



BÀI TẬP THỰC HÀNH 2

CUSTOM COMPONENTS, REACT NAVIGATION

Bài 1: Social Media Feed (cont.)

1.1. Mục tiêu

Biết được khả năng của Props trong việc tạo ra các thành phần tùy chỉnh (Custom Components). Biết cách chia ứng dụng thành các components nhỏ hơn, có thể tái sử dụng. Biết cách sử dụng Props một cách hiệu quả để truyền dữ liệu giữa các components.

1.2. Mô tả

Trong BTTH1 – Bài tập 1, chúng ta đã xây dựng một ứng dụng mô phỏng các bài đăng trên mạng xã hội. Ở bài tập này, chúng ta sẽ tiến hành chia ứng dụng đã xây dựng thành các components con, nhằm mục đích tái sử dụng, tăng tính dễ đọc, dễ bảo trì và sửa lỗi.

1.3. Yêu cầu

- **Props:** Chia chương trình thành các components nhỏ hơn, có thể tái sử dụng. Sử dụng Props để truyền dữ liệu từ component gốc tới các components con.
- **State:** Sử dụng Hook useState để quản lý thông tin của các bài đăng.
- Giao diện và các chức năng của ứng dụng được giữ nguyên, tương tự như trong BTTH1 – Bài tập 1.

1.4. Gợi ý

- Phân tích số thành phần chính trong bố cục của giao diện.
- Xem xét các components có tính lặp lại để xây dựng thành một component riêng. Trong các component con riêng, tiếp tục xem xét việc lặp lại nếu có để xây dựng các components con nhỏ hơn.
- Có thể tạo thêm các props trong từng component riêng để việc truyền dữ liệu được dễ dàng.
- Áp dụng các styles đã thiết kế ở BTTH1 sang từng component con cho phù hợp để giữ được giao diện.

1.5. Hình ảnh ví dụ

- Tương tự các hình ảnh trong BTTH1 – Bài tập 1.



Bài 2: Settings and Feedback

2.1. Mục tiêu

Hiểu được khả năng của Props và State trong việc tạo ra các module React Native có thể tương tác. Luyện tập cách chia ứng dụng thành các components nhỏ hơn, có thể tái sử dụng. Luyện tập cách sử dụng Props một cách hiệu quả để truyền dữ liệu giữa các components và cách sử dụng State để quản lý các giá trị động trong một component duy nhất.

2.2. Mô tả

Trong bài tập này, chúng ta sẽ thiết kế và hiện thực một giao diện cho phần Cài đặt và phản hồi của một ứng dụng di động. Giao diện bao gồm logo và tên ứng dụng, các điều khiển bật tắt để bật chế độ tối và thông báo, trường nhập phản hồi để người dùng cung cấp phản hồi và một phần hiển thị các phản hồi đã gửi (FAQs).

2.3. Yêu cầu

- Thiết lập phần logo và tên ứng dụng nằm trên cùng của ứng dụng và căn giữa.
- Tạo công tắc bật tắt chế độ tối để thay đổi màu nền và màu văn bản của giao diện.
 - Công tắc bật → Nền tối và Chữ sáng
 - Công tắc không bật → Nền sáng và Chữ tối (Áp dụng mặc định khi khởi động ứng dụng)
- Tạo công tắc bật tắt thông báo cho phép người dùng bật hoặc tắt thông báo.
 - Công tắc bật → Hiển thị thông báo khi người dùng gửi phản hồi (Áp dụng mặc định khi khởi động ứng dụng)
 - Công tắc không bật → Không hiển thị thông báo khi người dùng gửi phản hồi
- Đảm bảo rằng hai công tắc hoạt động độc lập nhau.
- Một trường cho phép người dùng nhập thông tin phản hồi với nhiều dòng (Có placeholder để người dùng dễ nhận biết).
- Một nút “**Send Feedback**” để gửi phản hồi mà người dùng đã nhập:
 - Sau khi nhấn gửi, phản hồi của người dùng sẽ xuất hiện ở phần FAQs và trường nhập phản hồi được reset.
 - Nếu thông tin trong trường nhập phản hồi là rỗng, tức người dùng không nhập phản hồi, thì khi nhấn nút gửi ứng dụng sẽ không thực hiện hành động gì.



- Một phần FAQs để hiển thị các phản hồi của người dùng:
 - Các phản hồi của người dùng sẽ được hiển thị tại đây theo thứ tự phản hồi mới nhất hiển thị trên cùng, tiếp theo là các phản hồi trước đó.
- **Props:** Chia chương trình thành các components nhỏ hơn, có thể tái sử dụng. Sử dụng Props để truyền dữ liệu từ component gốc tới các components con.
- **State:** Sử dụng Hook useState để quản lý hành vi động của giao diện.
- Sử dụng StyleSheet để tạo style cho các components bằng các styles được cung cấp sẵn và tùy chỉnh thêm theo mong muốn, không sử dụng Inline Style.
- Đảm bảo rằng thông báo chỉ được hiển thị khi công tắc thông báo được bật và người dùng có thể cuộn màn hình xuống để xem nếu có nhiều phản hồi.

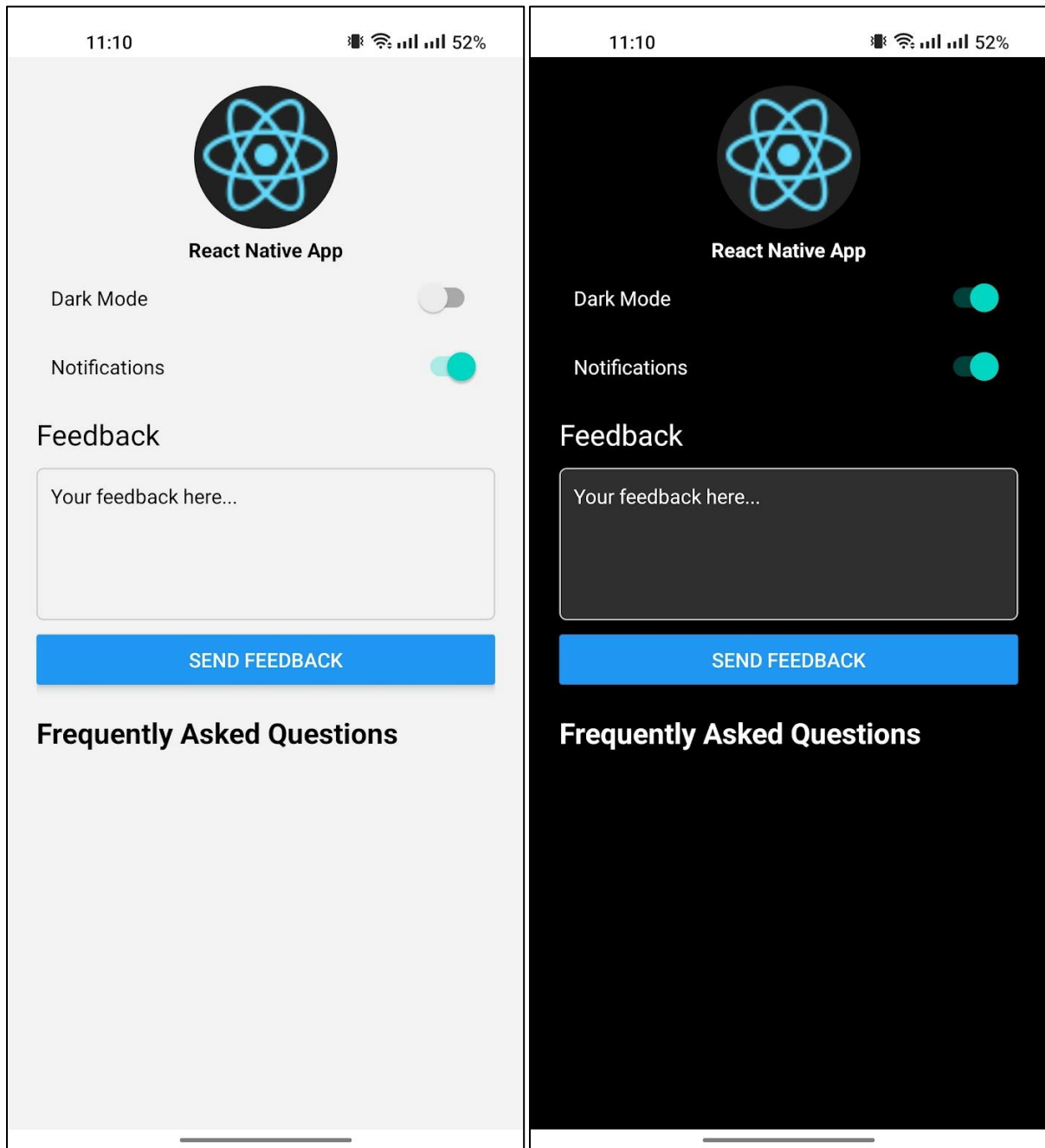
2.4. Gợi ý

- Sử dụng component “**Switch**” để xử lý các công tắc.
- Sử dụng component “**Alert**” để xử lý thông báo.
- Phân tích số thành phần chính trong bố cục của giao diện.
- Xem xét các components có tính lặp lại. Tạo các tệp riêng biệt cho từng component, cấu trúc các components rõ ràng, dễ đọc, đảm bảo mỗi component tập trung vào một công việc duy nhất.
- Sử dụng Props: Trong mỗi component con mà bạn tạo, hãy khai thác sức mạnh của Props để truyền dữ liệu từ thành phần gốc tới các thành phần con. Ví dụ, sử dụng Props để xử lý tiêu đề, giá trị và sự kiện cho mỗi công tắc.
- Sử dụng Hook useState để quản lý từng phần dữ liệu động. Ví dụ, sử dụng state để quản lý việc chuyển đổi chế độ tối.
- Sử dụng Style có điều kiện để điều chỉnh giao diện người dùng của bạn cho phù hợp với các cài đặt và chế độ khác nhau.
- Lưu ý: Một số components sẽ có các Props khác nhau trên Android và IOS. Ví dụ TextInput trên IOS sẽ không thể chỉ định được số dòng (Người dùng vẫn có thể nhập được nhiều dòng, TextInput sẽ tự động dài ra).



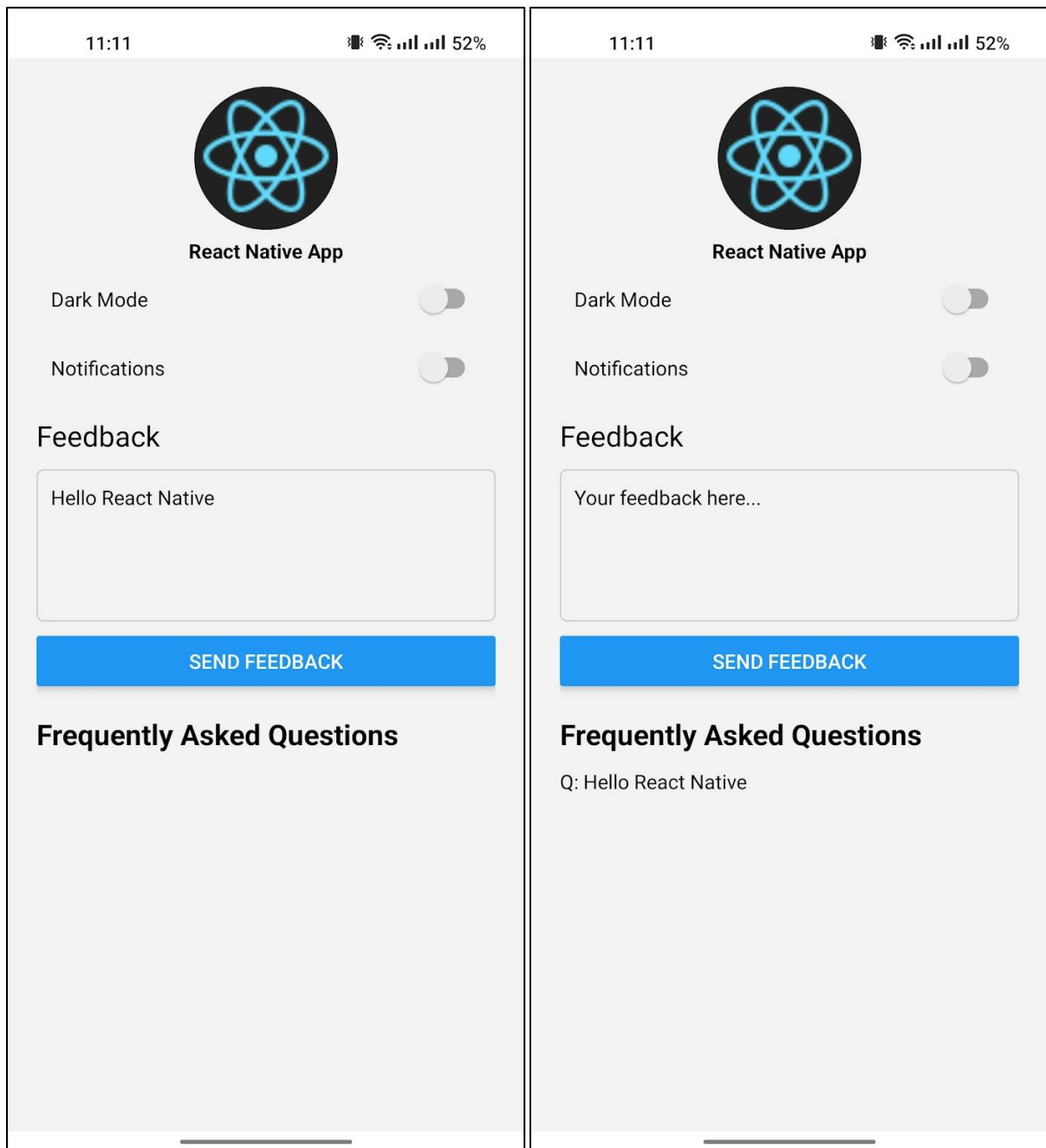
2.5. Hình ảnh ví dụ

- Tắt / Bật chế độ tối



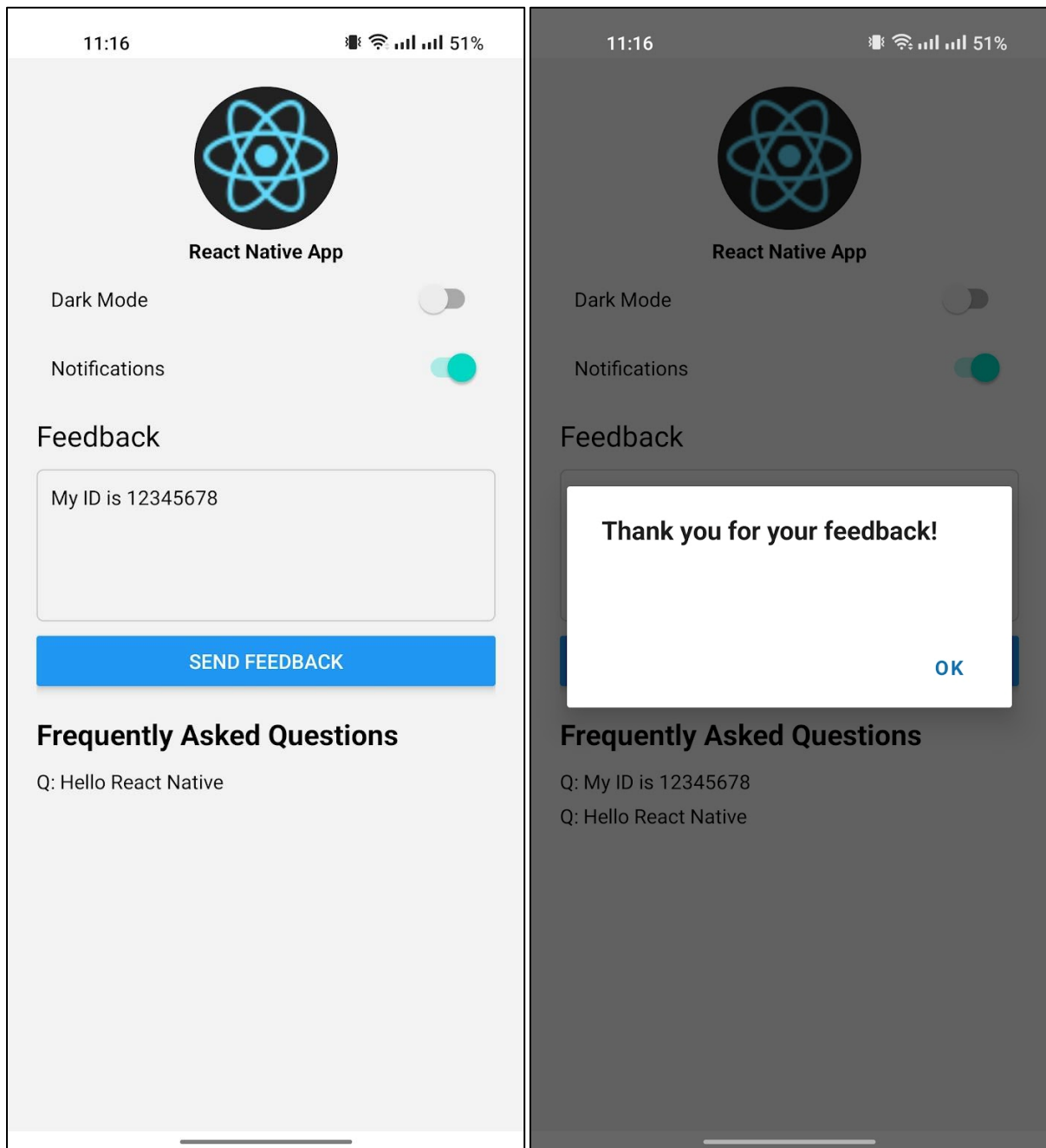


- Người dùng gửi phản hồi với công tắc thông báo đóng



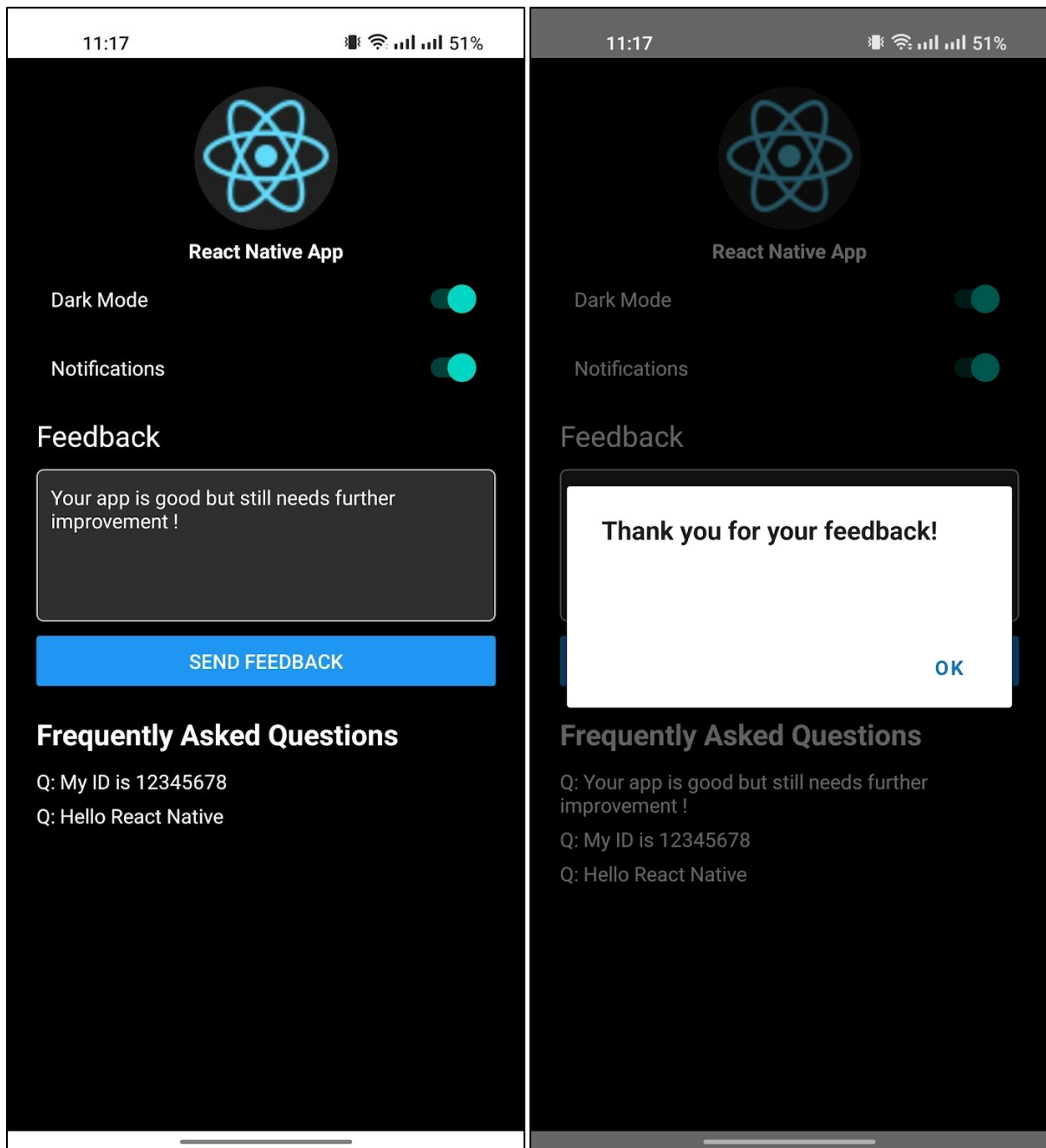


- Người dùng gửi phản hồi với công tắc thông báo bật





- Người dùng gửi phản hồi với công tắc thông báo bật và công tắc chế độ tối bật





- Người dùng không nhập thông tin phản hồi, khi nhấn nút gửi ứng dụng sẽ không thực hiện hành động gì

11:18 51%

React Native App

Dark Mode ☐

Notifications ☐

Feedback

Your feedback here...

SEND FEEDBACK

Frequently Asked Questions

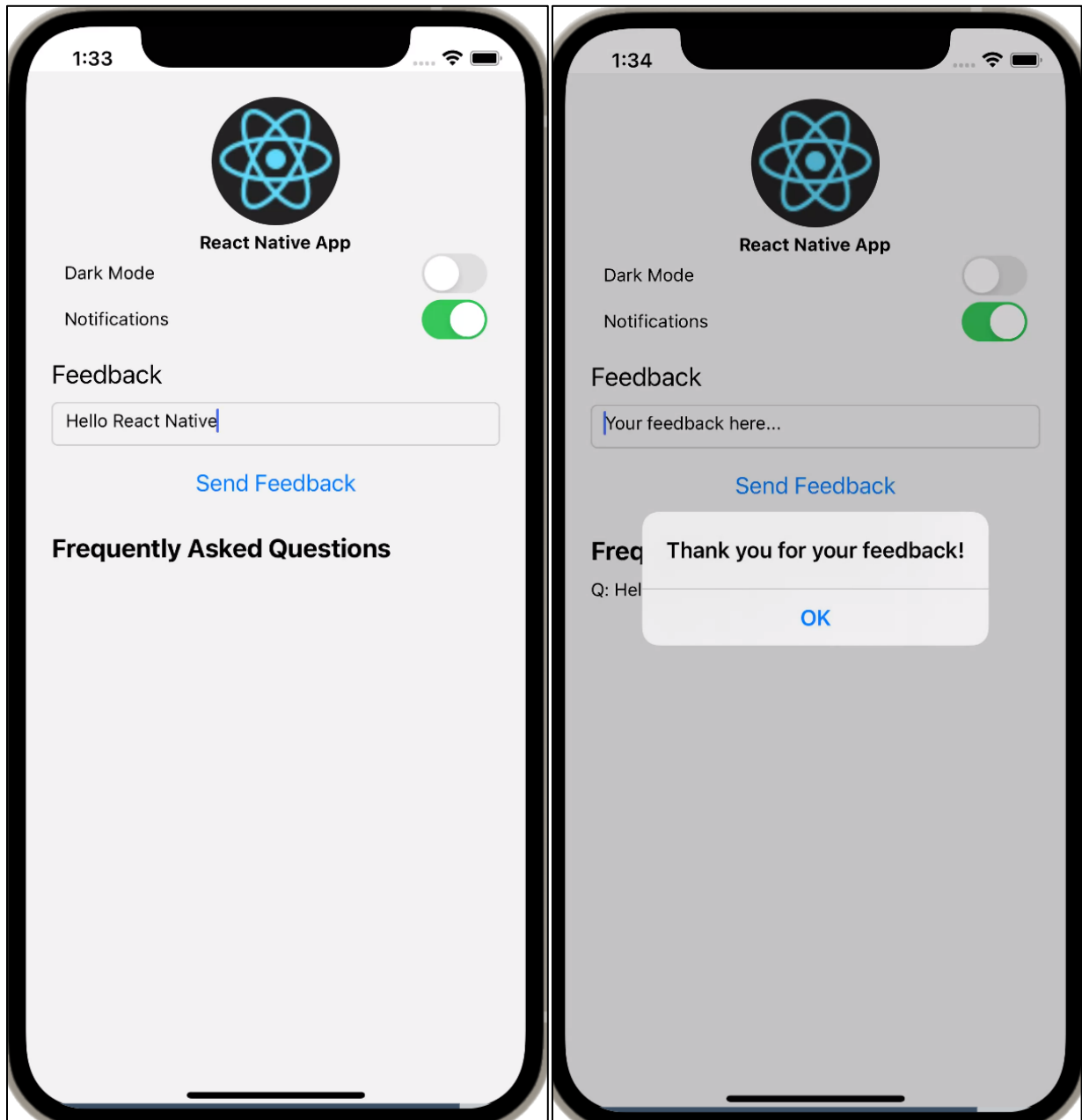
Q: Your app is good but still needs further improvement !

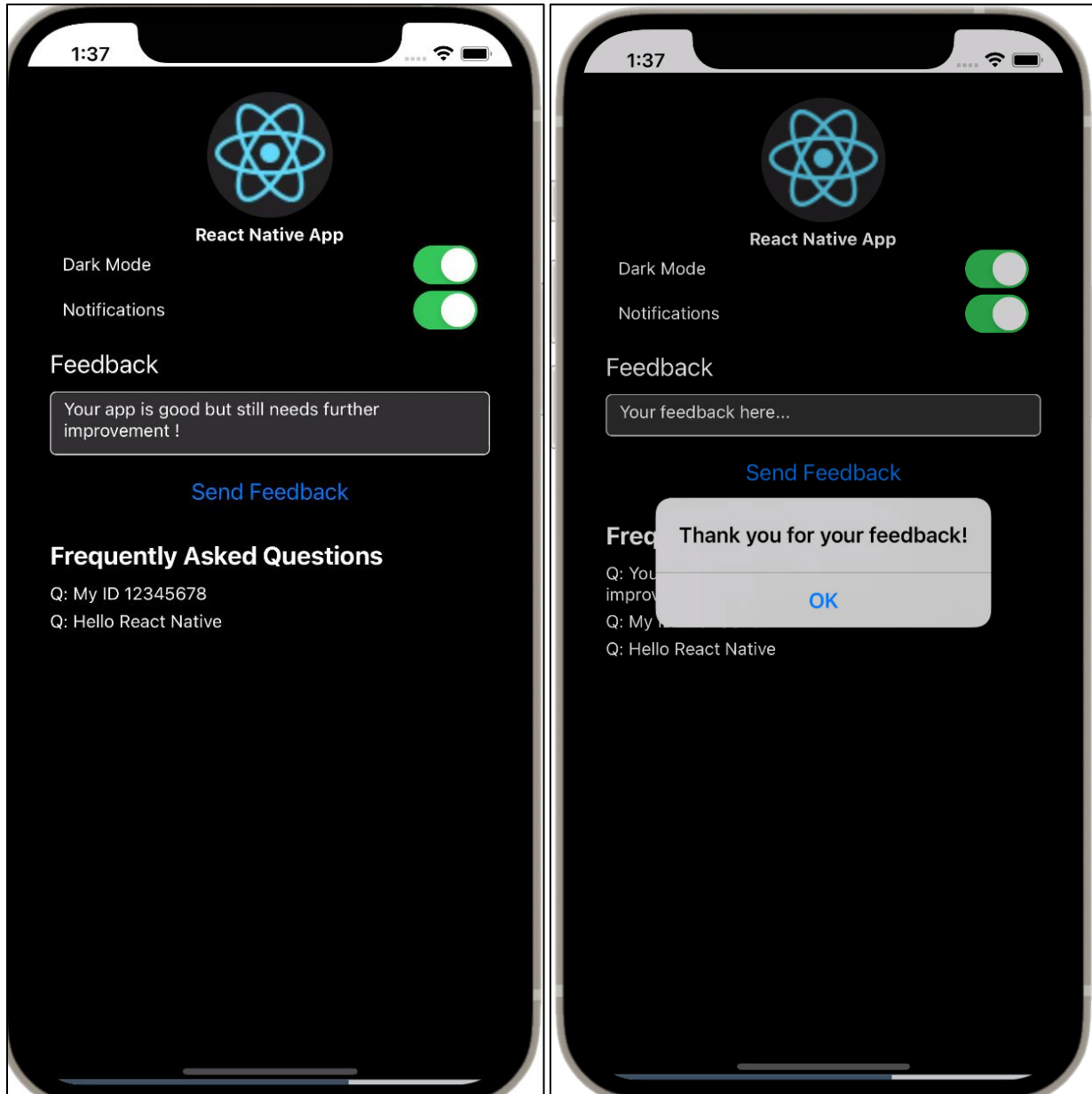
Q: My ID is 12345678

Q: Hello React Native



- Một số hình ảnh trên thiết bị IOS







Bài 3: React Navigation

3.1. Mục tiêu

Biết cách sử dụng các Navigations cơ bản trong React Native để quản lý việc điều hướng giữa các màn hình khác nhau của ứng dụng. Hiểu được luồng xác thực của một ứng dụng di động cơ bản.

3.2. Mô tả

Trong bài tập này, chúng ta sẽ triển khai một luồng điều hướng cơ bản giữa các màn hình trong một ứng dụng di động. Luồng điều hướng sẽ là luồng xác thực cơ bản từ đăng nhập, đăng ký, đến màn hình chính và đăng xuất.

3.3. Yêu cầu

- Khi người dùng mở ứng dụng, ứng dụng sẽ tải và kiểm tra thông tin trạng thái xác thực, người dùng sẽ được điều hướng đến màn hình xác thực hoặc màn hình chính, tùy thuộc vào giá trị của trạng thái xác thực.
- Màn hình xác thực: Bao gồm các màn hình sau:
 - Màn hình đăng nhập: Cho phép người dùng nhập email và mật khẩu. Nếu nhập đúng thông tin, người dùng đăng nhập thành công và được điều hướng đến màn hình trang chủ của ứng dụng. Nếu thông tin nhập không chính xác, hiển thị một thông báo cho người dùng.
 - Màn hình đăng ký: Cung cấp các trường thông tin như tên người dùng, email và mật khẩu để người dùng có thể tạo tài khoản mới. Màn hình này cũng cho phép người dùng điều hướng quay lại màn hình đăng nhập.
- Màn hình chính: Sau khi đăng nhập thành công, người dùng sẽ được chuyển hướng đến màn hình chính, bao gồm các màn hình sau:
 - Màn hình trang chủ
 - Màn hình thể loại
 - Màn hình yêu thích
 - Màn hình tài khoản
- Điều kiện để đăng nhập thành công: email là email sinh viên của bạn và mật khẩu là tên viết liền không dấu của bạn. Ví dụ, email: 12345678@gm.uit.edu.vn – mật khẩu: [nguyenvana](#).



- Các màn hình trang chủ, thể loại, yêu thích, tài khoản chỉ cần hiển thị một nội dung đơn giản thể hiện tên của màn hình ở chính giữa giao diện, không hiển thị tiêu đề trên thanh Header.
- Trong màn hình yêu thích, hiển thị một huy hiệu cho biết số lượng mục trong danh sách yêu thích của người dùng, ví dụ 3 (Ngụ ý rằng người dùng đang có 3 mục/đối tượng trong danh sách yêu thích).
- Trong màn hình tài khoản, sẽ có thêm một nút **LOG OUT** để người dùng đăng xuất khỏi ứng dụng, tức quay trở lại màn hình xác thực.
- Sử dụng React Context để lưu thông tin xác thực của người dùng.
- Gom các components lặp lại / tương tự nhau (Nếu có) thành một custom component để tái sử dụng.
- Tùy chỉnh giao diện của ứng dụng cho bắt mắt và thân thiện với người dùng. Ví dụ, trong các trường nhập thông tin sẽ có thêm icon giúp dễ nhận biết, tự động điều chỉnh khoảng cách, vị trí của các thành phần dựa trên chiều cao của bàn phím, chọn kiểu bàn phím phù hợp với loại dữ liệu cần nhập, thêm các tùy chọn đăng nhập khác như Facebook, Google trong màn hình đăng nhập, v.v.

3.4. Gợi ý


- Sử dụng React Context để quản lý thông tin về email, mật khẩu và trạng thái xác thực khi người dùng nhập để tiến hành xác thực.
- Tạo một StackNavigator chứa các màn hình đăng nhập và đăng ký (ví dụ tên AuthStack).
- Tạo một BottomTabNavigator chứa các màn hình sẽ hiển thị sau khi đăng nhập thành công (ví dụ tên MainBottom).
- Trong NavigationContainer, tạo một Stack chứa AuthStack và MainBottom. Tại đây, kiểm tra giá trị của trạng thái xác thực từ Context đã tạo và thực hiện điều hướng cho phù hợp.
- Tương tự, tại các màn hình đăng nhập và tài khoản, sử dụng thông tin về email, mật khẩu và trạng thái xác thực từ Context đã tạo để thực hiện các logic phù hợp.




3.5. Hình ảnh ví dụ


- Người dùng mở ứng dụng và được chuyển đến màn hình xác thực: Gồm màn hình đăng nhập và đăng ký. Khi nhấn vào nút **Sign up here!** sẽ điều hướng đến màn hình đăng ký và khi nhấn vào nút **Login now!** sẽ điều hướng đến màn hình đăng nhập. Trong bài tập này, chúng ta sẽ không hiện thực chức năng đăng ký.

3:28 45%



Welcome



 Email

 Password

[Forgot password?](#)

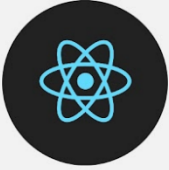
LOG IN

Or login with


 


Don't have an account? [Sign up here!](#)


3:28 45%




Create New Account

 Enter username

 Enter email

 Enter password

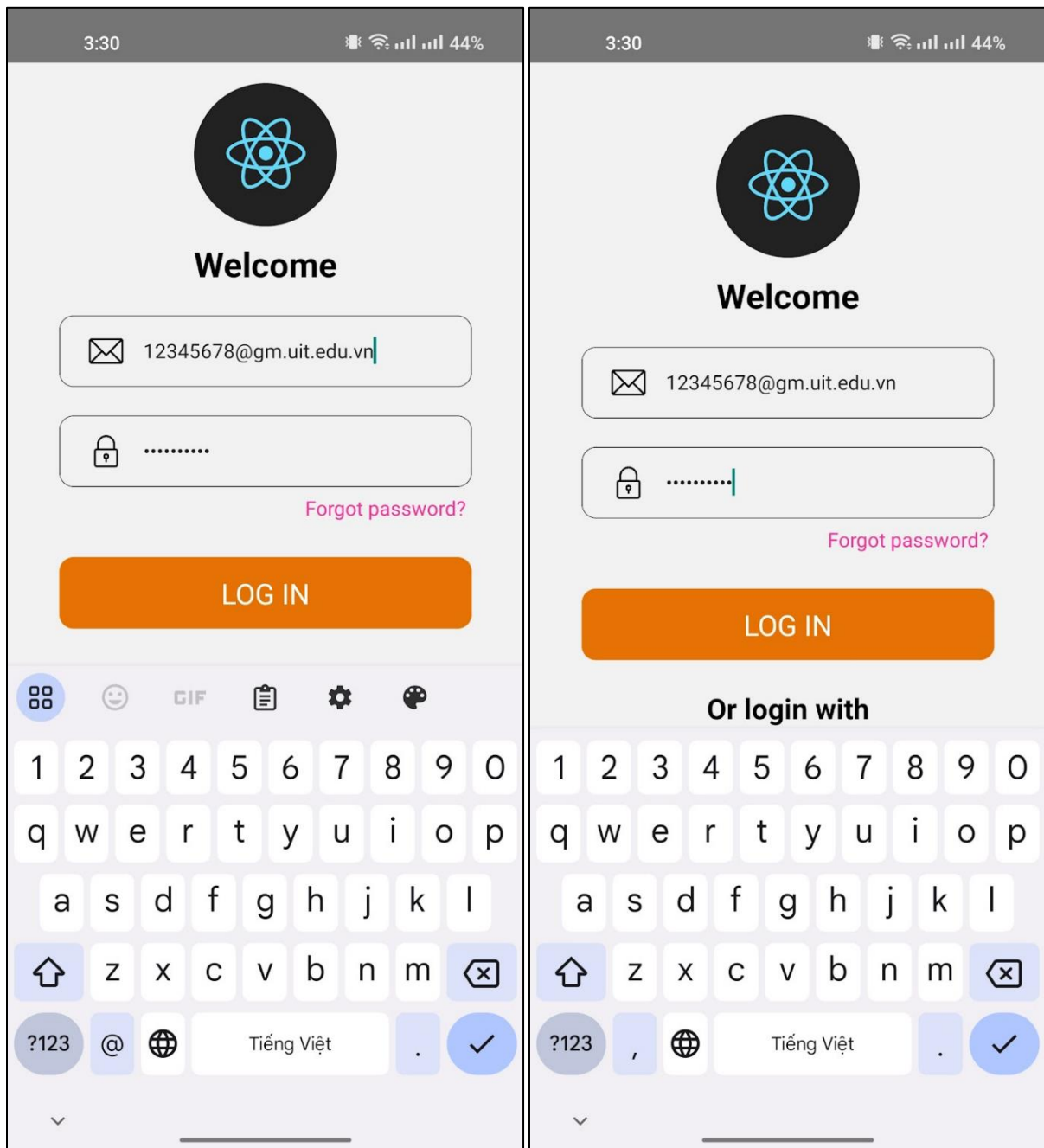
 Confirm password

CREATE

Already have an account? [Login now!](#)

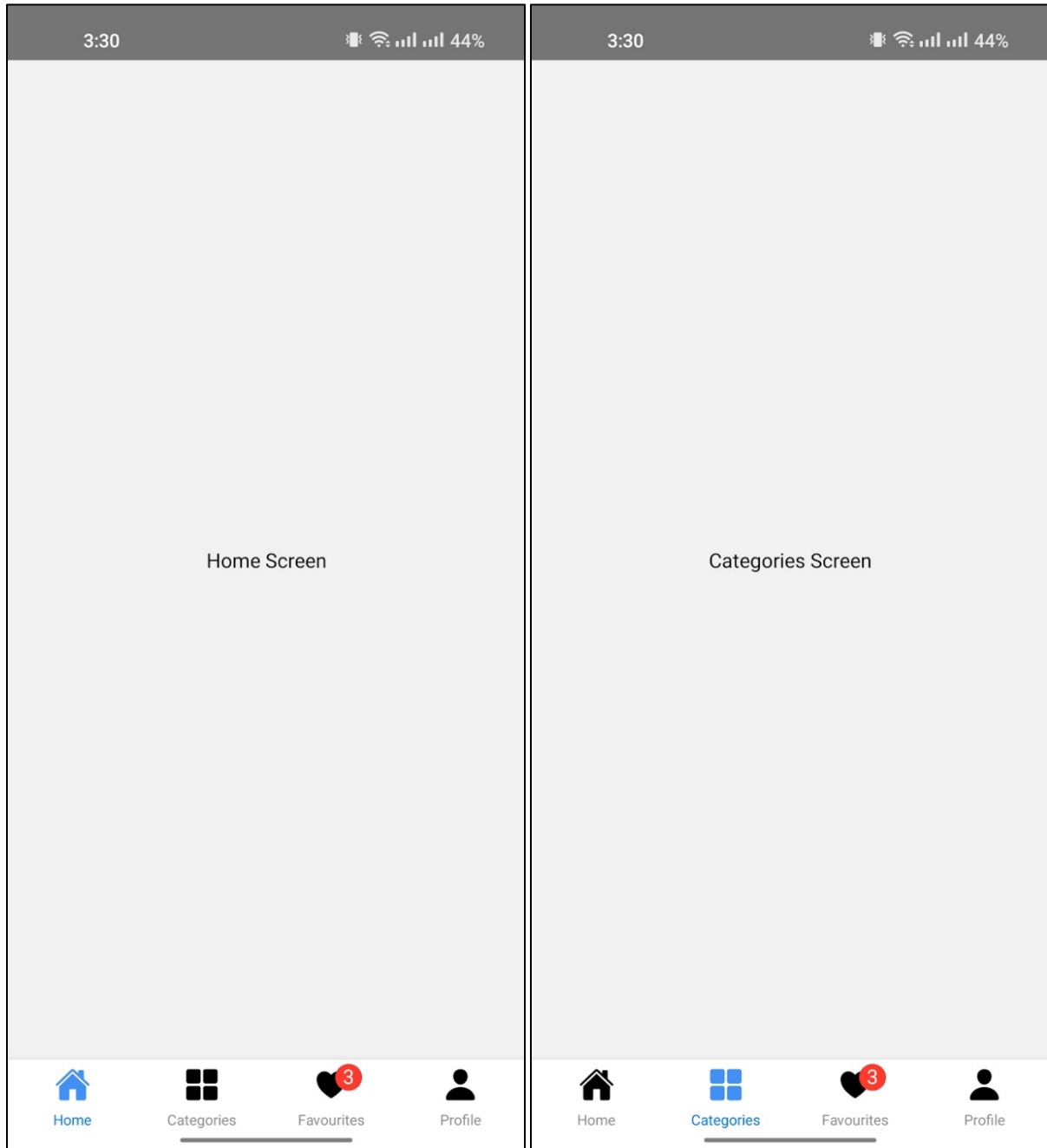


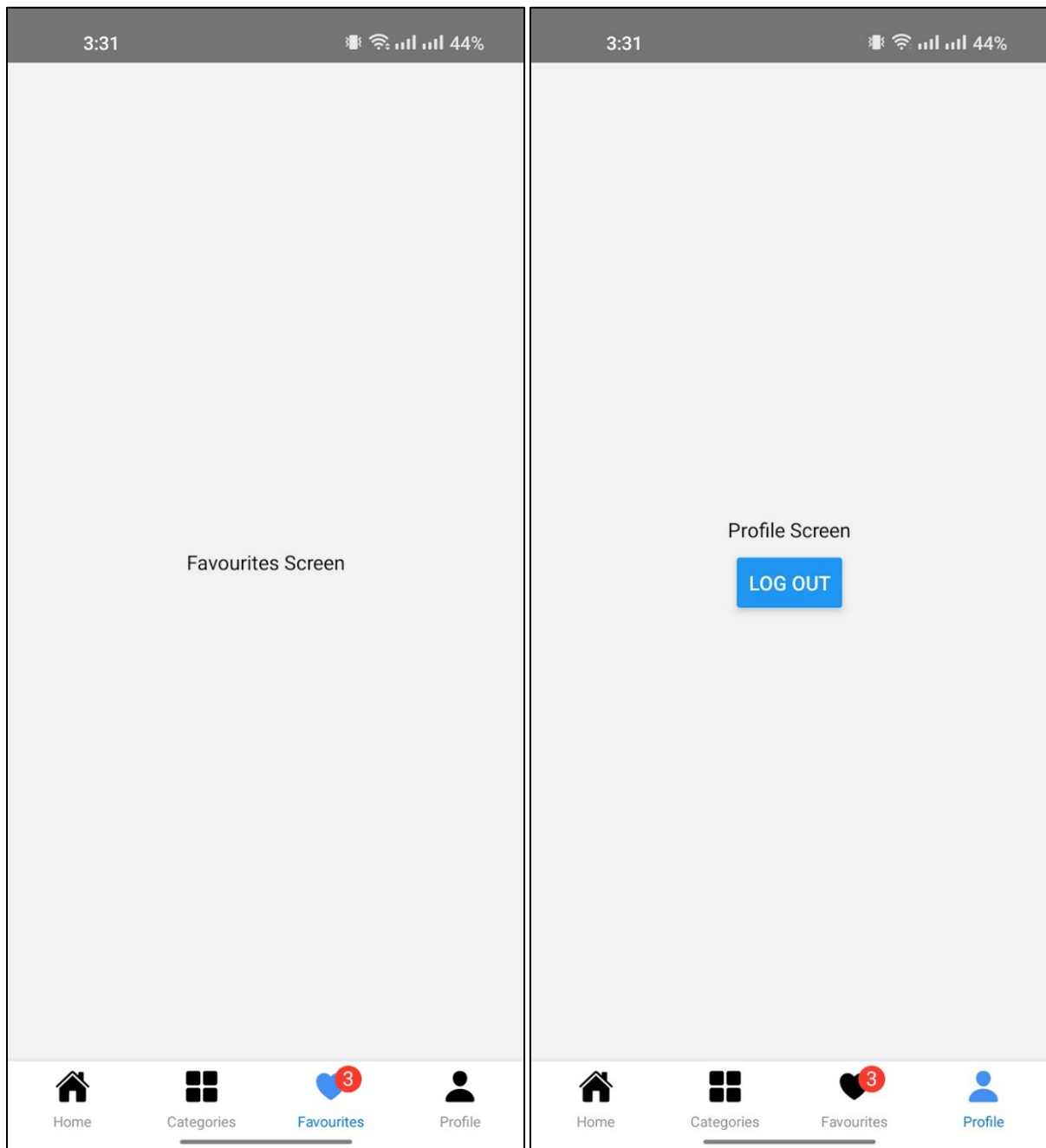
- Người dùng nhập thông tin email và mật khẩu ở màn hình đăng nhập.





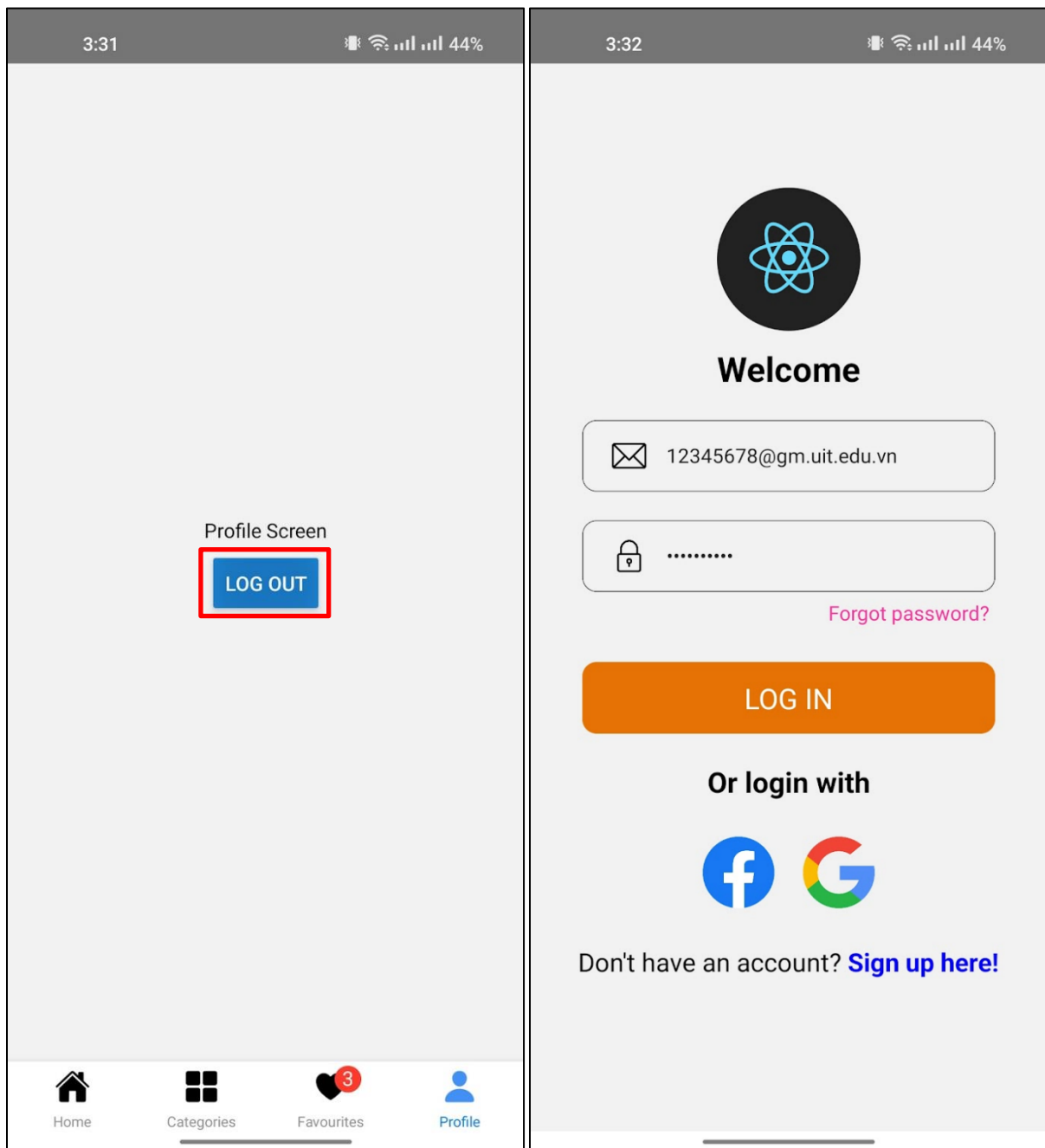
- Màn hình chính của ứng dụng: Bao gồm các màn hình trang chủ, thể loại, yêu thích và tài khoản. Khi người dùng nhập đúng thông tin, ứng dụng sẽ cập nhật thông tin trạng thái xác thực và chuyển đến màn hình chính.





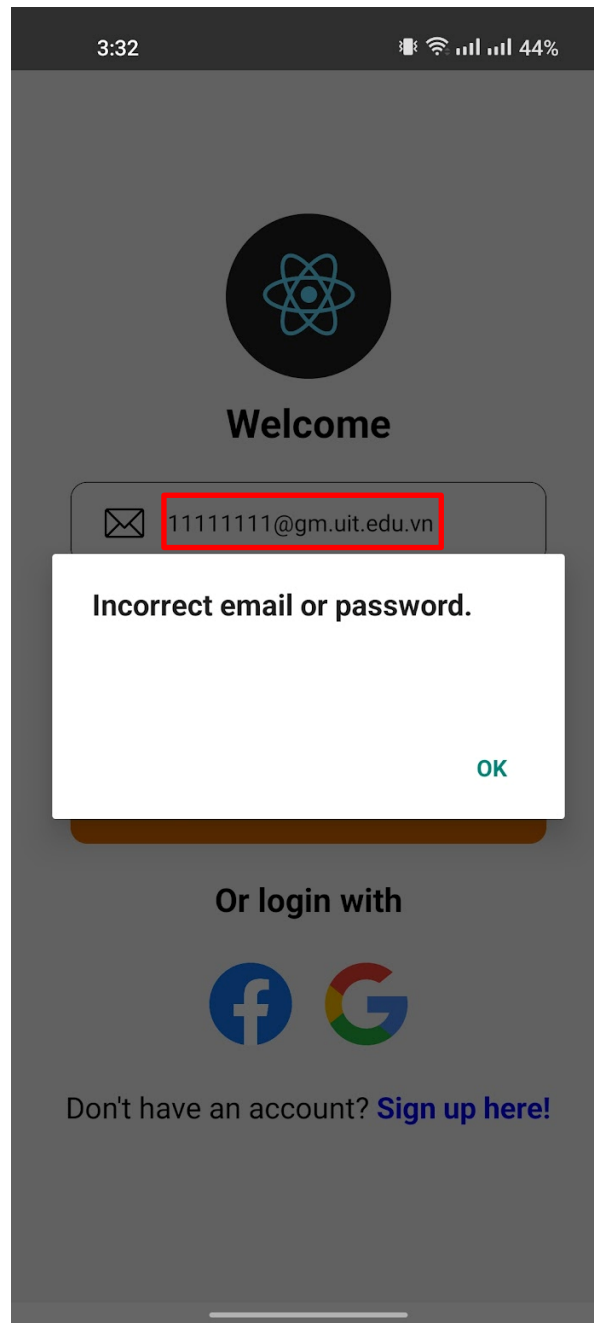


- Trong màn hình tài khoản, khi người dùng nhấn vào nút **LOG OUT**, trạng thái xác thực được cập nhật và ứng dụng sẽ đưa người dùng quay lại màn hình xác thực.





- Thông báo lỗi khi người dùng nhập sai thông tin ở màn hình đăng nhập.





- Một số hình ảnh trên thiết bị IOS
- Màn hình đăng nhập và đăng ký.

The image displays two side-by-side mobile app screens, both featuring a status bar at the top with the time 14:20, a camera icon, and signal/battery indicators. Both screens have a dark blue circular logo with a white atom-like symbol at the top center.

Left Screen (Welcome/Login):

- Title: **Welcome**
- Form fields: Email (with envelope icon), Password (with lock icon).
- Link: [Forgot password?](#)
- Button: **LOG IN** (orange)
- Text: **Or login with**
- Icons: Facebook (f) and Google (G)
- Text: Don't have an account? [Sign up here!](#)


Right Screen (Create New Account):

- Title: **Create New Account**
- Form fields: Enter username (with person icon), Enter email (with envelope icon), Enter password (with lock icon), Confirm password (with lock icon).
- Button: **CREATE** (orange)
- Text: Already have an account? [Login now!](#)




- 14:22


Camera



Welcome




12345678@gm.uit.edu.vn




Forgot password?

LOG IN

Or login with

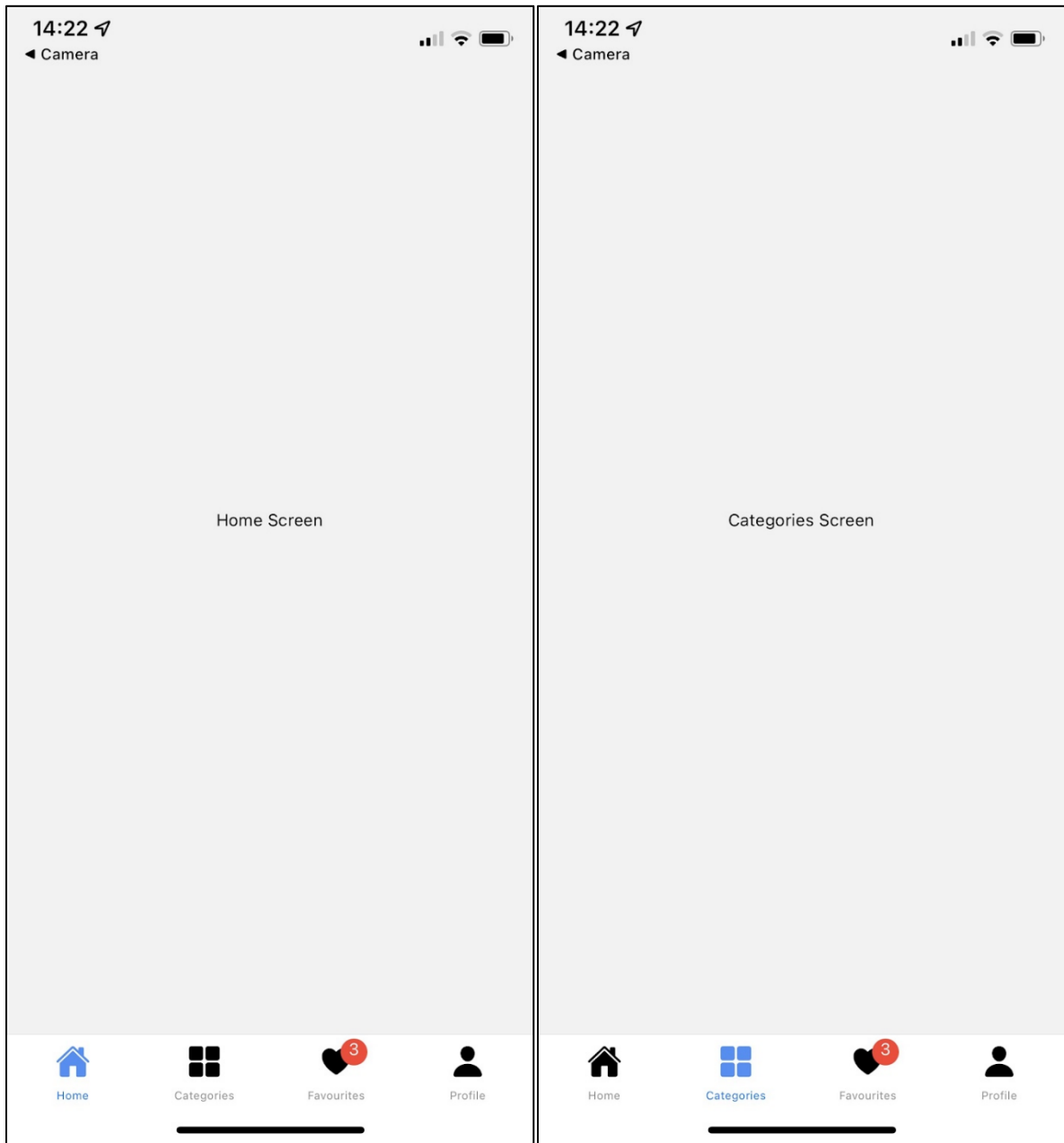


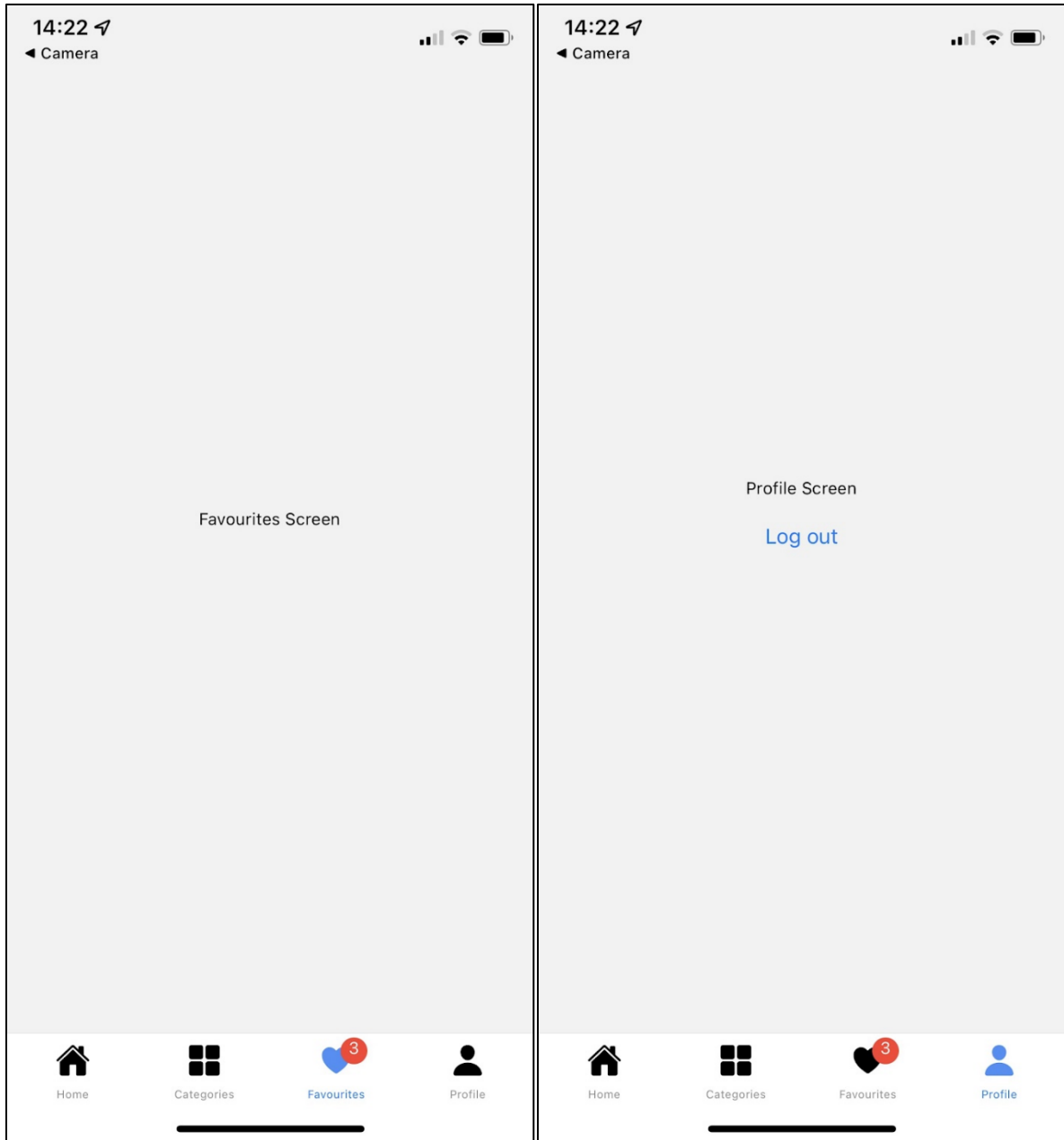


Don't have an account? [Sign up here!](#)



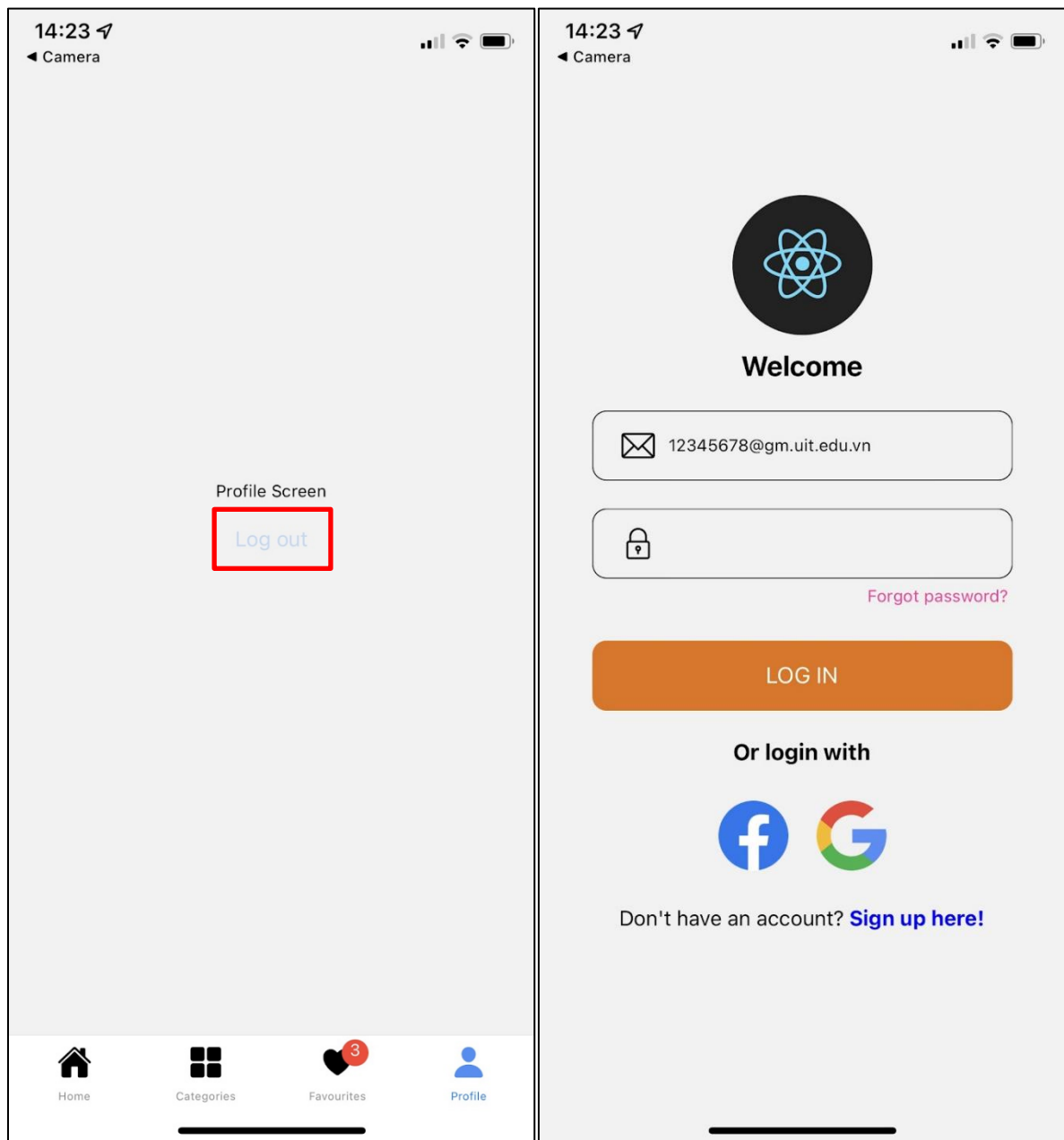
- Màn hình chính của ứng dụng.





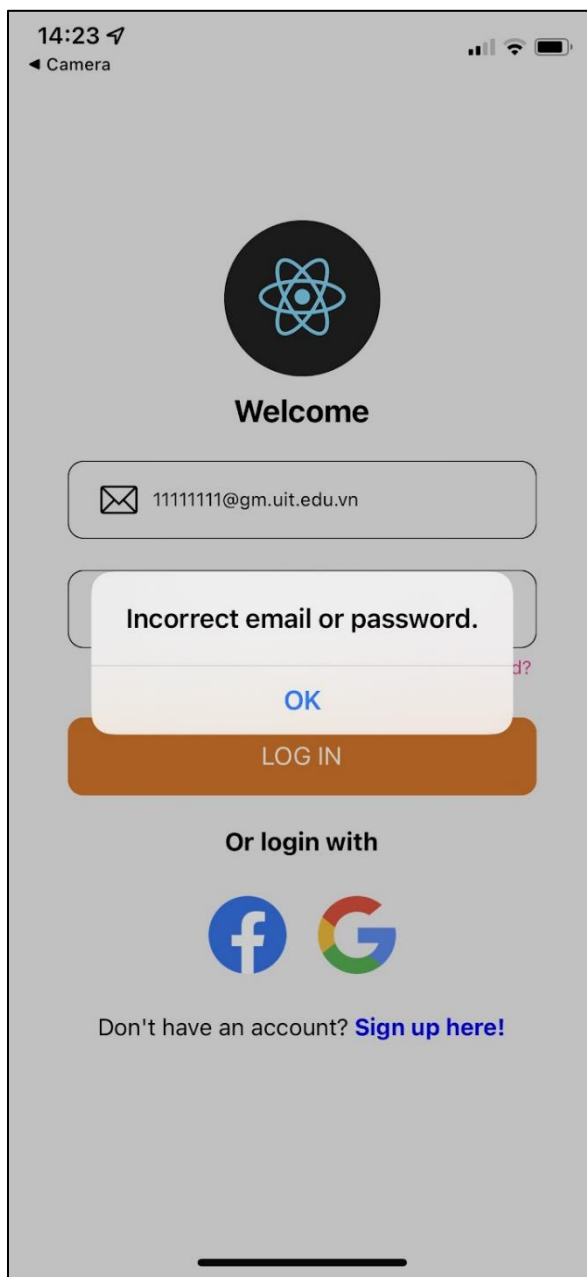


- Khi người dùng nhấn vào nút **LOG OUT**, trạng thái xác thực được cập nhật và ứng dụng sẽ đưa người dùng quay lại màn hình xác thực.





- Thông báo lỗi khi người dùng nhập sai thông tin ở màn hình đăng nhập.





HƯỚNG DẪN NỘP BÀI

- File nộp gồm 2 phần: PDF và ZIP.

- **PDF:**

- Chụp màn hình các mục sau đây và chuyển sang file pdf, đặt tên: **MSSV_BTTHX.pdf**.
 - Cấu trúc các folders / files của dự án.
 - Toàn bộ nội dung code của các file liên quan (tức phần thực hiện) và Màn hình kết quả của các câu. Trong đó, **màn hình code** phải thể hiện thông tin của cá nhân gồm **Tên và MSSV**; **tiêu đề (title) của CMD/Terminal là MSSV**.
- Quay video màn hình kết quả khi thực hiện các yêu cầu của bài tập (quay trên máy thật hoặc máy ảo đều được), upload lên google drive và **chèn link vào file pdf** ở trên (chia sẻ quyền truy cập cho mail duypn@uit.edu.vn).

- **ZIP:**

- Nén source code lại thành file zip, đặt tên: **MSSV_BTTHX.zip**
 - Bỏ chọn thư mục **node_modules** trước khi nén. Trường hợp sử dụng React Native CLI cần xóa thêm thư mục **build** tương ứng với nền tảng trước khi nén. Đường dẫn của thư mục build sẽ là: **Tên_Project/android/app/build** nếu chạy trên Android, **Tên_Project/ios/build** nếu chạy trên IOS.

- **Lưu ý:**

- **X** là số thứ tự của bài thực hành.
- File PDF không nằm trong file nén ZIP.
- Màn hình kết quả là:
 - Màn hình được chụp trên thiết bị thật của cá nhân.
 - Hoặc, Màn hình được chụp trên Android Virtual Device (Android Studio) đối với môi trường Android.
 - Hoặc, Màn hình được chụp trên Xcode đối với môi trường IOS.