

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**BÁO CÁO CUỐI KỲ  
MÔN THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG**

Giảng viên: Ngô Thị Duyên

**Nhóm 14**

Dương Thanh Hiền - 20020013

Nguyễn Trọng Tuế - 20020498

Lê Ngọc Minh - 20020058

Nguyễn Văn Dũng – 20020192

**Hà Nội - 2022**

## Mục lục

1.	Phát biểu bài toán .....	4
1.1.	Bài toán:.....	4
1.2.	Đối tượng sử dụng: .....	5
1.3.	Giải pháp:.....	5
2.	Thiết kế .....	5
2.1.	Bản thiết kế cuối cùng của giao diện: .....	5
3.	Những điểm quan trọng chính trong thiết kế .....	13
3.1.	Về tính dễ dùng: .....	13
3.2.	Về cách tổ chức hiển thị trong giao diện:.....	13
3.3.	Phân tích tác vụ: .....	14
3.4.	Thiết kế khái niệm: .....	16
3.4.1.	Danh sách Metaphors:.....	16
3.4.2.	Các ca sử dụng: .....	19
3.4.3.	Biểu đồ nội dung: .....	22
4.	Thảo luận về các thay đổi sau khi thực hiện ba phương pháp đánh giá .....	23
4.1.	Về các mẫu thiết kế (qua bản mẫu giấy): .....	23
4.2.	Thiết kế tổng thể: .....	24
4.3.	Về tính sử dụng (qua kinh nghiệm và thử nghiệm trên người dùng thật):.....	25
5.	Lập trình .....	25
5.1.	Mô tả cấu trúc chương trình, chức năng thực hiện: .....	25
5.1.1.	Tính năng:.....	25
5.1.2.	Mô hình cơ sở dữ liệu: .....	26
5.1.3.	Cấu trúc chương trình: .....	26

5.2. Các vấn đề trong thiết kế lập trình: .....	26
6. Kiểm thử: .....	27
7. Bài học: .....	28
7.1. Trong quá trình thực hiện dự án: .....	28
7.2. Điều làm khác đi nếu được làm lại: .....	29
8. Phân công công việc: .....	29

# 1. Phát biểu bài toán

## 1.1. Bài toán:

- *Tiền điện tử* (Crypto Currency) là một dạng tiền kỹ thuật số được giao dịch trên nền tảng online.
  - Các đơn vị tiền điện tử được tạo ra dựa trên nền tảng công nghệ dữ liệu chuỗi khối (blockchain) – một sổ cái công cộng không lộ liệt kê tất cả các giao dịch được xác thực bởi một hệ thống máy tính kết nối toàn cầu.
  - Tiền điện tử được xây dựng dựa trên những thuật toán phức tạp, trong đó cho phép các giao dịch được thực hiện trực tiếp giữa người gửi và người nhận mà không cần có sự kiểm soát của Chính phủ, ngân hàng hay các tổ chức tài chính mà vẫn đảm bảo tính an toàn và chính xác của giao dịch.
  - Hiện nay nền tài chính phi tập trung – với nhân tố chính là các loại tiền điện tử đang ngày càng phát triển. Một trong những công cụ chủ yếu của tài chính phi tập trung chính là sàn giao dịch phi tập trung.
- ⇒ Nhóm em muốn xây dựng một sàn giao dịch phi tập trung cơ bản nhất, chức năng cơ bản của ứng dụng này là chuyển tài sản điện tử của người dùng từ token này sang token khác với tỷ giá theo tỷ giá được cho trước (lấy theo dữ liệu trên các sàn giao dịch điện tử đã có sẵn trên internet).

## **1.2. Đối tượng sử dụng:**

- Tập trung vào những người đang có tài sản điện tử.
- Những người quan tâm đến tiền điện tử, có nhu cầu chuyển đổi giá trị qua lại giữa các loại tiền điện tử.
- Những đối tượng này thường là những người ít nhiều có hiểu biết về công nghệ.

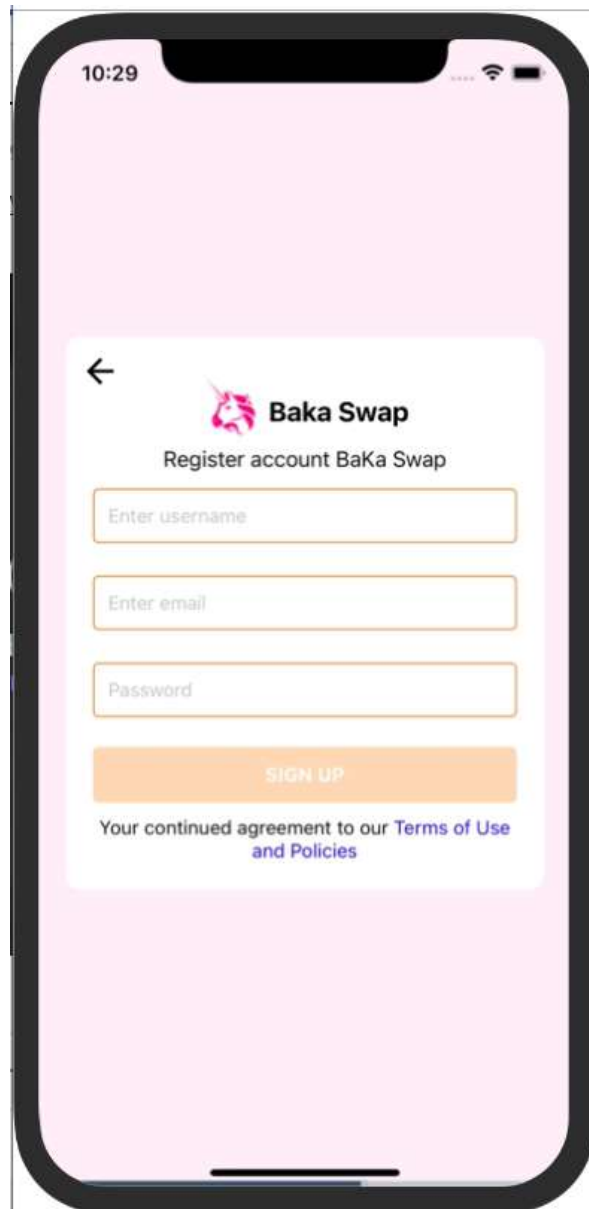
## **1.3. Giải pháp:**

- Sử dụng các công thức có trong thuật toán để swap 2 token bất kỳ trên sàn giao dịch.

# **2. Thiết kế**

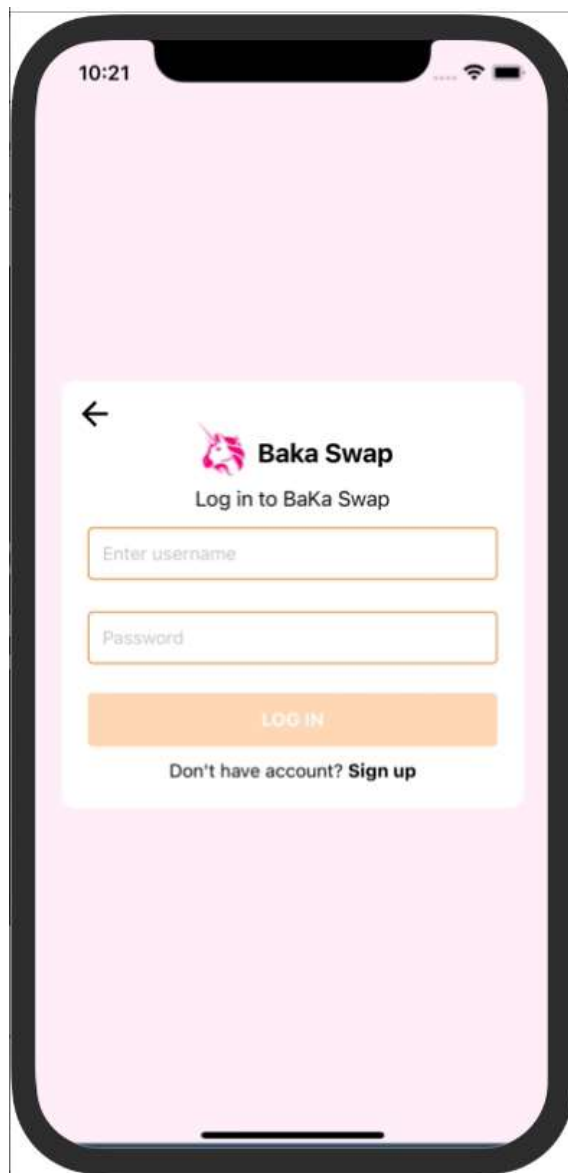
## **2.1. Bản thiết kế cuối cùng của giao diện:**

➤ Chức năng tạo tài khoản:

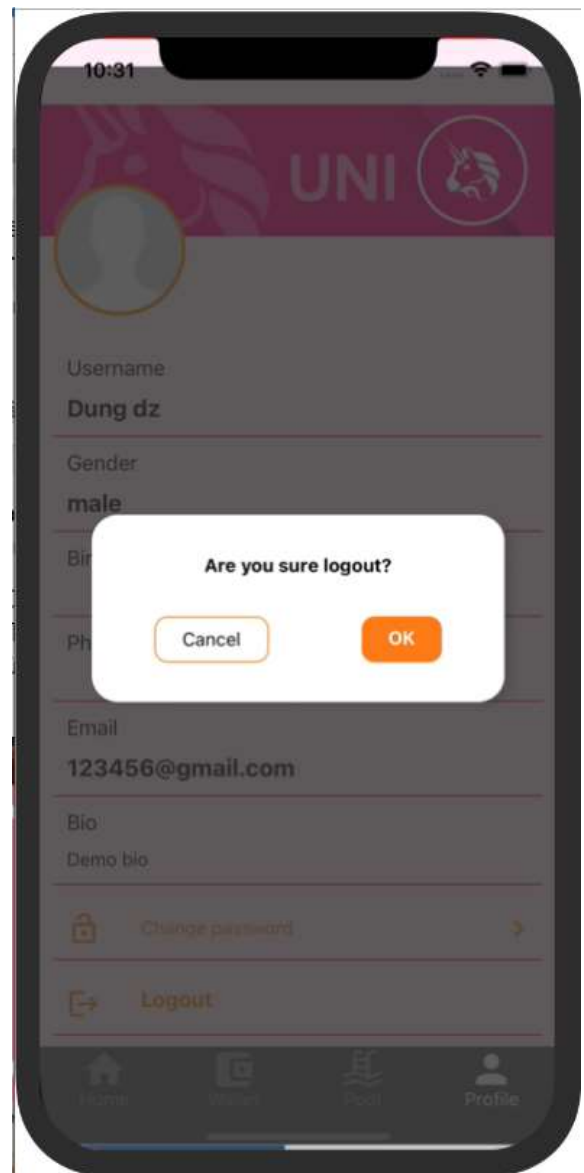


The image shows a mobile application interface for creating an account. At the top, the status bar displays the time 10:29, signal strength, and battery level. The app's header features a back arrow, a pink unicorn logo, and the text "Baka Swap". Below the header, the title "Register account BaKa Swap" is centered. The registration form consists of three input fields: "Enter username", "Enter email", and "Password". Below these fields is an orange "SIGN UP" button. At the bottom of the form, there is a line of text: "Your continued agreement to our [Terms of Use and Policies](#)".

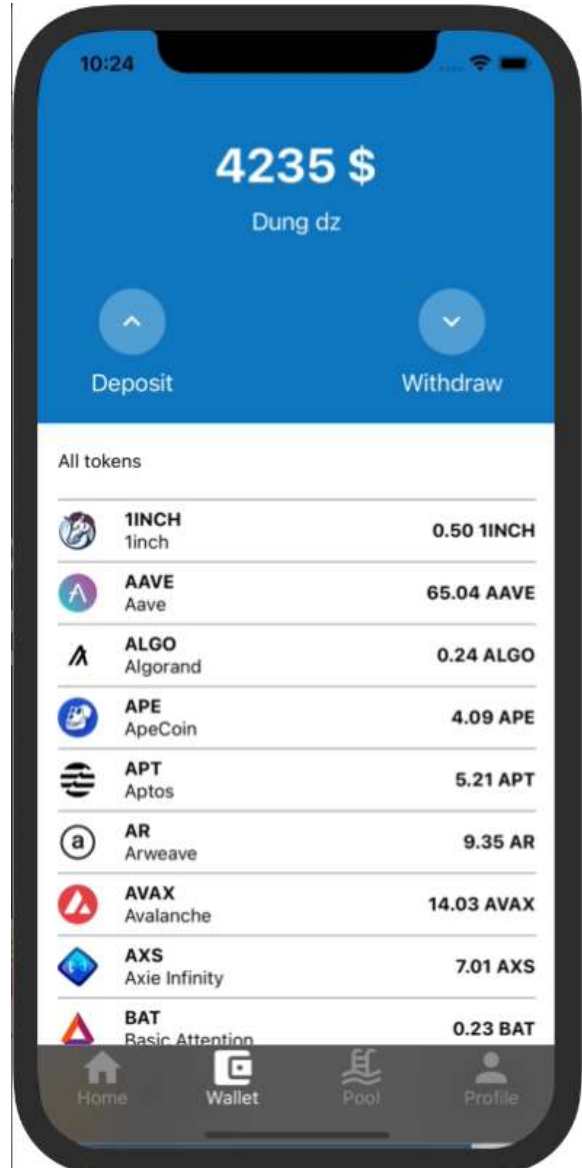
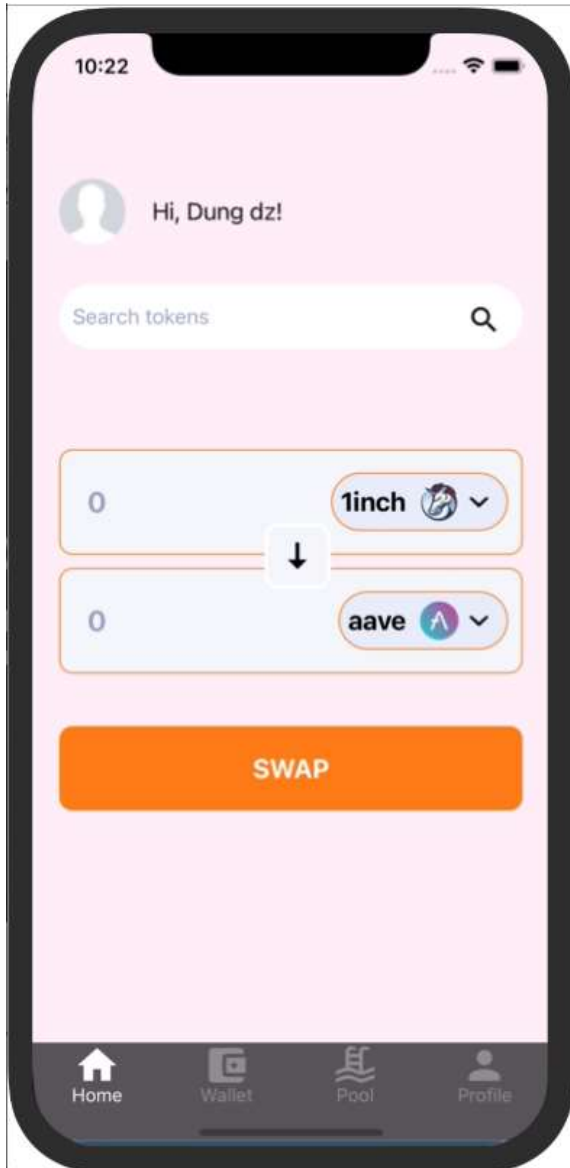
➤ Chức năng đăng nhập:



➤ Chức năng đăng xuất:

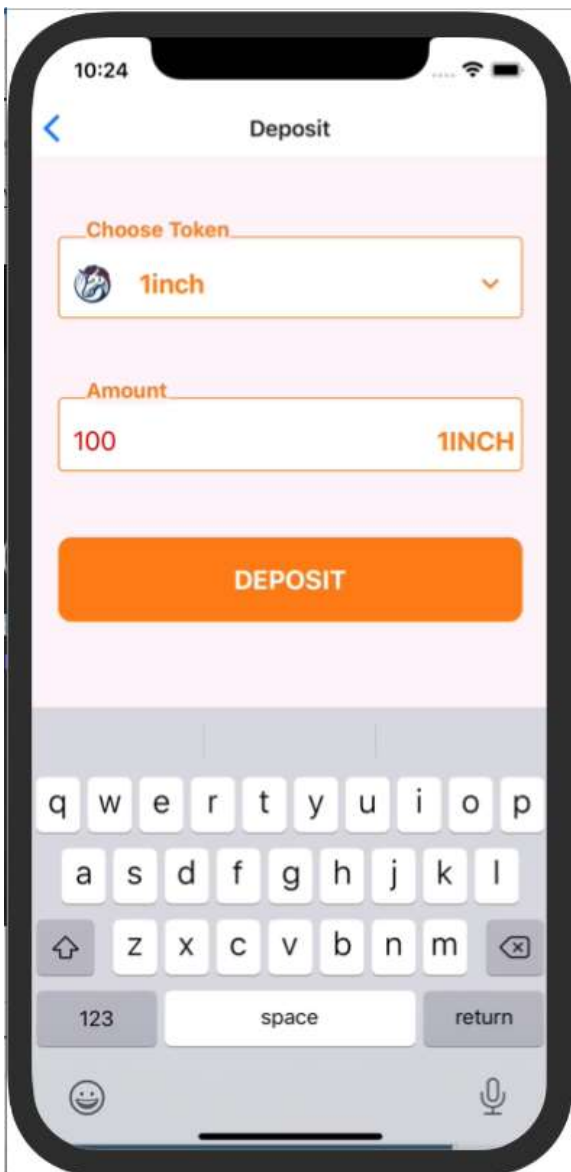


➤ Giao diện chính và ví:

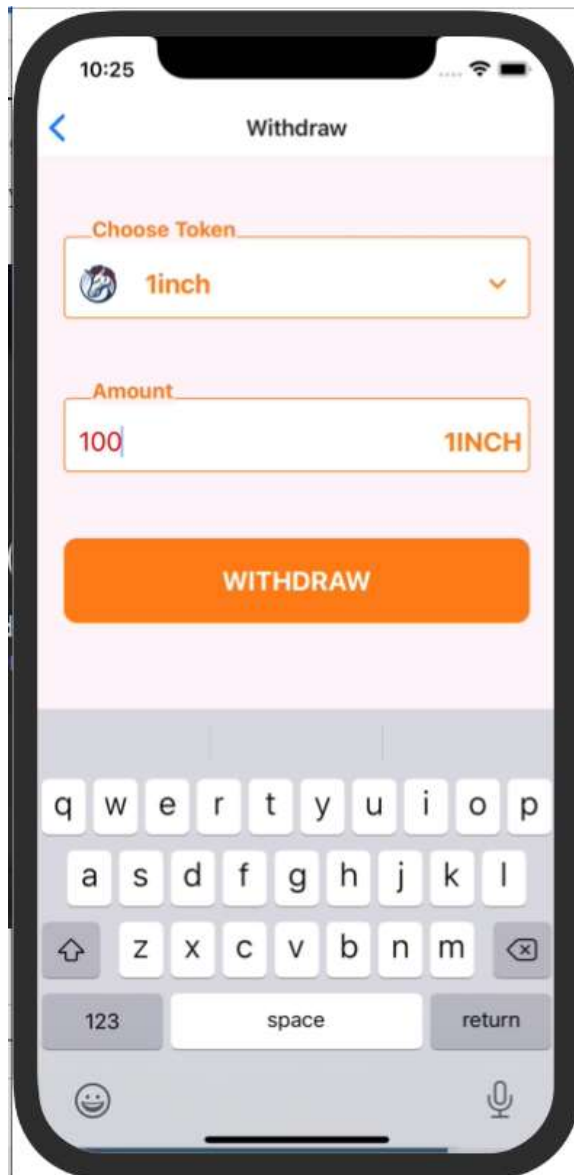




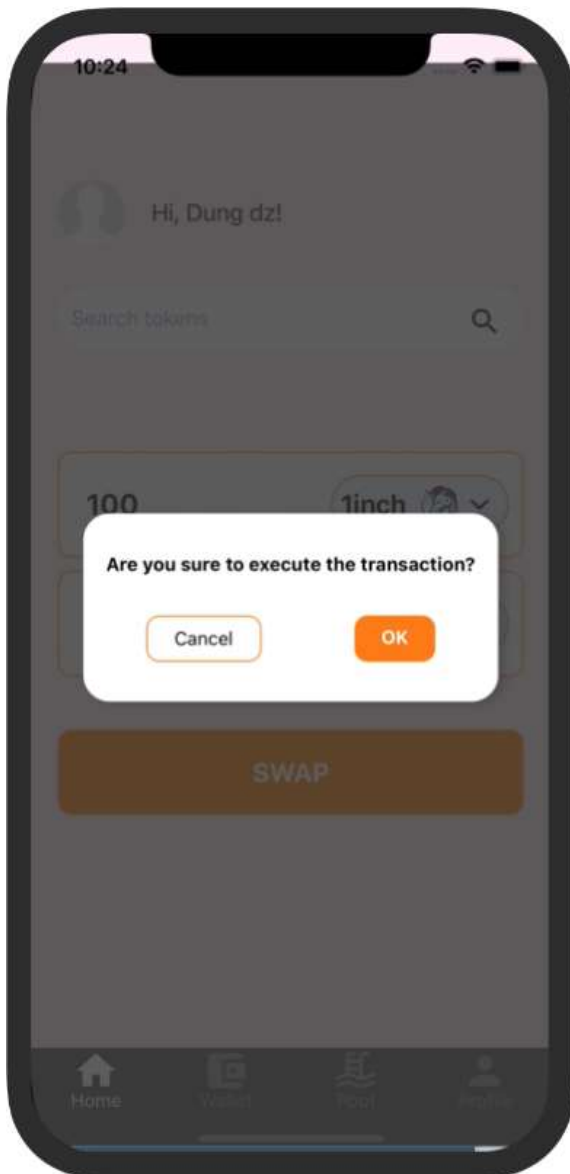
➤ Thêm tiền vào tài khoản:



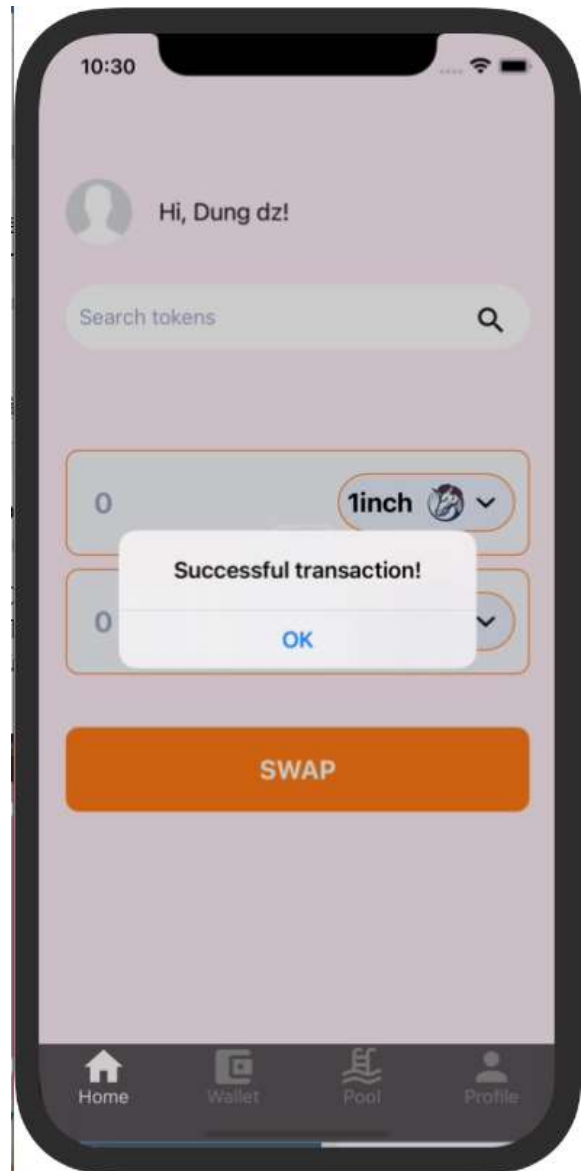
➤ Rút tiền ra khỏi tài khoản:



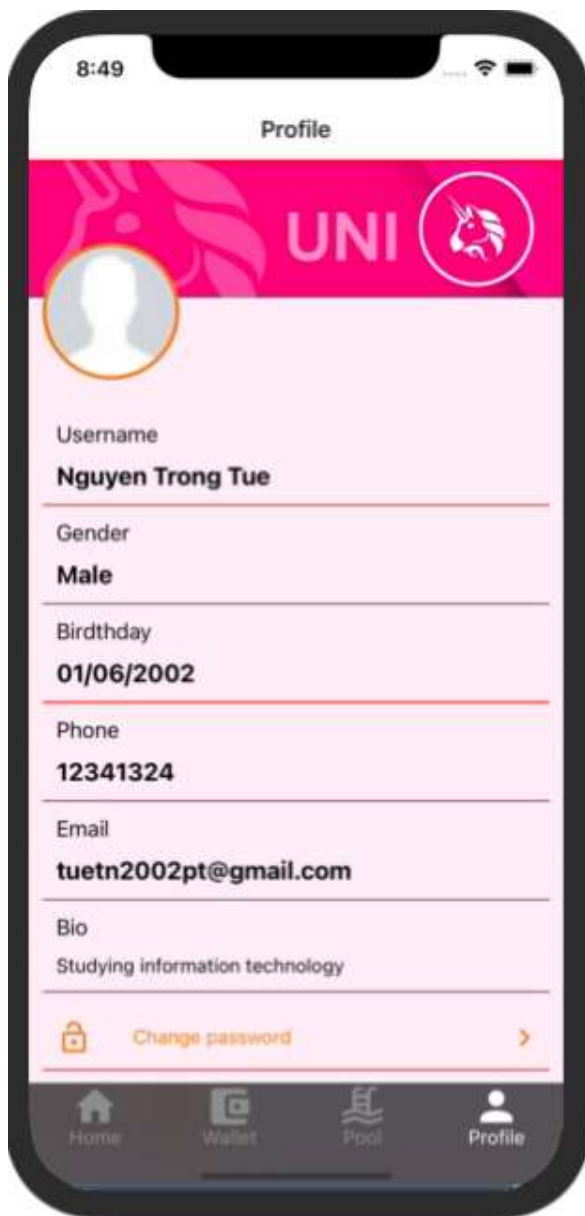
- Cảnh báo hành động đang thực hiện:



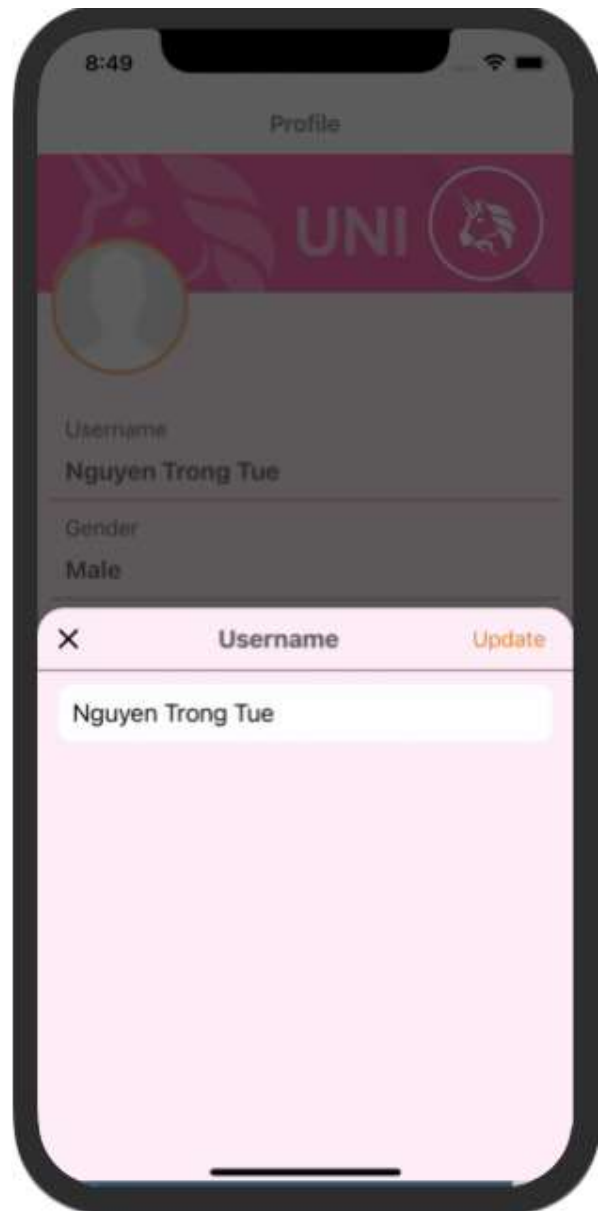
- Thông báo thực hiện hành động thành công:



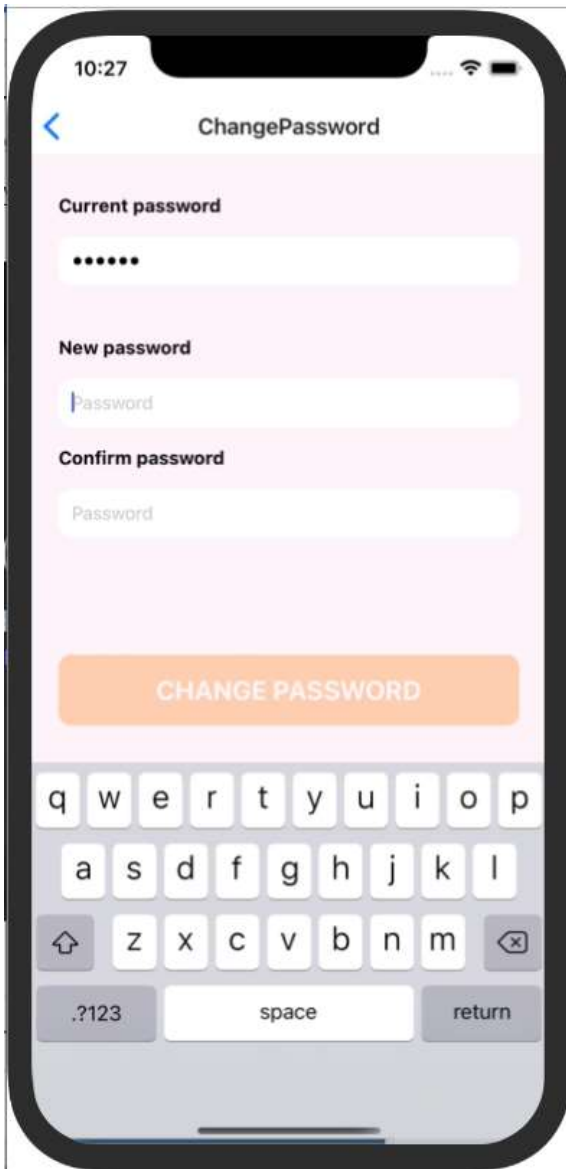
➤ Xem thông tin tài khoản:



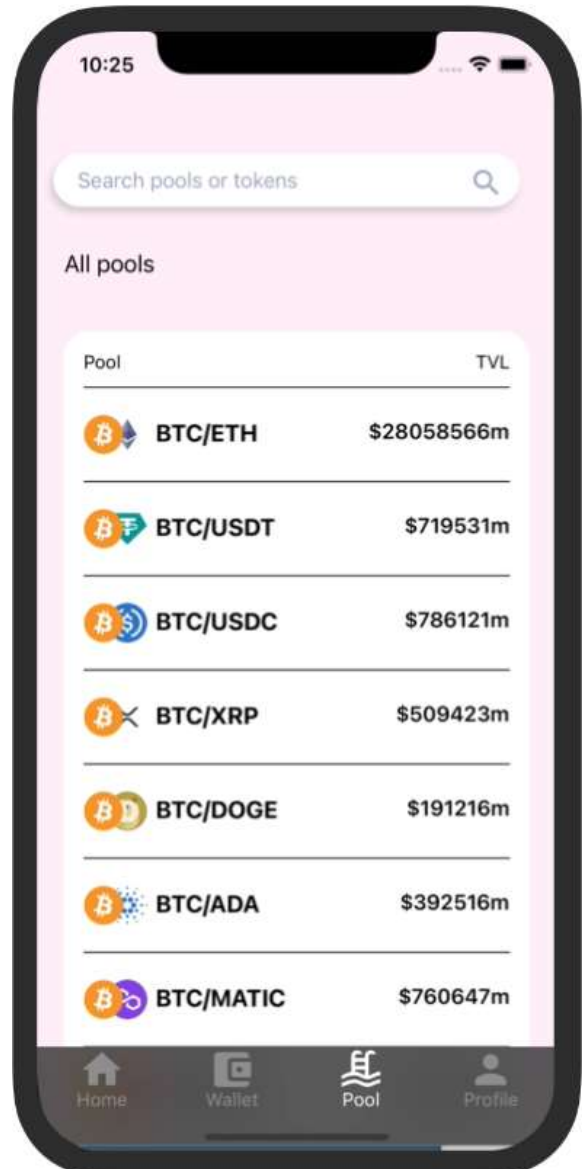
➤ Chỉnh sửa thông tin:



➤ Đổi mật khẩu:



➤ Pool:



### **3. Những điểm quan trọng chính trong thiết kế**

#### **3.1. Về tính dễ dùng:**

- Phù hợp để sử dụng: Khi người dùng muốn chuyển đổi qua lại giữa các loại tiền ảo.
- Dễ học: Dễ học cho các nhóm người sử dụng khác nhau.
- Các tác vụ hiệu quả: Người dùng thường xuyên có thể thực hiện nhanh các chức năng swap các loại token, nạp tiền, rút tiền khỏi tài khoản, quản lý tài khoản....
- Dễ nhớ: Người dùng không thường xuyên vẫn có thể nhớ được cách sử dụng.
- HÀi lòng chủ quan: Người dùng nói chung sẽ hài lòng về hệ thống bởi tính dễ dùng hiệu quả.
- Dễ hiểu: Người dùng dễ hiểu hệ thống hoặc từng module làm các chức năng khác nhau: Quản lý tài khoản, nạp tiền, rút tiền khỏi tài khoản....

#### **3.2. Về cách tổ chức hiển thị trong giao diện:**

- Tính nhất quán của dữ liệu hiển thị: Trong quá trình thiết kế giao diện, các thuật ngữ, từ viết tắt, màu sắc, chữ viết hoa...đều được đồng bộ và chuẩn hóa.
- Tính hiệu quả trong đồng bộ hóa thông tin người dùng: Hướng trái/phải, khoảng cách, nhãn... format của thông tin gắn với khuôn dạng dữ liệu sử dụng, và có quan hệ với dữ liệu được yêu cầu.
- Tối ưu hóa sự ghi của người sử dụng:

- Không có yêu cầu nhớ từ một màn hình hiển thị này tới màn hình khác: Màn hình hiển thị các chức năng, các hộp thoại thông báo... đều rõ ràng, không có quá nhiều thông tin.
  - Sử dụng ít các thao tác hành động cho mỗi công việc: Mỗi thao tác cho các chức năng liên quan tới tài khoản, nạp và rút tiền... đều chỉ cần 1 đến 2 nút bấm là thực hiện được.
  - Nhấn và các kiểu thông dụng cho người dùng chưa có kinh nghiệm: Các icon được sử dụng cho từng chức năng đều dễ nhìn, phong phú...
- Khả năng tương thích giữa dữ liệu hiển thị với dữ liệu vào: Trường output chỉnh sửa phù hợp với trường input trong mục chỉnh sửa thông tin, mục đăng nhập, đăng ký...
- Tính linh hoạt cho người sử dụng kiểm soát dữ liệu hiển thị: Người sử dụng có thể lấy thông tin về token họ đang làm một cách dễ dàng nhất.

### **3.3. Phân tích tác vụ:**

Sau khi xác định các đặc điểm của người dùng ứng dụng, nhóm bắt đầu phân tích và lên ý tưởng các tác vụ có trong bài toán nêu ra. Tại bước này, nhóm tập trung vào các bước thực hiện các tác vụ đã nêu (thông qua các kịch bản tác vụ).

Dưới đây là bảng các tác vụ mà nhóm đã lên ý tưởng để thực hiện:

<b>Tên tác vụ</b>	<b>Nội dung tác vụ</b>	<b>Kịch bản tác vụ</b>
Tạo tài khoản	Người dùng tạo tài khoản cho bản thân bao gồm các thông tin cần thiết.	Mở ứng dụng lên, bấm nút tạo tài khoản, điền thông tin cần thiết vào các trường ( tên đăng nhập, mật khẩu, họ tên, ngày tháng năm sinh, ...), nhấn xác nhận để tạo tài khoản.
Đăng nhập	Người dùng đăng nhập tài khoản đã được đăng ký của bản thân.	Để thực hiện, bấm nút đăng nhập và điền tài khoản đã tạo trước đó vào ô tên đăng nhập và mật khẩu.
Đăng xuất	Người dùng đăng xuất tài khoản đã được đăng nhập của bản thân.	Nhấn vào tài khoản, chọn nút đăng xuất để thoát tài khoản khỏi ứng dụng.
Chỉnh sửa thông tin	Người dùng cập nhật, chỉnh sửa những thông tin muốn sửa đổi.	Đăng nhập vào tài khoản, nhấn nút chỉnh sửa, chỉnh sửa các thông tin người dùng muốn (mật khẩu, thông tin người dùng...).
Thêm tiền vào tài khoản	Người dùng có thể thêm bất kì loại tiền với số lượng mong muốn vào tài khoản của mình.	Đăng nhập vào tài khoản, nhấn nút nạp tiền, chọn loại token và số lượng mong muốn, sau đó xác nhận.
Rút tiền ra khỏi tài khoản	Người dùng rút một lượng tiền đang có trong tài khoản.	Đăng nhập vào tài khoản, nhấn nút rút tiền, chọn loại token

		và số lượng muốn rút, sau đó xác nhận.
Xem trạng thái tài khoản	Người dùng có thể xem thông tin cá nhân của mình, những loại token mình đang nắm giữ và số lượng của chúng.	Đăng nhập vào tài khoản, nhấn nút hồ sơ người dùng, xem thông tin trạng thái tài khoản hiện tại.
Chọn loại token muốn swap	Người dùng chọn một loại token bất kỳ có trong tài khoản của mình để đổi sang loại token bất kỳ khác.	Đăng nhập vào tài khoản, chọn mục swap, chọn loại token muốn chuyển đổi và loại token muốn chuyển đổi thành.
Chuyển đổi token	Chuyển đổi giá trị 2 loại token đã chọn theo tỷ giá được quy định tại pool chứa 2 loại token đó.	Sau khi chọn loại token muốn swap, nhập số lượng token được chuyển đổi, bấm nút swap để chuyển đổi thành giá trị của token đích.

### 3.4. Thiết kế khái niệm:

Phần này nhóm bắt đầu quy trình thiết kế ứng dụng, các kịch bản tác vụ và các ca sử dụng quan trọng sẽ được chuyển đổi sang các thiết kế khái niệm. Ngoài ra, nhóm tiếp tục tập trung vào các tác vụ ở mức trừu tượng, bao gồm hành động đầu vào từ người dùng, hành động đầu ra của hệ thống.

#### 3.4.1. Danh sách Metaphors:

Ở phần này, nhóm đưa ra danh sách các ẩn dụ của từng kịch bản tác vụ sử dụng trong giao diện của ứng dụng:



Kịch bản tác vụ	Ân dụ
Tạo tài khoản	<ul style="list-style-type: none"> <li>➤ Ân dụ bằng cách: nếu trường nhập đúng thì màu nền của ô input có màu xanh lá cây, có màu đỏ trong trường hợp người dùng nhập dữ liệu lỗi.</li> <li>➤ Đưa ra thông báo chào mừng người dùng tạo tài khoản thành công. Thông báo cần có màu sắc sáng, chữ rõ ràng, có hình vẽ con người vui vẻ chào mừng, có hình vẽ hoặc icon mặt buồn trong trường hợp thất bại.</li> </ul>
Đăng nhập	<ul style="list-style-type: none"> <li>➤ Ân dụ bằng cách: Khi người dùng nhập xong tên tài khoản và đăng nhập, hệ thống xuất ra màn hình thông báo, tích xanh nếu đăng nhập thành công, tích đỏ nếu sai tên tài khoản hoặc tài khoản không tồn tại.</li> </ul>
Đăng xuất	<ul style="list-style-type: none"> <li>➤ Ân dụ bằng cách tô đỏ nút đăng xuất, để người dùng tránh nhầm lẫn với các chức năng khác. Hộp thông báo đăng xuất bố trí icon dấu chấm than để người dùng xác nhận lại lần nữa việc đăng xuất tài khoản.</li> </ul>
Chỉnh sửa thông tin tài khoản	<ul style="list-style-type: none"> <li>➤ Nếu trường chỉnh sửa đúng với input thì có dấu tích xanh hoặc dấu tích đỏ nếu thông tin nhập vào sai. Điều này làm người dùng dễ nắm bắt</li> </ul>

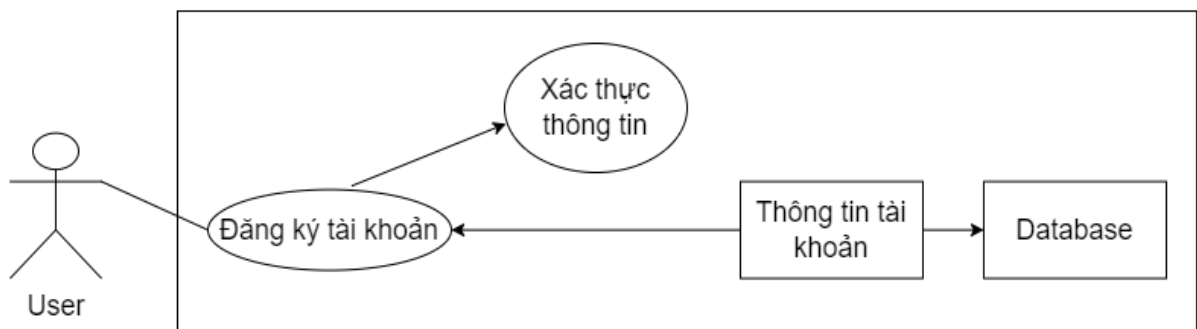
	<p>được thông tin họ chỉnh sửa có chính xác hay không.</p> <ul style="list-style-type: none"> <li>➤ Đưa ra các icon tương tự như tiêu đề của các trường sửa đổi ngoài tên tiêu đề của trường.</li> <li>➤ Đưa ra âm thanh vừa đủ khi người dùng có muốn xác nhận đổi thay đổi thông tin hay không</li> </ul>
Thêm tiền vào tài khoản	<ul style="list-style-type: none"> <li>➤ Ấn dụ bằng icon mũi tên màu xanh đi lên, hình ảnh token tương ứng với loại token người dùng đang chọn để người dùng dễ nắm bắt loại tiền mình muốn thêm đã chính xác hay chưa.</li> <li>➤ Đưa ra âm thanh “ting ting” để thông báo thêm tiền thành công, người dùng có thể an tâm hơn về việc tiếp nhận của hệ thống.</li> </ul>
Rút tiền khỏi tài khoản	<ul style="list-style-type: none"> <li>➤ Ấn dụ bằng icon mũi tên màu đỏ đi xuống, hình ảnh token tương ứng với loại token người dùng muốn rút, có thông báo đỏ nếu người dùng rút quá số lượng, thông báo xanh nếu số lượng rút thỏa mãn.</li> <li>➤ Có thể có thêm hình ảnh dấu chấm than nếu số lượng rút của người dùng vượt quá ngưỡng an toàn, tức là số</li> </ul>

	token đó trong pool quá chênh lệch.
Chọn loại token muốn chuyển đổi	➤ Ấn dụ bằng icon mũi tên đi xuống khác, bấm vào hiện ra bảng danh sách các loại token để người dùng lựa chọn, mỗi loại token hiển thị theo hàng với các thông tin: Logo, tên token, mã.
Chuyển đổi giá trị token	➤ Ấn dụ bằng 2 textbox để hiện giá trị hai loại token được chuyển đổi. ➤ Nút “Swap” ở giữa 2 textbox là icon mũi tên 2 chiều

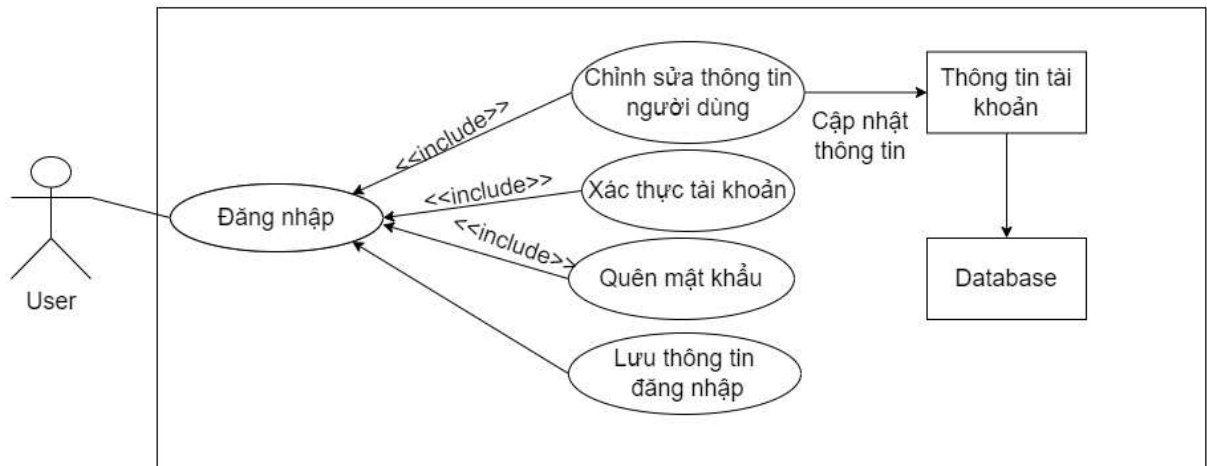
### 3.4.2. Các ca sử dụng:

Ở phần này, nhóm xây dựng mỗi kịch bản tác vụ thành một ca sử dụng, bao gồm chức năng, liên kết, đối tượng và các ràng buộc.

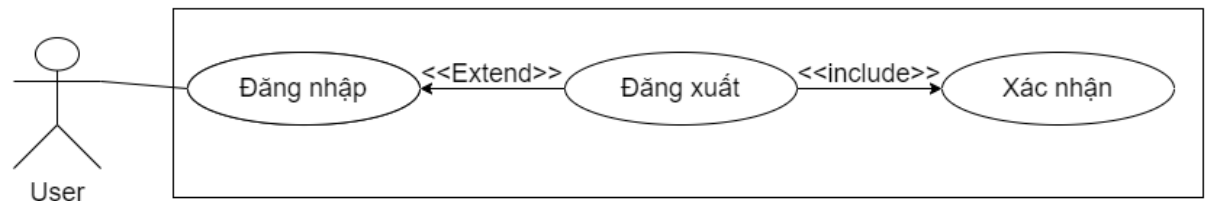
#### ➤ Đăng ký tài khoản:



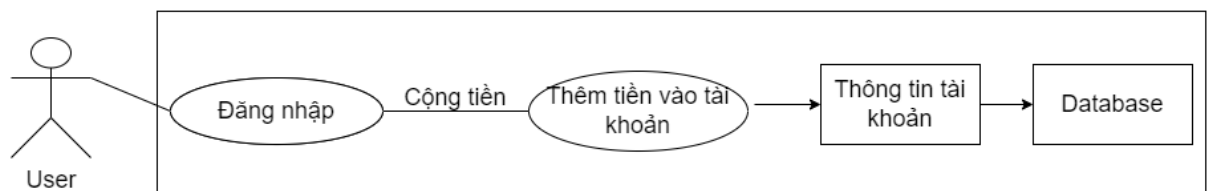
➤ Đăng nhập:



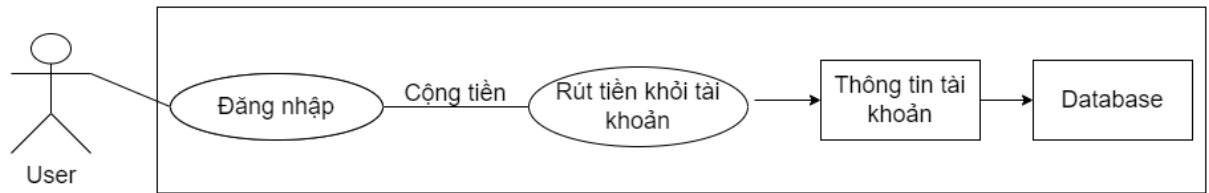
➤ Đăng xuất:



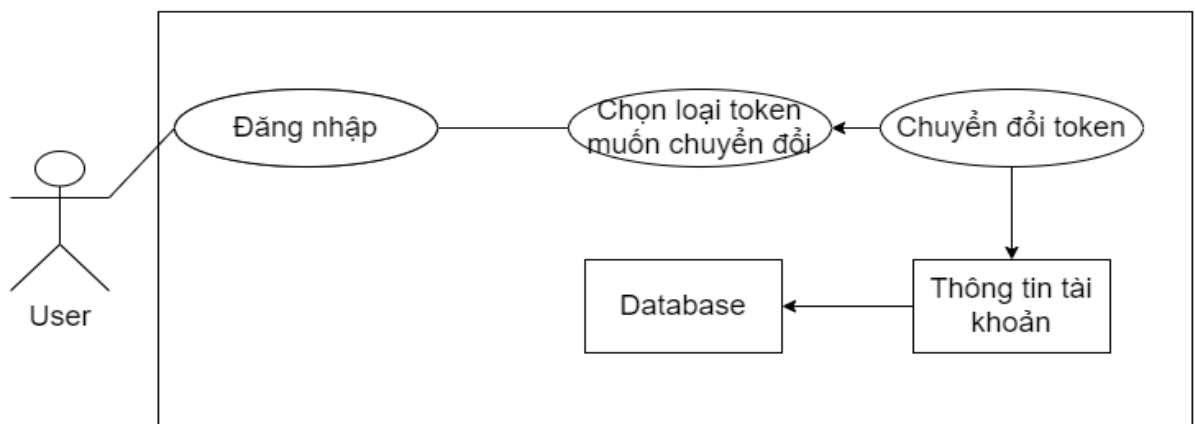
➤ Thêm tiền vào tài khoản:



➤ Rút tiền khỏi tài khoản:

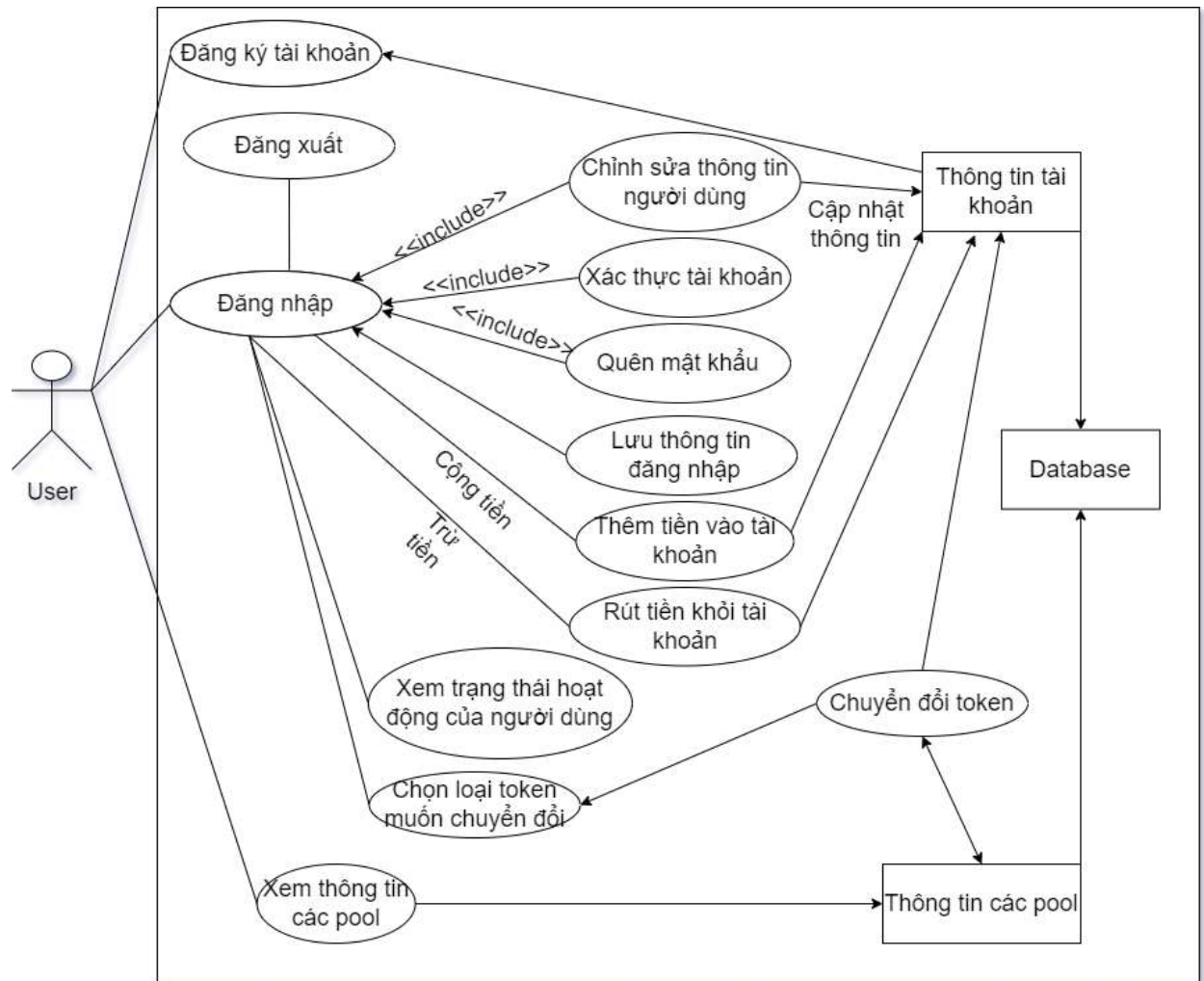


➤ Đổi giá trị token:



### 3.4.3. Biểu đồ nội dung:

Rút gọn nội dung của các ca sử dụng vào 1 biểu đồ nội dung:



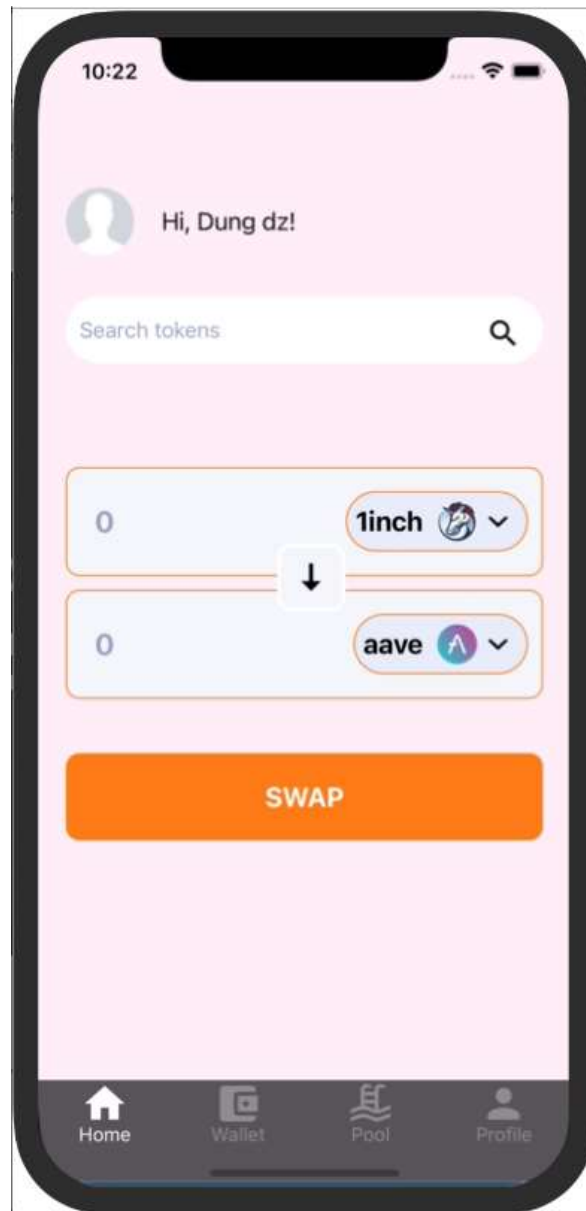
## 4. Thảo luận về các thay đổi sau khi thực hiện ba phương pháp đánh giá

### 4.1. Về các mẫu thiết kế (qua bản mẫu giấy):



- Mẫu thiết kế 1: Do màu sắc quá sặc sỡ, phối màu khó nhìn. Phần điều hướng trùng màu với màu nền, khó phân biệt ⇒ không chọn mẫu thiết kế này.
- Mẫu thiết kế 2: Phần điều hướng đặt dọc chiếm không gian hiển thị, không thuận tiện cho thao tác trên nền tảng mobile ⇒ không chọn mẫu thiết kế này.
- Mẫu thiết kế 3: Khắc phục được các lỗi ở 2 mẫu thiết kế trước ⇒ chọn mẫu thiết kế này

## 4.2. Thiết kế tổng thể:



Đây là giao diện chính sau khi mở ứng dụng (trường hợp đã đăng nhập). Giao diện bao gồm những chức năng:

- Phần đầu: Hiển thị tên người dùng và thanh tìm kiếm.
- Phần nội dung: là chức năng chính của ứng dụng (chuyển đổi token).
- Thanh điều hướng: đặt ở phía dưới màn hình, gồm 4 tab: Home (đang được chọn), Wallet (để thực hiện các chức năng thêm tiền,



rút tiền), Pool (để xem trạng thái token hiện tại) và Profile (để xem và chỉnh sửa thông tin tài khoản).

#### **4.3. Về tính sử dụng (qua kinh nghiệm và thử nghiệm trên người dùng thật):**

- Ban đầu ứng dụng phản hồi khá chậm (0,5s tới 0,6s), sau khi kiểm tra và tối ưu các dòng code, tốc độ phản hồi chỉ còn 0,05s tới 0,06s.
- Số lượng dữ liệu (token) ban đầu trả về cho người sử dụng chưa chính xác, sau khi phân tích và sửa lại hệ thống cơ sở dữ liệu, ứng dụng đã đưa ra được số lượng token chính xác.
- Ban đầu ứng dụng có một bài bug nhỏ: Cộng tiền vô hạn, rút tiền bị âm.... sau khi qua người dùng thật, tất cả đều đã được sửa lỗi.

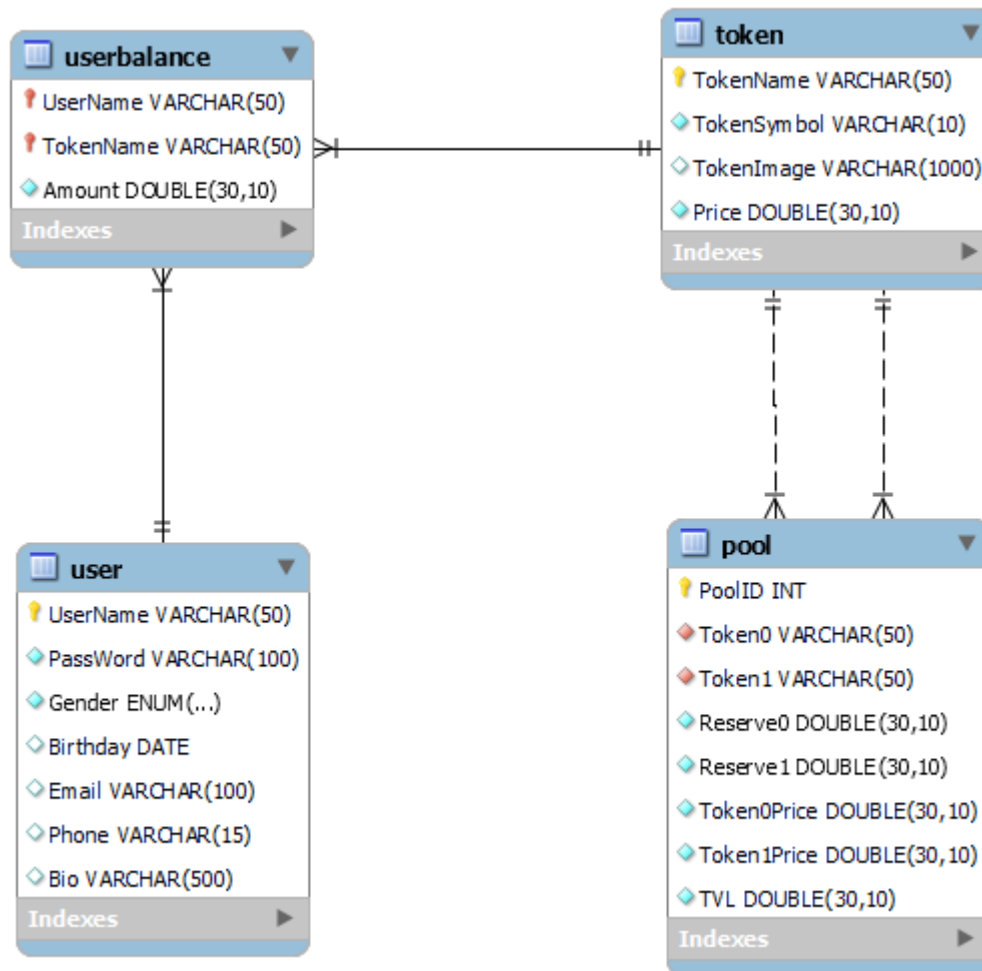
### **5. Lập trình**

#### **5.1. Mô tả cấu trúc chương trình, chức năng thực hiện:**

##### **5.1.1. Tính năng:**

- Người dùng có thể tạo tài khoản, chỉnh sửa thông tin tài khoản như ngày sinh, giới tính, sđt, email, bio.
- Người dùng có thể nạp token hoặc rút token ra khỏi ví.
- Người dùng có thể chuyển đổi giữa các loại token với nhau.

### 5.1.2. Mô hình cơ sở dữ liệu:



### 5.1.3. Cấu trúc chương trình:

- `main.py`: file main xây dựng FastAPI service, định nghĩa cấu trúc các API.
- `crud.py`: Định nghĩa các method (read, write) để kết nối đến MySQL.
- `database.py`: Kết nối MySQL.
- `models.py`: Định nghĩa class object models cho FastAPI.
- `schemas.py`: Định nghĩa lược đồ để làm việc với yêu cầu/phản hồi API cụ thể.

## 5.2. Các vấn đề trong thiết kế lập trình:

- Nhiều tính năng chuyển từ bản mẫu sang lập trình còn khó khăn.
- Giao diện trong lúc lập trình phải bổ sung khá nhiều so với bản thiết kế cho phù hợp với logic.

## 6. Kiểm thử:

- Các câu hỏi trong buổi phỏng vấn:
  - Bạn cảm thấy giao diện ứng dụng như thế nào ?
  - Bạn có thấy các chức năng dễ nhớ, dễ sử dụng không ?
  - Bạn có đóng góp gì về giao diện cho ứng dụng không ?
  - Bạn có muốn cải thiện thêm gì về chức năng cho ứng hay không?
  - Bạn thấy ứng dụng này có cần thiết đối với bản thân bạn hay không?
- Thực hiện phỏng vấn trên 2 người dùng sau khi đã được sự đồng ý của người dùng và đã mô tả ngắn gọn về ứng dụng cho người dùng hiểu, kết quả thu được như sau:

### ➤ Người dùng 1:

- **Địa Điểm:** sân trường Đại học Công nghệ - ĐHQGHN.
- **Thời gian:** khoảng 10h sáng ngày 01/11/2022.
- **Thông tin người phỏng vấn:** bạn Hùng - sinh viên năm 3, đang theo học tại đây.
- **Hành động người dùng:** Người dùng đăng ký tài khoản và thực hiện chức năng rút tiền, chuyển đổi token.
- **Ưu điểm về tính dễ dùng:**
  - Giao diện nhìn chung thân thiện, dễ dùng.
  - Giao diện chuyển đổi token bắt mắt, dễ sử dụng, đặc biệt nút chuyển đổi từ trên xuống dưới thuận tiện cho người dùng.
- **Vấn đề về tính dễ dùng:**
  - Có chỗ đang bị sai chính tả (Withdraw).

- Ở phần xem và edit profile đang chưa có sự thống nhất: Kích thước tiêu đề của 2 mục chưa đều nhau.

- **Gợi ý về giải pháp:** Sửa lại source code trong phần Front-end.

### ➤ **Người dùng 2:**

- **Địa Điểm:** Phỏng vấn qua zoom.
- **Thời gian:** khoảng 21h tối ngày 01/11/2022.
- **Thông tin người phỏng vấn:** bạn Hà - sinh viên năm 3, đang theo học Đại học Ngoại Ngữ - ĐHQGHN.
- **Hành động người dùng:** Người dùng đăng ký tài khoản và thực hiện chức năng nạp tiền, rút tiền và chuyển đổi token.
- **Ưu điểm về tính dễ dùng:**
  - Giao diện nạp và rút tiền đơn giản, dễ ghi nhớ trong lần đầu sử dụng.
  - Bố cục ứng dụng hợp lí, dễ nhìn. Các icon của các chức năng sinh động, giúp ghi nhớ các chức năng dễ dàng.
- **Vấn đề về tính dễ dùng:**
  - Trong giao diện nạp và rút tiền, phần tên của ví (Baka-wallet) đang bị thiết kế như là 1 textbox, người dùng dễ nhầm tưởng đó là trường thông tin có thể thay đổi được.
  - Tiêu đề ở phần nạp và rút tiền chưa đồng bộ (1 bên là baka-wallet, 1 bên là baka-feature).
- **Gợi ý về giải pháp:** Sửa lại source code trong phần Front-end.

## **7. Bài học:**

### **7.1. Trong quá trình thực hiện dự án:**

- Tính đoàn kết, khả năng làm việc nhóm của bản thân: Tôn trọng ý kiến của các thành viên trong nhóm, tích cực xây dựng ý tưởng, đưa

ra giải pháp giải quyết các vấn đề phát sinh trong quá trình thực hiện dự án.

- Học hỏi thêm kinh nghiệm thông qua việc làm 1 dự án hoàn chỉnh:  
Từ các khâu đơn giản đến phức tạp, mọi thứ đều phải thực hiện tỉ mỉ để cuối cùng cho ra sản phẩm hoàn thiện nhất.

## 7.2. Điều làm khác đi nếu được làm lại:

⇒ Chọn lại đề tài cho sản phẩm bởi sản phẩm hiện tại có giao diện tương đối đơn giản.

## 8. Phân công công việc:

STT	Tên thành viên	Đóng góp (%)	Công việc đã làm
1	Dương Thanh Hiền (trưởng nhóm)	27	Code phần Back-end, tham gia làm báo cáo từng tuần
2	Nguyễn Trọng Tuế	27	Code phần Front-end, xây dựng base dự án, tham gia làm báo cáo từng tuần
3	Lê Ngọc Minh	23	Phân tích thiết kế, hỗ trợ hoàn thiện giao diện, tổng hợp báo cáo kết quả
4	Nguyễn Văn Dũng	23	Phân tích thiết kế, call API, chuẩn bị slide, tham gia làm báo cáo từng tuần

