**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**



**BÁO CÁO**

**MÔN HỌC: THỰC TẬP CƠ SỞ**

**Chủ đề: XÂY DỰNG ỨNG DỤNG CHẠY BỘ**

**Họ và tên:** Đặng Minh Đức

**Ngày sinh:** 26/10/2003

**Số điện thoại:** 0858189821

**Mã sinh viên:** B21DCCN236

**Lớp:** D21CQCN08-B

**Giảng viên hướng dẫn:** Nguyễn Văn Tiến

[I. Giới thiệu 3](#_Toc167720199)

[II. Tìm hiểu công nghệ 4](#_Toc167720200)

[1. Tìm hiểu về Flutter framework 4](#_Toc167720201)

[2. Tìm hiểu về Django rest framework 5](#_Toc167720202)

[III. Những việc chưa làm được 7](#_Toc167720203)

[IV. Ứng dụng 7](#_Toc167720204)

[1. Database diagram 7](#_Toc167720205)

[2. Màn hình và các chức năng 12](#_Toc167720206)

[2.1 Màn hình giới thiệu về ứng dụng 12](#_Toc167720207)

[2.2. Màn hình đăng nhập 13](#_Toc167720208)

[2.3 Màn hình đăng ký 14](#_Toc167720209)

[2.4 Màn hình tạo Profile 14](#_Toc167720210)

[2.5 Màn hình chính (Home) 15](#_Toc167720211)

[2.6 Các màn hình về User 16](#_Toc167720212)

[a) Màn hình chung 16](#_Toc167720213)

[b) Màn hình về danh sách hoạt động của User 18](#_Toc167720214)

[c) Màn hình về thống kê thông số các hoạt động của User 19](#_Toc167720215)

[2.7 Màn hình về cộng đồng, các hoạt động chung (Community) 24](#_Toc167720216)

[a) Màn hình về Event 24](#_Toc167720217)

[2.8 Màn hình bảng xếp hạng 33](#_Toc167720218)

[2.9 Màn hình về Store 34](#_Toc167720219)

[2.10 Màn hình về theo dõi hoạt động 36](#_Toc167720220)

I. Giới thiệu

Báo cáo này trình bày về quá trình xây dựng một ứng dụng chạy bộ sử dụng Flutter framework và Django REST framework. Ứng dụng này có các tính năng sau:

* Theo dõi quãng đường, tốc độ và thời gian chạy
* Ghi lại lịch sử chạy
* Hiển thị bản đồ tuyến đường chạy
* Chạy tích lũy điểm để mua các sản phẩm
* Tham gia các sự kiện, câu lạc bộ để thi đua với các vận động viên khác

Ứng dụng được xây dựng với mục tiêu cung cấp cho người dùng một công cụ đơn giản và hiệu quả để theo dõi và cải thiện thành tích chạy bộ của họ.

**Lựa chọn công nghệ:**

* **Flutter framework:** Flutter là một framework mã nguồn mở được phát triển bởi Google để xây dựng các ứng dụng di động đa nền tảng. Flutter sử dụng Dart, một ngôn ngữ lập trình hướng đối tượng, để viết code. Flutter được chọn vì nó có hiệu suất cao, dễ sử dụng và có cộng đồng nhà phát triển lớn.
* **Django REST framework:** Django REST framework là một framework mã nguồn mở được xây dựng trên nền tảng Django để xây dựng các API REST. Django REST framework được chọn vì nó cung cấp một cách đơn giản và hiệu quả để xây dựng các API REST an toàn và có thể mở rộng.

**Lợi ích của ứng dụng**

Ứng dụng này mang lại một số lợi ích cho người dùng, bao gồm:

* Giúp người dùng theo dõi và cải thiện thành tích chạy bộ của họ.
* Cung cấp động lực cho người dùng tập luyện thường xuyên hơn.
* Giúp người dùng kết nối với những người chạy bộ khác.
* Cung cấp dữ liệu có giá trị để người dùng theo dõi tiến độ của họ.

**Đối tượng người dùng**

Ứng dụng này hướng đến những người đam mê các môn thể thao như chạy bộ, đi bộ, đạp xe ở mọi trình độ. Ứng dụng này đặc biệt hữu ích cho những người mới bắt đầu muốn theo dõi tiến trình của họ và cải thiện thành tích của họ.

**Tóm lại**

Ứng dụng chạy bộ này là một công cụ đơn giản và hiệu quả để giúp người dùng theo dõi và cải thiện thành tích chạy bộ của họ. Ứng dụng được xây dựng với công nghệ hiện đại và mang lại nhiều lợi ích cho người dùng.

II. Tìm hiểu công nghệ

1. Tìm hiểu về Flutter framework

- Flutter là mobile UI framework mã nguồn mở, miễn phí của Google để tạo ra các giao diện chất lượng cao trên iOS và Android trong khoảng thời gian ngắn. Flutter hoạt động với những code sẵn có được sử dụng bởi các lập trình viên, các tổ chức.

- Một số đặc điểm:

* **Giao Diện User Phẳng (UI Flat):** Flutter sử dụng một giao diện User phẳng, điều này có nghĩa là mọi thành phần giao diện đều là các widget. Các widget này được kết hợp lại với nhau để tạo ra giao diện User cuối cùng. Flutter cung cấp một bộ sưu tập đa dạng các widget để tạo ra giao diện User phong phú và linh hoạt.
* **Đa Nền Tảng:** Flutter cho phép bạn phát triển ứng dụng cho nhiều nền tảng, bao gồm iOS, Android, Web và Desktop (Windows, macOS, Linux), từ một mã nguồn duy nhất. Điều này giúp giảm bớt thời gian và công sức cần thiết cho việc phát triển và duy trì ứng dụng trên các nền tảng khác nhau.
* **Hot Reload**: Hot Reload là một tính năng mạnh mẽ của Flutter cho phép bạn thay đổi mã nguồn và xem kết quả ngay lập tức trên ứng dụng đang chạy mà không cần phải khởi động lại ứng dụng. Điều này giúp tăng tốc quá trình phát triển và làm việc hiệu quả với giao diện User.
* **Dart Programming Language:** Flutter sử dụng ngôn ngữ lập trình Dart, được phát triển bởi Google, là một ngôn ngữ lập trình hiện đại, linh hoạt và dễ học. Dart hỗ trợ tính năng như kiểu tĩnh và động, asynchronous programming, và có cú pháp dễ đọc.
* **Cộng Đồng Lớn:** Flutter có một cộng đồng lớn và năng động với hàng ngàn các thành viên trên toàn thế giới. Cộng đồng này cung cấp các tài liệu, hướng dẫn, gói phần mềm (packages) và hỗ trợ kỹ thuật cho lập trình viên.
* Dựng view trong Flutter:
* **Thiết kế view:** Flutter sử dụng các widget để xây dựng giao diện User. Widget là các khối xây dựng cơ bản của UI, có thể được kết hợp với nhau để tạo ra các bố cục phức tạp. Flutter cung cấp một thư viện widget phong phú bao gồm các widget cơ bản như Text, Button, Image, Container, v.v. và các widget nâng cao hơn như ListView, GridView, Drawer, v.v.
* **Quản lý State**: State là dữ liệu xác định trạng thái hiện tại của ứng dụng. Ví dụ, trong một ứng dụng danh sách mua sắm, state có thể bao gồm danh sách các mục cần mua và trạng thái đã mua của từng mục. Flutter cung cấp nhiều cách để quản lý state của ứng dụng, bao gồm:
* **setState**: Sử dụng phương thức setState để cập nhật state của một widget và kích hoạt việc xây dựng lại widget đó và các widget con của nó.
* **Provider**: Sử dụng Provider để quản lý global state và cung cấp quyền truy cập vào state cho nhiều widget trong ứng dụng.
* **BLoC**: Sử dụng BLoC (Business Logic Component) để quản lý logic nghiệp vụ và state của ứng dụng một cách tách biệt khỏi giao diện User.
* **Provider**: Provider là một giải pháp quản lý state phổ biến trong Flutter. Nó cho phép bạn quản lý state toàn cầu và cung cấp quyền truy cập vào state cho nhiều widget trong ứng dụng một cách dễ dàng và hiệu quả. Provider sử dụng một hệ thống dựa trên widget để phân phối state thay đổi đến các widget quan tâm.

**Lợi ích của việc sử dụng Provider:**

* **Dễ sử dụng**: Provider cung cấp một API đơn giản và dễ sử dụng để quản lý state.
* **Tái sử dụng**: State được quản lý bởi Provider có thể được tái sử dụng trong nhiều widget khác nhau trong ứng dụng.
* **Phân tách**: Provider giúp tách biệt logic quản lý state khỏi giao diện User.
* **Kiểm soát**: Provider cung cấp nhiều tùy chọn để kiểm soát cách state được cập nhật và phân phối.

2. Tìm hiểu về Django rest framework

- Django Rest Framework là một framework phổ biến được sử dụng để xây dựng các ứng dụng WebAPI. Nó cung cấp đầy đủ các tính năng và công cụ hỗ trợ để phát triển và quản lý các WebAPI một cách dễ dàng và hiệu quả.

- Một số đặc điểm:

* Serialization: DRF cung cấp một cách dễ dàng để chuyển đổi dữ liệu của các model Django thành định dạng JSON hoặc XML và ngược lại. Điều này giúp giảm thiểu công việc lập trình của bạn trong việc xử lý và truyền dữ liệu qua API.
* Authentication và Permissions: DRF cung cấp các lớp xác thực và quyền hạn, giúp bạn dễ dàng xác định và kiểm soát quyền truy cập vào các tài nguyên API. Các phương pháp xác thực hỗ trợ bao gồm token authentication, session authentication và OAuth.
* Viewsets và Routers: DRF hỗ trợ sử dụng viewsets để tổ chức các API endpoint. Viewset cho phép bạn xử lý các yêu cầu CRUD (Create, Retrieve, Update, Delete) trên một loạt các đối tượng mô hình một cách dễ dàng. Routers giúp tự động thiết lập các URL cho các endpoint.
* Throttling: DRF cung cấp các công cụ giới hạn tốc độ yêu cầu của User, giúp đảm bảo tính ổn định và an toàn cho hệ thống của bạn.
* Serialization Validation: DRF cung cấp các công cụ để kiểm tra và xác thực dữ liệu đầu vào trong quá trình serialization. Điều này giúp đảm bảo tính toàn vẹn của dữ liệu mà User gửi đến và hiển thị các thông báo lỗi phù hợp nếu có sự cố.
* Đa ngôn ngữ: DRF hỗ trợ việc tạo các API hỗ trợ nhiều ngôn ngữ khác nhau. Bạn có thể dễ dàng tạo các phiên bản API dựa trên ngôn ngữ yêu cầu của User.
* Documentation: DRF đi kèm với một công cụ tạo tài liệu tự động. Bạn có thể dễ dàng tạo báo cáo tài liệu cho các API của mình, giúp tăng khả năng hiểu và sử dụng của nhóm phát triển và User cuối.

- Django Rest Framework xử lý và truy vấn dữ liệu trong database bao gồm các thành phần chính sau:

* Serializer: Serializer là một phần quan trọng trong DRF giúp chuyển đổi dữ liệu thành các định dạng phổ biến như JSON hoặc XML và ngược lại. Nó cũng hỗ trợ việc xác thực và kiểm tra dữ liệu đầu vào.
* ViewSets và View Functions: DRF cho phép bạn xác định các endpoint cho API của mình bằng cách sử dụng ViewSets hoặc View Functions. ViewSets là một lớp gợi ý để tạo ra các endpoint cho một model hoặc một nhóm model tương tự, trong khi View Functions là các hàm cụ thể xác định các endpoint.
* Routing: DRF sử dụng routing để liên kết các endpoint của API với các viewsets hoặc view functions. Bằng cách sử dụng routing, bạn có thể xác định URL và các phương thức HTTP được hỗ trợ cho mỗi endpoint.
* Authentication và Permissions: DRF cung cấp các cơ chế xác thực và kiểm soát quyền truy cập để bảo vệ API của bạn khỏi việc truy cập trái phép. Bạn có thể thiết lập xác thực thông qua token hoặc session và kiểm soát quyền truy cập dựa trên vai trò của User, như là quản trị viên hoặc User thường.
* Pagination: DRF hỗ trợ phân trang thông qua pagination classes. Bạn có thể xác định số lượng items hiển thị trên mỗi trang và xem danh sách các trang trong kết quả truy vấn.
* Filtering và Searching: DRF cho phép bạn tạo các bộ lọc và tìm kiếm để lọc và tìm kiếm dữ liệu trong cơ sở dữ liệu dễ dàng. Bạn có thể thực hiện các query như tìm kiếm theo từ khóa, lọc theo các điều kiện với các tham số đầu vào từ request.  
  DRF giúp bạn nhanh chóng xây dựng các API mạnh mẽ và linh hoạt thông qua các công cụ và thư viện tích hợp sẵn. Nó giúp tiết kiệm thời gian và công sức trong việc xử lý và truy vấn dữ liệu trong database của bạn.

III. Những việc chưa làm được

- Theo dõi khi người dùng hoạt động với vị trí thực.

- Chưa xử lý về các hình ảnh trong ứng dụng (toàn bộ hình ảnh là cứng không lấy từ database).

- Các thống kê về người dùng, các hoạt động, các bài post chưa được lọc để đúng với thời gian Event hay Club được tạo ra.

- Thống kê trong bảng xếp hạng của Club, Event chưa đúng với thời gian người dùng tham gia và một Club, Event.

- Chưa xây dựng tính năng quản lý các người dùng trong một Club, Event.

- Chưa phân quyền cho người quản lý của Club, Event.

- Các Club, Event với Privacy là Private, Public thì đều giống nhau.

- Chưa xây dựng Group Event (Giải thích: Event sẽ có 2 loại là Group Event, Individual Event. Các group event sẽ bao gồm nhiều người dùng trong 1 group cùng thi đua để group mình dẫn đầu)

- Chưa xây dựng tính năng mỗi Event sẽ có một

- Chưa xây tính năng thử thách các vận động viên khi tham gia Event  
Ví dụ: Tích lũy hoàn thành 100km, 100 giờ chạy bộ

- Chưa phân quyền người dùng khi cài đặt, chỉnh sửa các thuộc tính về Privacy.

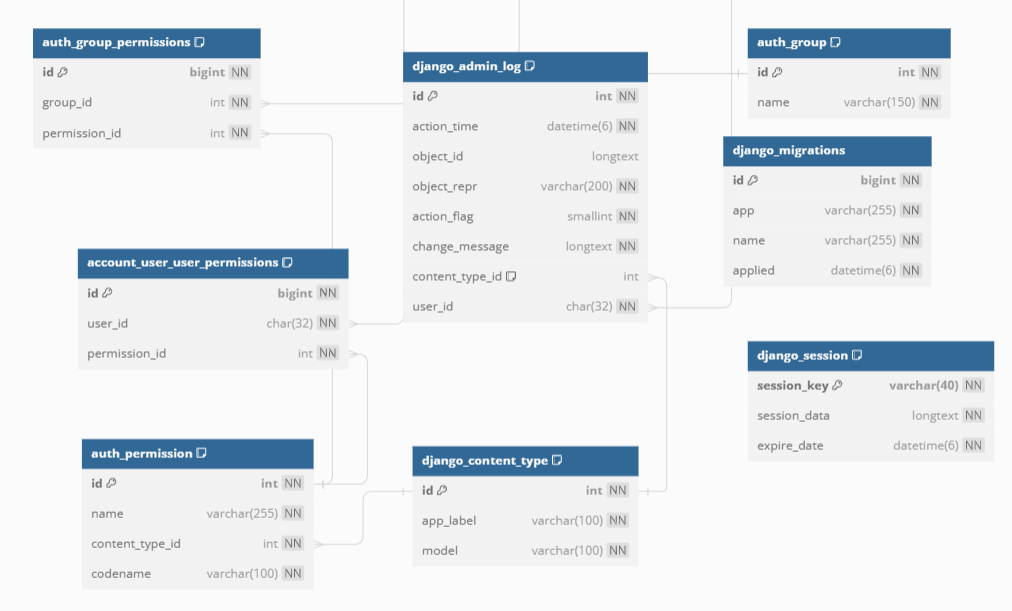
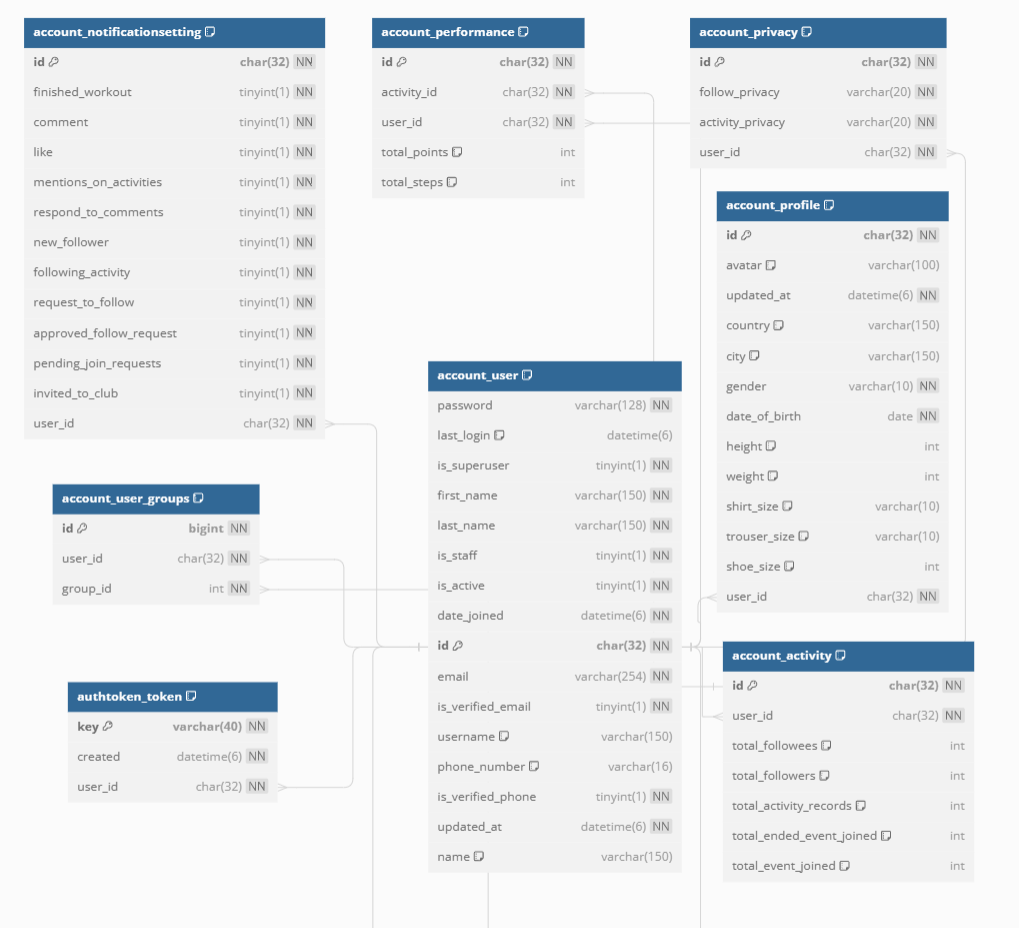
IV. Ứng dụng

1. Database diagram

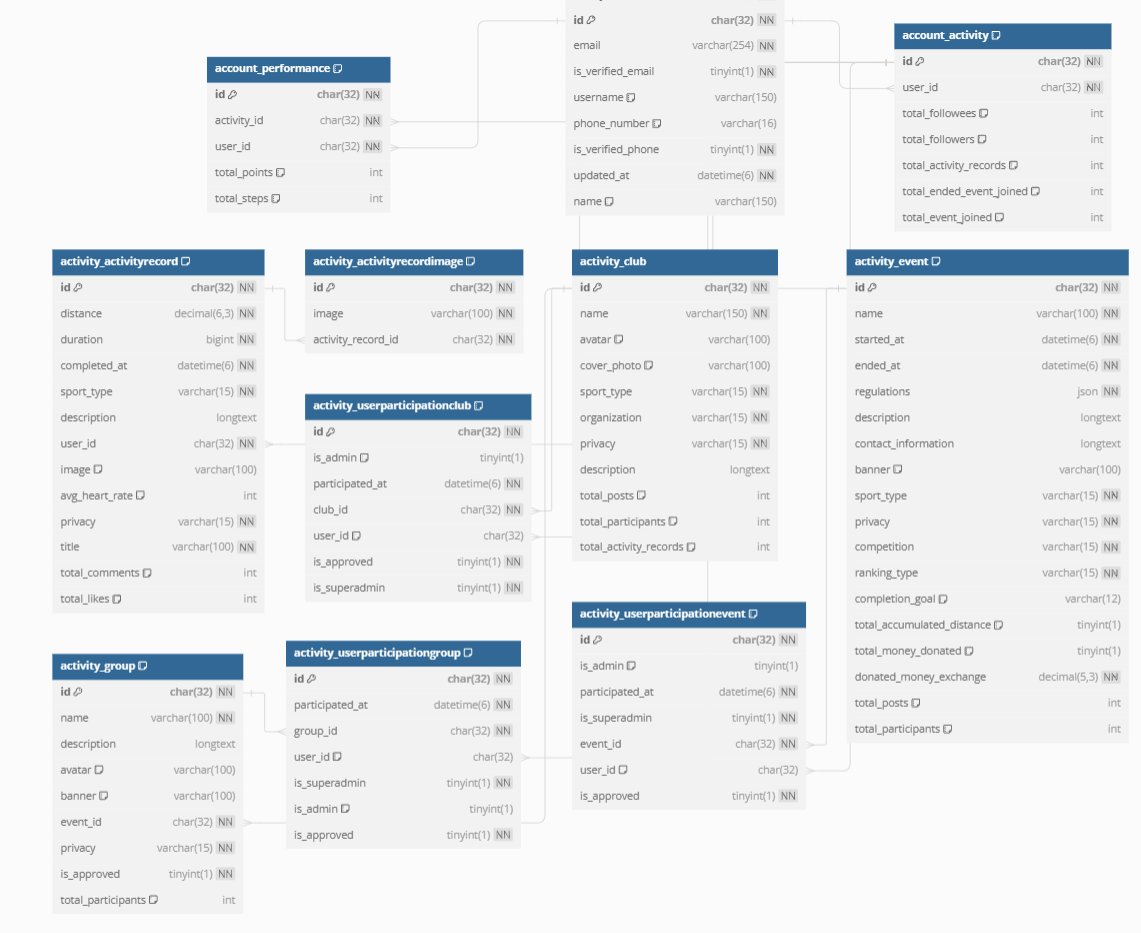
- Link:

[Running Diagram - dbdiagram.io](https://dbdiagram.io/d/Running-Diagram-663df3b79e85a46d557f85f4)

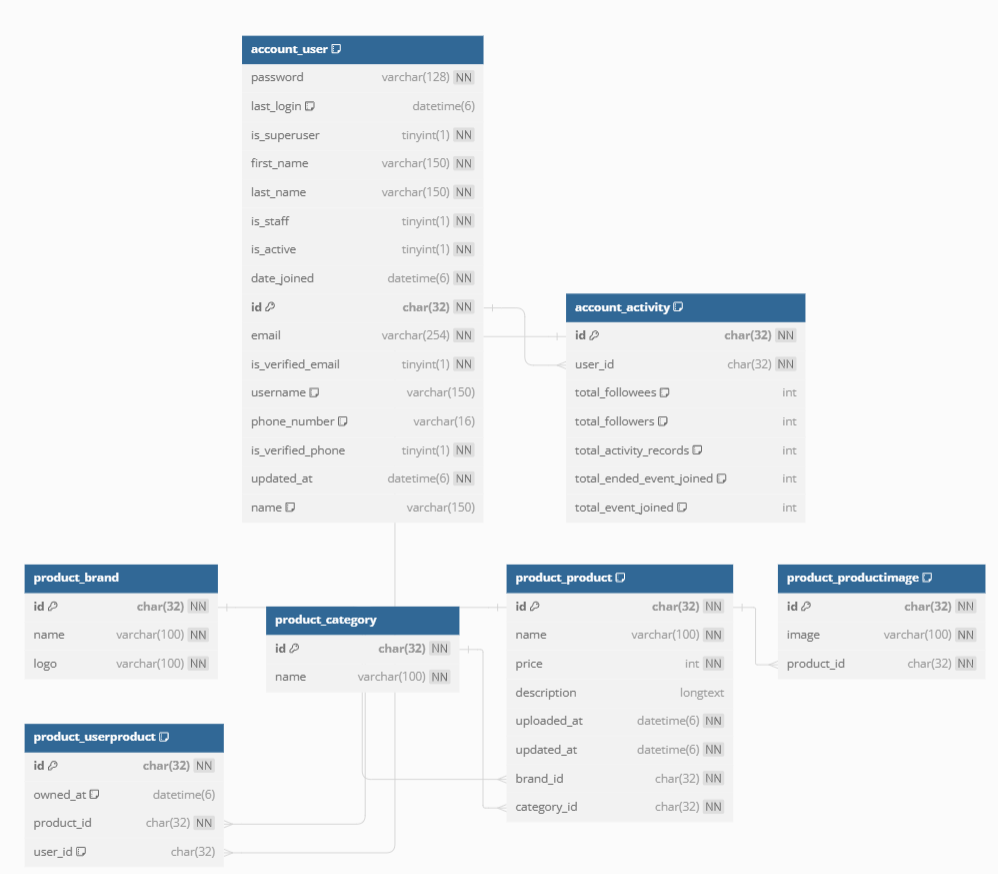
- Account:



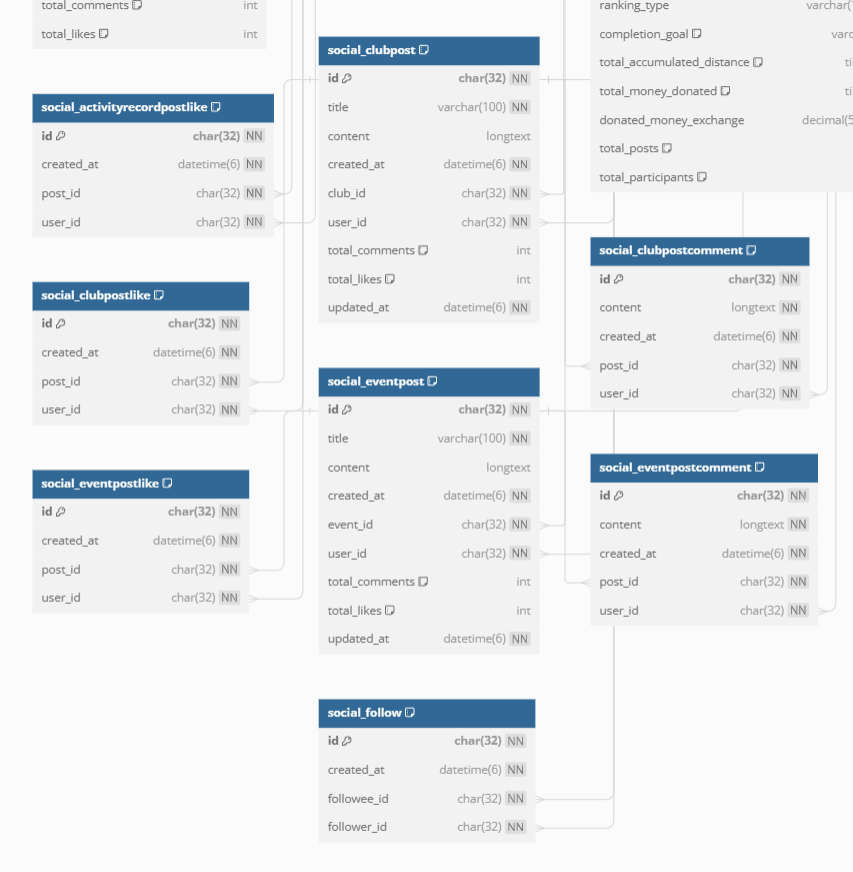
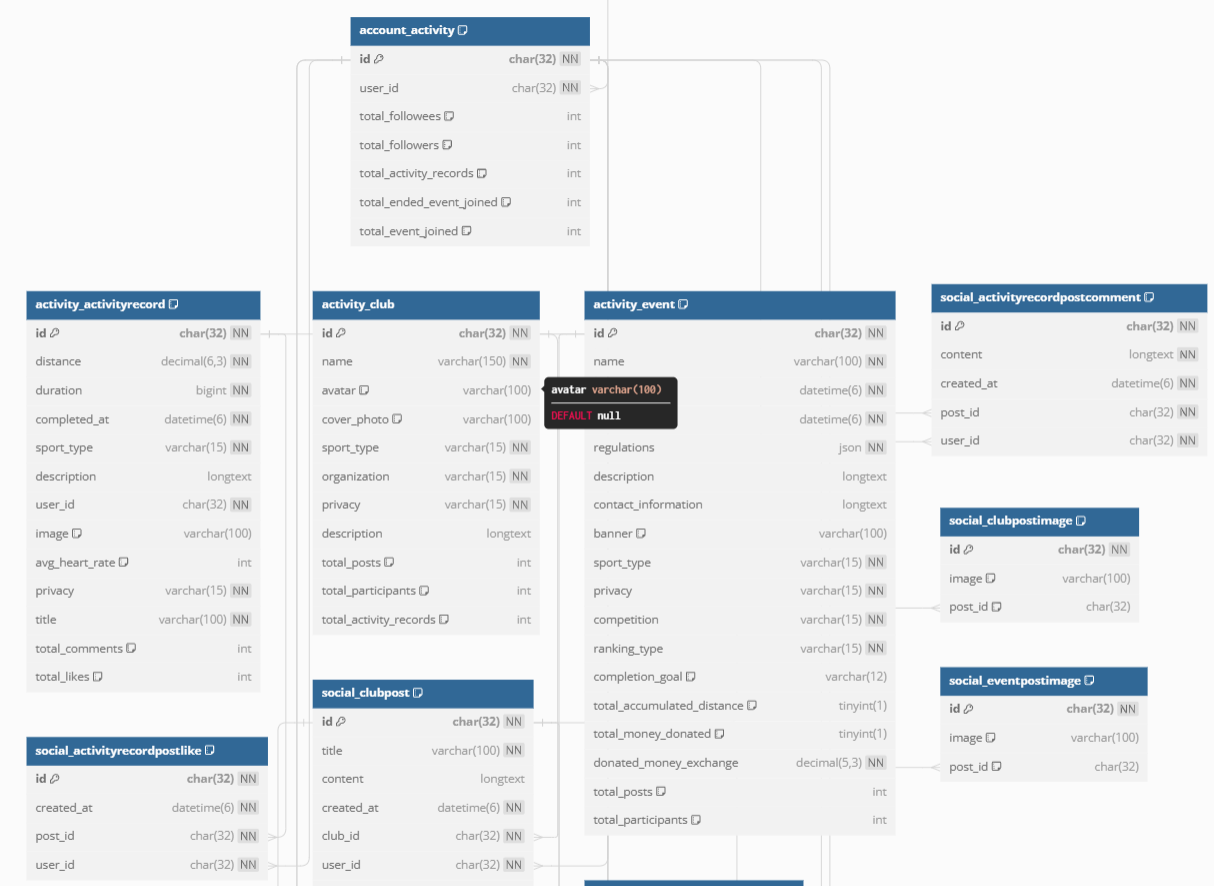
- Activity



- Product:



- Social:



2. Màn hình và các chức năng

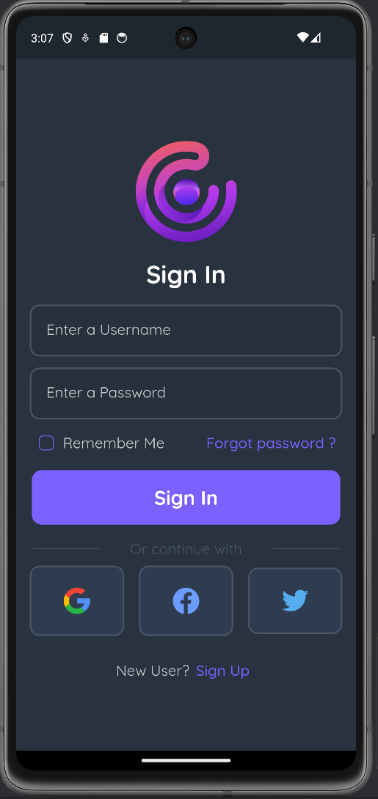
2.1 Màn hình giới thiệu về ứng dụng

Bao gồm 2 view: GetStartedView và OnBoardingView

* GetStartedView: là màn hình chào mừng hoặc giới thiệu ban đầu cho ứng dụng. Nó có vai trò thu hút sự chú ý của User, đồng thời khuyến khích họ bắt đầu sử dụng ứng dụng.
* OnBoardingView: cung cấp hướng dẫn từng bước cho User về cách sử dụng các tính năng chính của ứng dụng. Nó giúp User hiểu rõ cách thức hoạt động của ứng dụng và cách thực hiện các tác vụ cơ bản một cách hiệu quả.

|  |  |
| --- | --- |
| 2.1.1 GetStartedView | 2.1.2 OnBoardingView |

2.2. Màn hình đăng nhập

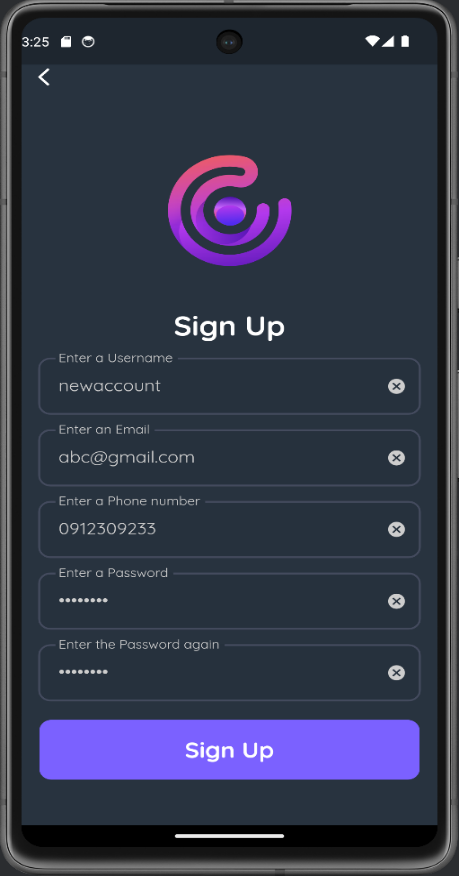
LoginView: xác thực danh tính User và kiểm soát quyền truy cập.

2.2.1 LoginView

Luồng hoạt động:

1. Người dùng truy cập màn hình đăng nhập.
2. Nhập tên người dùng và mật khẩu.
3. Nhấp vào nút "Đăng nhập".
4. Hệ thống xử lý thông tin đăng nhập.
5. Xác thực danh tính người dùng.
6. Hiển thị thông báo thành công hoặc thất bại.
7. Chuyển hướng người dùng đến màn hình chính của ứng dụng nếu đăng nhập thành công.

2.3 Màn hình đăng ký



2.3.1 SignUpView

Luồng hoạt động:

1. Người dùng truy cập màn hình đăng ký.
2. Nhập thông tin đăng ký.
3. Nhấp vào nút "Đăng ký".
4. Hệ thống xử lý thông tin đăng ký.
5. Hiển thị thông báo thành công hoặc thất bại.
6. Chuyển hướng người dùng đến màn hình tạo Profile.

2.4 Màn hình tạo Profile

Luồng hoạt động:

1. Người dùng truy cập màn hình tạo profile.
2. Nhập thông tin cá nhân.
3. Nhấp vào nút "Lưu".
4. Hệ thống lưu thông tin cá nhân của người dùng.
5. Chuyển hướng tới màn hình chính

|  |  |
| --- | --- |
| *2.4.1 CreateProfileView* | *2.4.2 CreateProfileView* |

2.5 Màn hình chính (Home)

Bao gồm:

* Level của User.
* Thông số về hoạt động ngày hôm nay của User bao gồm:
  + Tổng thời gian hoạt động
  + Số bước chân
  + Số điểm tích lũy được
* 5 hoạt động mới nhất gần đây và nút “See all” xem toàn bộ các hoạt động của User này.

|  |  |
| --- | --- |
| 2.5.1 HomeView | 2.5.2 HomeView |

2.6 Các màn hình về User

a) Màn hình chung

- Bao gồm:

* Số điểm tích lũy của User.
* Thống kê sẽ bao gồm 2 phần:
* Thống kê hoạt động của User về tổng khoảng cách, số ngày hoạt động, tốc độ trung bình, tổng thời gian, nhịp tim trung bình, và số bước chân trên phút.
* Các sản phẩm mà User đã dùng point để mua.

|  |  |
| --- | --- |
| *2.6.a.1 UserView* | *2.6.a.2 UserView* |

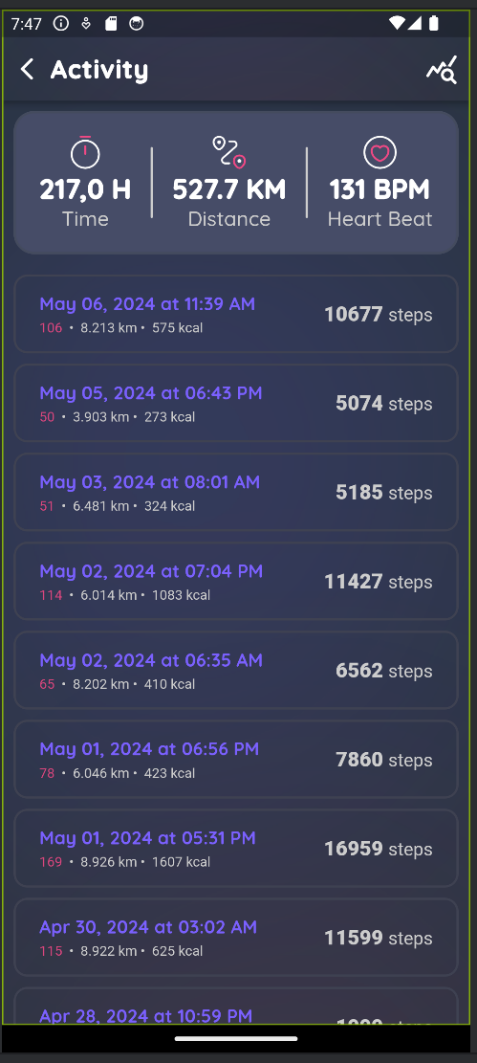
- Tính năng:

* Thống kê tuần, tháng, năm, và tất cả thời gian.

|  |  |  |  |
| --- | --- | --- | --- |
| 2.6.a.3 Tuần | 2.6.a.4 Tháng | 2.6.a.5 Năm | 2.6.a.6 Toàn bộ thời gian |

* Tìm kiếm theo tên các sản phẩm theo tên, tên brand, giá của sản phẩm.

|  |  |  |
| --- | --- | --- |
| 2.6.a.7 Tìm kiếm theo tên | 2.6.a.8 Tìm kiếm theo tên brand | 2.6.a.9 Tìm kiếm theo giá |

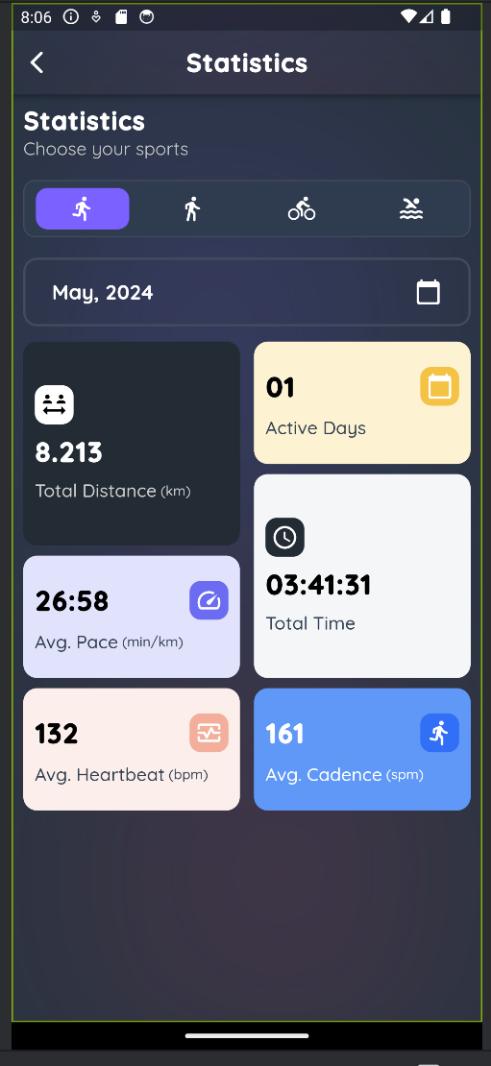
b) Màn hình về danh sách hoạt động của User

2.6.b.1 ActvitiyRecordListView

Bao gồm:

* Thống kê chung
* Danh sách các hoạt động của User (pagination).
* Các hoạt động bao gồm:
* Ngày hoàn thành.
* Số điểm tích lũy.
* Quãng đường thực hiện.
* Uớc tính số kcal.
* Số bước chân ước tính.

c) Màn hình về thống kê thông số các hoạt động của User



2.6.c.1 ActvitiyRecordStatsView

- Bao gồm:

* Loại hoạt động: chạy bộ, đi bộ, đi xe đạp, bơi
* Thống kê theo khoảng thời gian nào đó
* Các thông số

- Tính năng:

* Lọc các hoạt động thuộc một khoảng thời gian nào đó và tính thông số

|  |  |  |  |
| --- | --- | --- | --- |
| 2.6.c.2 Lọc theo tháng | 2.6.c.3 Lọc theo năm | 2.6.c.4 Kết quả | |
| *2.6.c.5 Chạy bộ* | *2.6.c.6 Đi bộ* | *2.6.c.7 Đạp xe* | *2.6.c.8 Bơi* | |

**d) Màn hình về danh sách các User mà User này theo dõi**

- Bao gồm:

* Danh sách các User đang theo dõi
* Danh sách các User theo dõi

|  |  |
| --- | --- |
| *2.6.d.1 theo dõiView* | *2.6.d.2 theo dõiView* |

**-** Tính năng:

* Tìm kiếm theo tên User

|  |  |
| --- | --- |
| *2.6.d.3 Tìm kiểm User theo dõing* | *2.6.d.4 Tìm kiếm User theo dõier* |

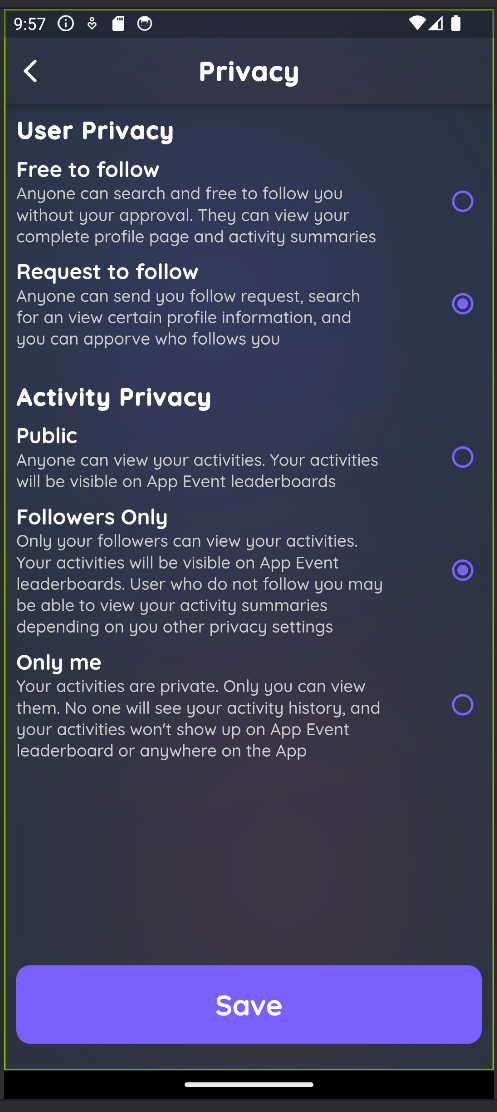
**e) Màn hình cài đặt**

- Bao gồm 4 màn hình:

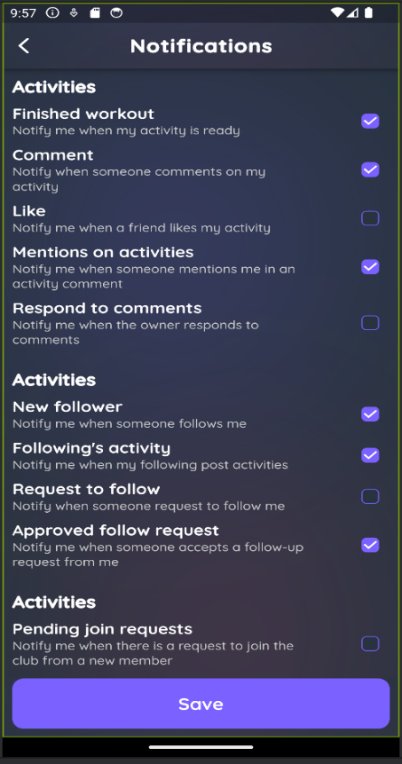
* Màn hình Profile Setting
  + Bao gồm thông tin User: tên, email, quốc gia, thành phố, giới tính, ngày sinh nhật, chiều cao, cân nặng, số điện thoại, kích cỡ quần áo giày.

|  |  |
| --- | --- |
| *2.6.e.1 ProfileSettingView* | *2.6.e.2 ProfileSettingView* |

* Màn hình Privacy Setting

Bao gồm:

2.6.e.3 PrivacySettingView

* User Privacy: Khi User yêu cầu theo dõi
* Free to theo dõi: User nào cũng có thể theo dõi tài khoản của User khác mà họ quan tâm mà không cần sự chấp thuận của họ.
* Request to theo dõi: User muốn theo dõi một người dùng khác, họ cần gửi một yêu cầu theo dõi.
* Activity Privacy:
* Public: Ai cũng có thể xem bài đăng về hoạt động.
* Followers Only: Chỉ những người theo dõ mới có thể xem về hoạt động
* Only me: Chỉ mình tôi
* Màn hình Notification Setting

Bao gồm về việc cho phép các thông báo:

Finished workout: Thông báo khi lưa một hoạt động

Comment: Thông báo khi ai đó bình luận về hoạt động của mình

Like: Thông báo khi ai đó thích hoạt động của mình

…

2.6.e.4 NotificationSettingView

* Màn hình Password Setting
* Những trường cơ bản để xác thực cho việc thay đổi mật khẩu

|  |  |
| --- | --- |
| *2.6.e.5 ChangePasswordSettingView* | *2.6.e.6 ChangePasswordSettingView* |

* Thông báo khi thay đổi thành công
* Thông báo khi thay đổi thất bại

2.7 Màn hình về cộng đồng, các hoạt động chung (Community)

a) Màn hình về Event

- Bao gồm:

* Thanh tìm kiếm và lọc các Event
* Các event đã tham gia và kết thúc
* Danh sách các Event phổ biến với nhiều người tham gia
* Danh sách tất cả các Event
* Tạo một event mới bằng cách nhấn nút cộng dưới cùng bên phải

|  |  |
| --- | --- |
| *2.7.a.1 EventView* | *2.7.a.2 EventView* |

- Tính năng:

* Tìm kiếm và lọc các Event

|  |  |
| --- | --- |
| *2.7.a.3 Tìm kiếm Event* | *2.7.a.4 Kết quả* |

|  |  |
| --- | --- |
| *2.7.a.5 Lọc Event* | *2.7.a.6 Kết quả* |

* Tạo mới một Event

|  |  |  |
| --- | --- | --- |
| *2.7.a.7 EventInformation*  *CreateView* | *2.7.a.8 EventFeature*  *CreateView* | *2.7.a.9 EventAdvanced*  *OptionCreateView* |

Kết quả khi tạo thành công

|  |  |
| --- | --- |
| 2.7.a.10 Tạo thành công | *2.7.a.11 Tạo thành công* |

**b) Màn hình về các bài đăng hoạt động của User**

Bao gồm 3:

* Bài đăng về hoạt động của tất cả User trên hệ thống
* Bài đăng về hoạt động của các User đã theo dõi trên hệ thống
* Bài đăng về hoạt động của cá nhân

|  |  |  |
| --- | --- | --- |
| *2.7.b.1 FeedView* | *2.7.b.2 FeedView* | *2.7.b.3 FeedView* |

**c) Màn hình về Club**

**-** Bao gồm:

2.7.c.1 ClubView

* Danh sách các Club mà User đã tham gia
* Thanh tìm kiếm vào lọc các Club mà User đã tham gia
* Tạo một event mới bằng cách nhấn nút cộng dưới cùng bên phải

- Tính năng:

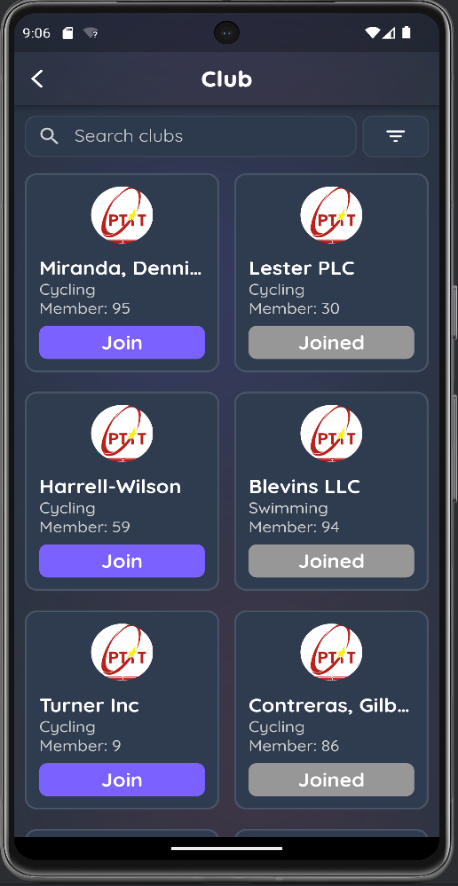
* Tìm kiếm và lọc các Club đã tham gia

|  |  |
| --- | --- |
| *2.7.c.2 Lọc Club* | *2.7.c.3 Kết quả* |

* Tạo mới một Club

|  |  |
| --- | --- |
| *2.7.c.4 ClubCreateView* | *2.7.5 ClubCreateView* |

|  |  |
| --- | --- |
| *2.7.c.6 Kết quả* | *2.7.c.7 Kết quả* |

**-** Màn hình về danh sách toàn bộ các Club

Bao gồm:

* Thanh tìm kiếm và lọc các Club
* Nút Join để tham gia Club
* Tính năng:
* Tìm kiếm và lọc các Club đã tham gia

|  |  |
| --- | --- |
| *2.7.c.8 Lọc Club* | *2.7.c.9 Kết quả* |

* Tham gia và rời Club

|  |  |
| --- | --- |
| *2.7.c.10 Rời Club* | *2.7.c.11 Kết quả* |

|  |  |
| --- | --- |
| *2.7.c.12 Tham gia Club* | *2.7.c.13 Kết quả* |

* Màn hình về chi tiết thông tin một Club

|  |  |
| --- | --- |
| *2.7.c.14 ClubDetailView* | *2.7.c.15 ClubDetailInformationView* |

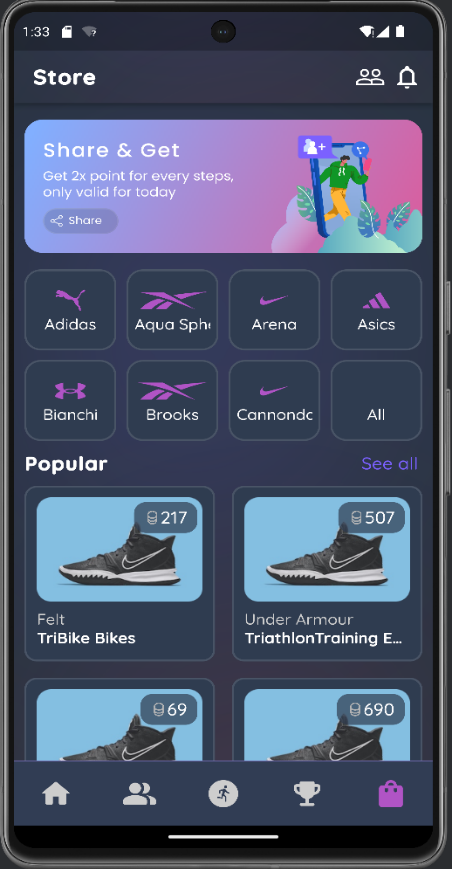
2.8 Màn hình bảng xếp hạng

- Bao gồm:

* Thông số về hoạt động của các User
* Sắp xếp theo tổng thời gian, khoảng cách, lọc ra các User là nam hoặc nữ
* Thống kê hoạt động theo tuần, tháng năm

|  |  |  |
| --- | --- | --- |
| *2.8.1 RankView* | *2.8.2 RankView* | *2.8.3 RankView* |
| *2.8.4 RankView* | *2.8.5 RankView* | *2.8.6 RankView* |

2.9 Màn hình về Store

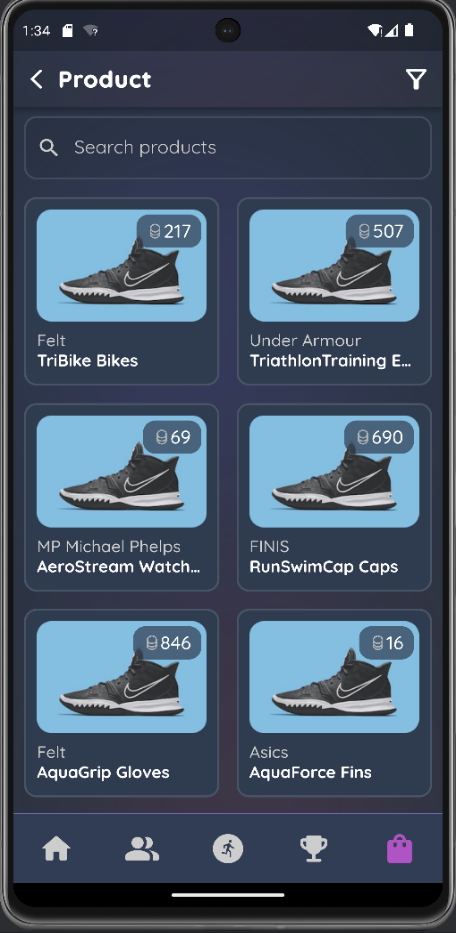
****

2.9.1 StoreView

Bao gồm:

* Các brand chính
* Các sản phẩm chính
* Lọc theo các brand

- Màn hình danh sách các sản phẩm

Bao gồm:

2.9.2 ProductView

* Thanh tìm kiếm
* Bộ lọc ra các loại sản phẩm theo category, brand
* Tính năng tìm kiếm và lọc sản phẩm

|  |  |
| --- | --- |
| *2.9.3 Lọc và tìm kiếm* | *2.9.4 Kết quả* |

* Tính năng mua sản phẩm
* Chọn một sản phẩm bất kỳ
* Thông tin chi tiết của sản phẩm hiện ra
* Nhấn mua, User nhập thông tin mật khẩu để xác nhận mua
* Mua thành công nếu mật khẩu đúng và số Point hiện tại dủ

|  |  |  |
| --- | --- | --- |
| *2.9.5 ProductDetailView* | *2.9.6 ProductDetailView* | *2.9.7 ProductDetailView* |

|  |  |
| --- | --- |
| *2.9.8 Không đủ Point* | *2.9.9 Sai mật khẩu* |

2.10 Màn hình về theo dõi hoạt động

|  |  |  |
| --- | --- | --- |
| *2.10.1 ActivityRecordView* | *2.10.2 ActivityRecordView*  *Khi di chuyển* | *2.10.3 ActivityRecordView*  *Khi dừng* |
|  |  |  |

**-** Màn hình lưu lại hoạt động

* Bao gồm thông tin: ảnh map về hoạt động, loại hình thể thao, quãng đường, tốc độ trung bình, tiêu đề và mô tả cho hoạt động, những ai có thể nhìn thấy hoạt động

|  |  |  |
| --- | --- | --- |
| *2.10.4 ActivityRecordCreateView* | *2.10.5 ActivityRecordCreateView* | *2.10.6 ActivityRecordDetailView* |

**Tài liệu tham khảo**

* [Run & Earn App (Community) – Figma](https://www.figma.com/design/KKwMfggkwtkNaoES8RUtbr/Run-%26-Earn-App-(Community)?node-id=1-43&t=OFUaTmIdVwDHYT58-0)
* Ứng dụng Up race
* [Home - Django REST framework (django-rest-framework.org)](https://www.django-rest-framework.org/)
* [Docs | Flutter](https://docs.flutter.dev/)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |
|  |
|  |
|  |
|  |
|  |
|  | |  |  | |  |