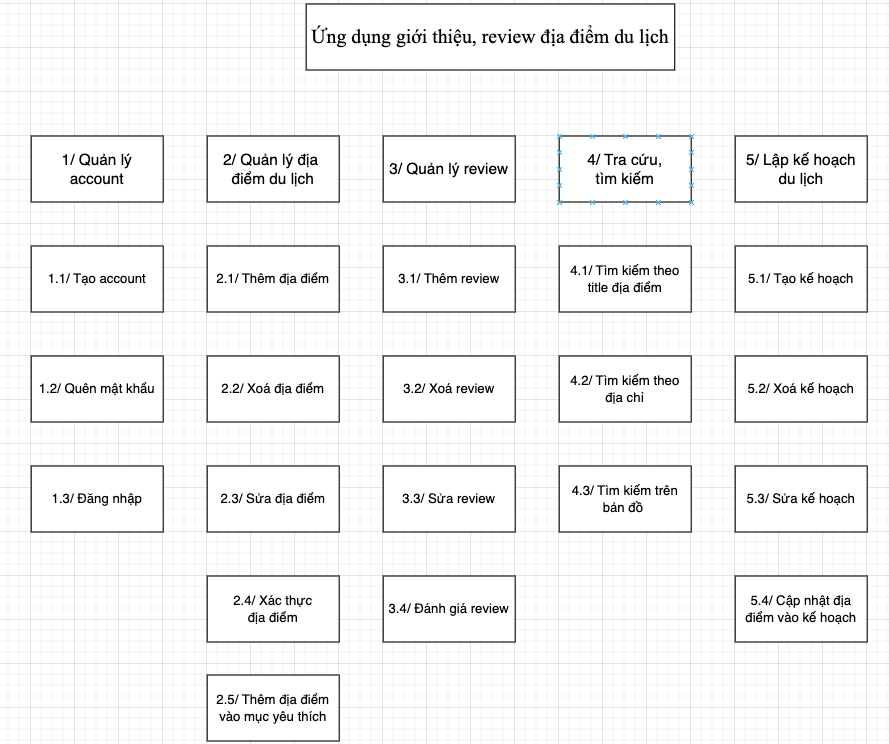
- Kết Luận

Android Studio đã chứng tỏ sự quan trọng và tối quan trọng của một môi trường phát triển ứng dụng chất lượng trong việc phát triển ứng dụng di động. Từ việc thiết kế giao diện đến kiểm thử và phân tích, Android Studio cung cấp mọi công cụ bạn cần để xây dựng ứng dụng Android thành công và đáp ứng nhu

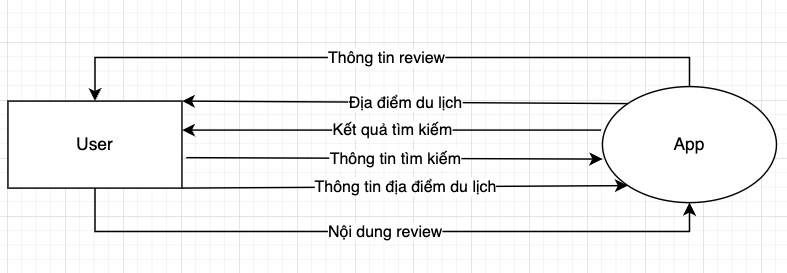
Dưới đây là hướng dẫn cài đặt Android Studio trên hệ điều hành Windows. Nếu bạn đang sử dụng hệ điều hành khác như macOS hoặc Linux, quá trình cài đặt cũng tương tự nhưng có một số khác biệt nhỏ.

**III.BIỂU ĐỒ DFD VÀ THIẾT KẾ CSDL:**

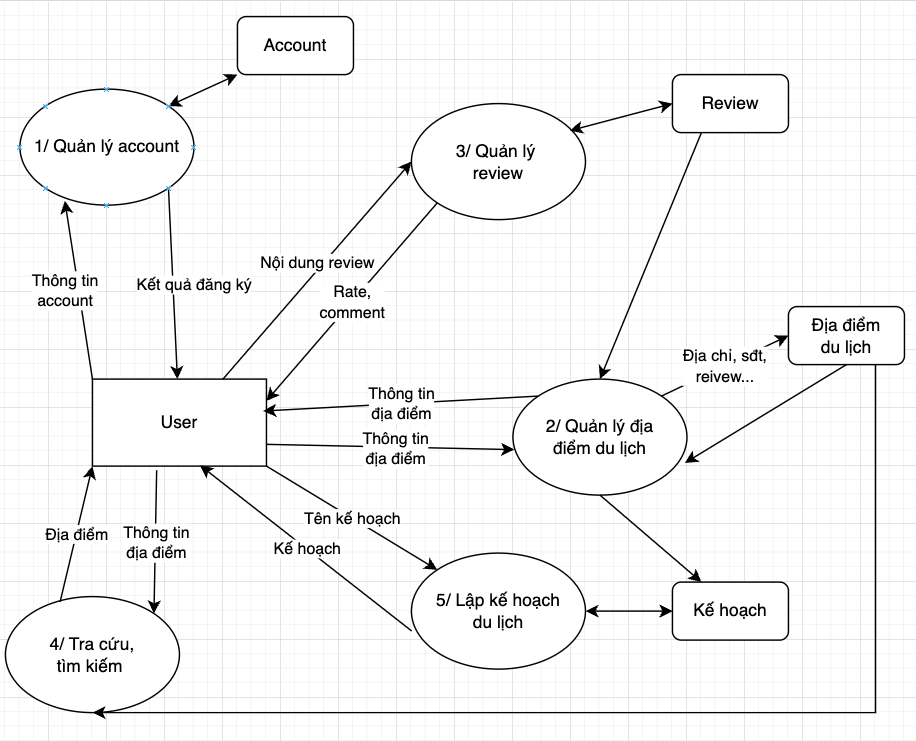
1. Biểu đồ phân rã chức năng:



2. Biểu đồ DFD mức ngữ cảnh:

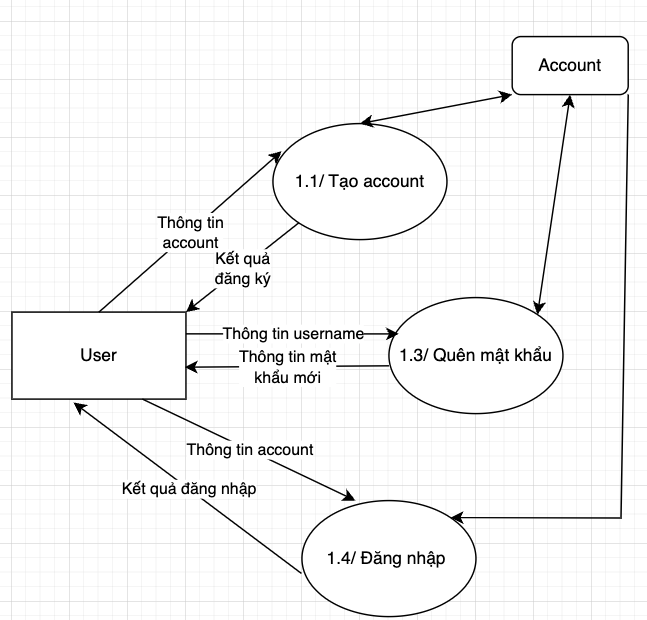


3. Biểu đồ DFD mức đỉnh:

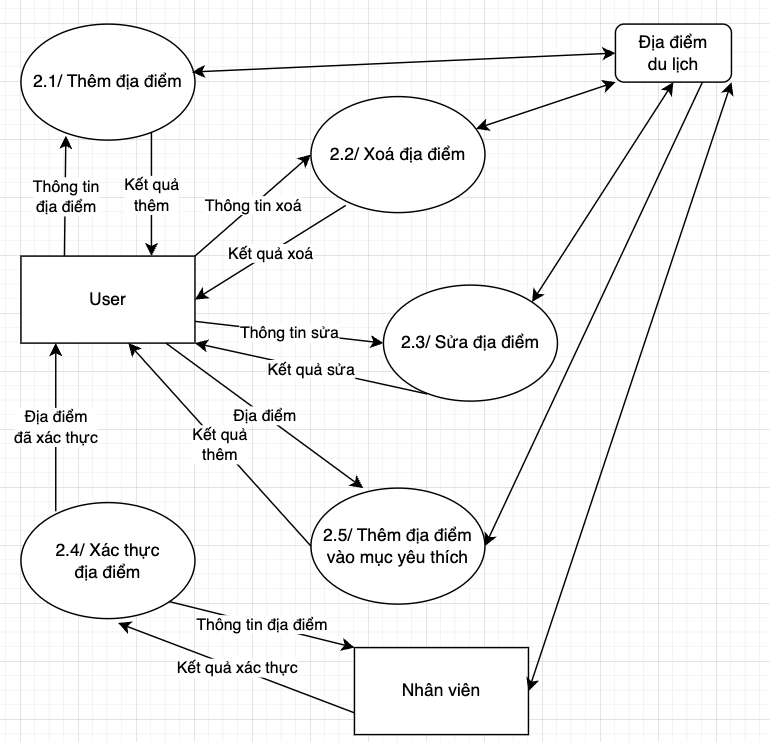


4. Biểu đồ DFD của từng chức năng:

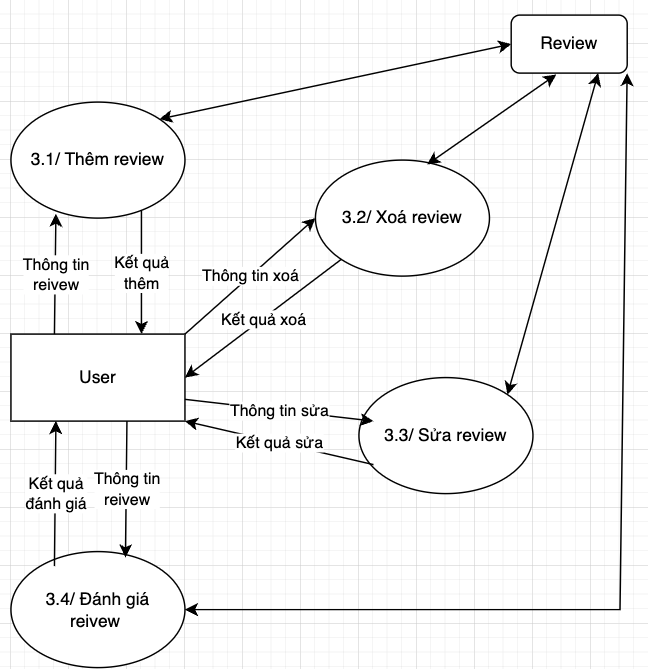
4.1 Chức năng quản lý account:



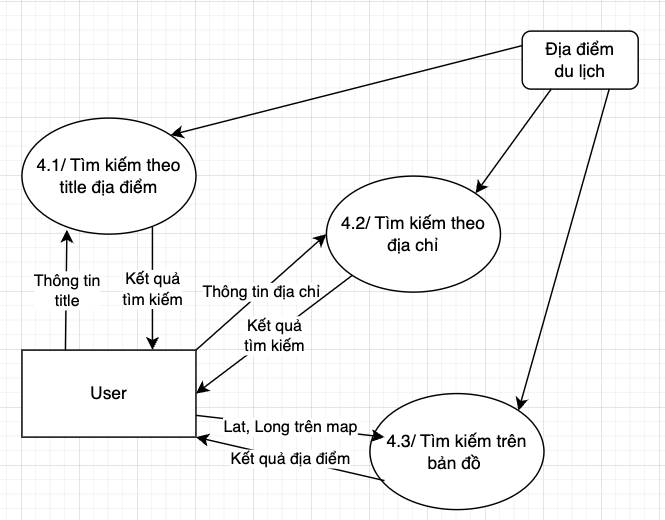
4.2 Chức năng quản lý địa điểm du lịch:



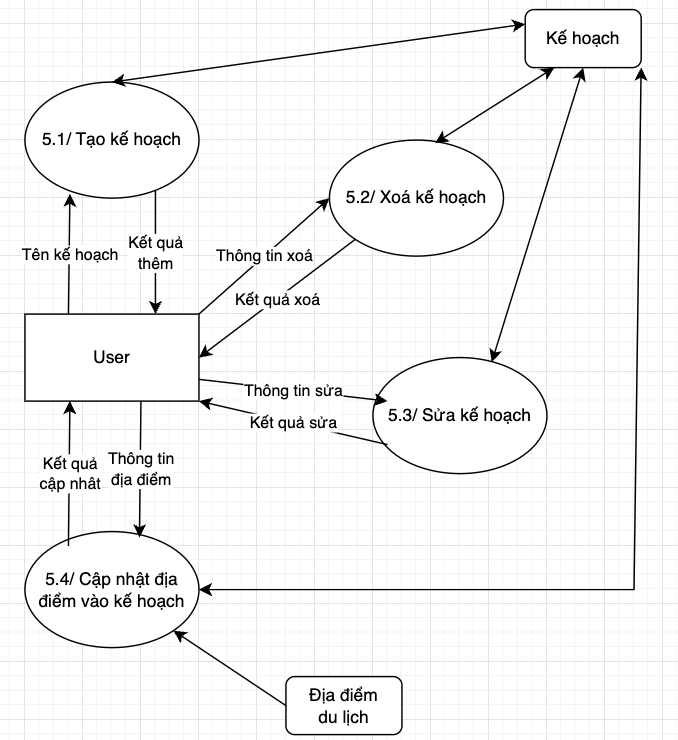
4.3 Chức năng quản lý review:



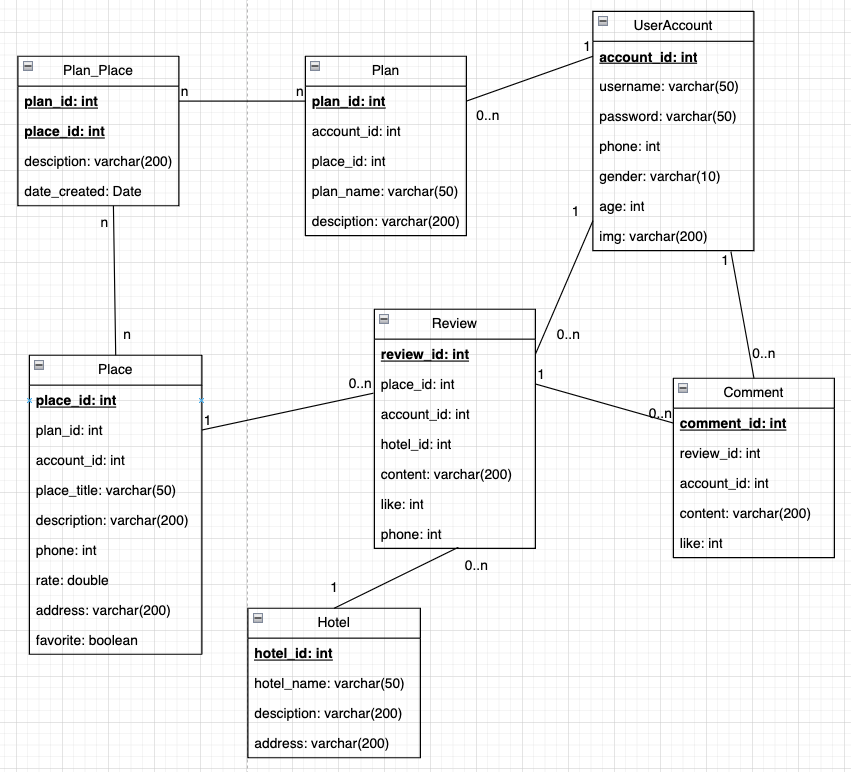
4.4 Chức năng tra cứu, tìm kiếm:



4.5 Chức năng quản lý kế hoạch:



5. Thiết kế CSDL:



UserAccount(account\_id, username, password, phone, gender, age, img)

Review(review\_id, #place\_id, #account\_id, content, like, phone)

Place(place\_id, #plan\_id, #account\_id, place\_title, description, phone, rate, address, favorite)

Hotel(hotel\_id, hotel\_name, address)

Comment(comment\_id, #review\_id, #account\_id, content, like)

Plan(plan\_id, #place\_id, #account\_id, plan\_name, description)

Plan\_Place(#plan\_id, #place\_id, description, date\_created)