

**VIETNAM NATIONAL UNIVERSITY - HO CHI MINH
UNIVERSITY OF SCIENCE**

Coursera - React Native of Meta

Module 4

Academic Supervisors:

Student:

23127438 - Dang Truong Nguyen

Bui Duy Dang

Huynh Lam Hai Dang

Tran Trung Kien

Semester: I

Academic Year: 2025 - 2026

Ho Chi Minh City, November 13, 2025

TABLE OF CONTENTS

TABLE OF CONTENTS	1
LIST OF FIGURES	4
1 React Navigation	6
1.1 Video: What is React Navigation	6
1.1.1 Problem	6
1.1.2 React Native Navigation	6
1.1.3 Benefit of using React Native Navigation	6
1.1.4 Trade off	7
1.1.5 Other commonly used library	7
1.2 Reading: Installation and Setup of React Navigation	7
1.3 Video: Setting up React Navigation	8
1.4 Video: Using the Stack.Navigator	8
1.5 Reading: Creating and Configuring Native Stack Navigator	8
1.6 Video: Approaches to Passing Props to Screen	9
1.7 Reading: Passing props to Screen	9
1.8 Reading: Exercise: Set up Routes	10
1.9 Practice Assignment: Self review: Set up Routes	10
1.10 Practice Assignment: Knowledge Check: Introduction to React Navigation	11
2 Screen	12
2.1 Video: Moving to a New Screen	12
2.1.1 Recall the browser navigation	12

2.1.2	navigate	12
2.1.3	push	13
2.1.4	goBack	13
2.1.5	goToTop	13
2.2	Video: Moving Between Screen	13
2.3	Reading: Navigating to a Screen	14
2.4	Video: Going Back to Previous Screen	15
2.5	Video: Configuring a Header Bar	16
2.6	Exercise: Moving between screens	17
2.7	Practice Assignment: Self review: Moving between screens	17
2.8	Practice Assignment: Knowledge Check: Moving between screens	18
3	Tab Navigation	19
3.1	Video: What is Tab Navigation	19
3.1.1	Stack Navigation vs Tab Navigation	19
3.1.2	Options	19
3.2	Video: Tab Navigation Example	20
3.3	Video: Customizing your tab navigator	21
3.4	Reading: Exploring Tab Navigation	22
3.5	Reading: Exercise: Configure Tab Navigation	22
3.6	Practice Assignment: Self review: Configure Tab Navigation	23
3.7	Practice Assignment: Knowledge Check: Tab Navigation	24
4	Drawer Navigation	25
4.1	Video: What is Drawer Navigation	25
4.1.1	Drawer Navigation	25
4.1.2	Syntax	25
4.2	Video: Drawer Navigation Example	26

4.3	Reading: Exploring Drawer Navigation	27
4.4	Reading: Exercise: Configure Drawer Navigation	28
4.5	Practice Assignment: Self review: Configure Drawer Navigation	28
4.6	Practice Assignment: Knowledge check: Drawer Navigation	29
4.7	Video: Module Summary: React Navigation	29
4.8	Graded Assignment: Module Quiz: React Navigation	30

LIST OF FIGURES

Figure 1.1	Using the Stack.Navigator	8
Figure 1.2	Passing prop into Stack	9
Figure 1.3	Exercise: Set up Routes	10
Figure 1.4	Self review: Set up Routes	10
Figure 1.5	Knowledge Check: Introduction to React Navigation	11
Figure 2.1	Welcome Screen before click to move to Login Screen	14
Figure 2.2	After click to move to Login Screen	14
Figure 2.3	Login Screen with the go back button	15
Figure 2.4	Welcome Screen with customize header bar	16
Figure 2.5	The Login Screen's header bar is not affected	17
Figure 2.6	Self review: Moving between screens	17
Figure 2.7	Knowledge Check: Moving between screens	18
Figure 3.1	Welcome Screen with Tab Navigation	20
Figure 3.2	Login Screen with Tab Navigation	20
Figure 3.3	Welcome Screen with customize Tab Navigation	21
Figure 3.4	Login Screen with customize Tab Navigation	21
Figure 3.5	Exercise: Configure Tab Navigation - Welcome Screen	22
Figure 3.6	Exercise: Configure Tab Navigation - Login Screen	23
Figure 3.7	Self review: Configure Tab Navigation	23
Figure 3.8	Knowledge Check: Tab Navigation	24
Figure 4.1	Drawer with Home and Login Screen	26

Figure 4.2	Drawer with Home and Login Screen	27
Figure 4.3	Drawer with Home and Login Screens	28
Figure 4.4	Self review: Configure Drawer Navigation	28
Figure 4.5	Knowledge check: Drawer Navigation	29
Figure 4.6	Module Quiz: React Navigation	30

CHAPTER 1

React Navigation

1.1. Video: What is React Navigation

1.1.1. Problem

- Moving between screens is referred to as navigation or routing in mobile apps.
- Mobile apps have multiple screens and the user's experience relies on smooth navigation between these screens.
- React Native does not have inbuilt navigation, it must pick a navigation library to implement moving between the screens and the app. → React Navigation comes onto the screen

1.1.2. React Native Navigation

- React Navigation is the most popular navigation library for React Native apps today.
- It is easy to use, customizable based on your needs, and has platform-specific components for both iOS and Android.
- Can be used for both React Native CLI apps and React Native built with Expo.
- It provides a Native Stack Navigator that can transition between screens and manage the navigation history.

1.1.3. Benefit of using React Native Navigation

- The difference between routing and navigation works in web browsers versus React Navigation for mobile app: **React Navigation provides the gestures and animations.**
- React Native provides Screens, which provide native navigation container components

- Its also provides gesture handler, which make React Native more compatible with native touch and gesture system.
- Which brings:
 - Easier to push OTA updates.
 - Easier to debug.
 - Possible to customize navigator components.

1.1.4. Trade off

- Some of the navigators don't directly use the native navigation APIs on iOS and Android.
- While gaining access to new features after upgrading is generally good news, it's possible that code using older APIs will no longer work as expected.

1.1.5. Other commonly used library

- React-native-router-flux
- React-native-navigation

Practice Question: React Navigation is the most popular navigation library for React Native apps. Which of the following are characteristics of this library? Select all that apply. → Answer:

- iOS-specific components.
- Customizable navigators
- Android-specific components.

1.2. Reading: Installation and Setup of React Navigation

```
npm install @react-navigation/native  
  
npx expo install react-native-screens  
      react-native-safe-area-context
```

1.3. Video: Setting up React Navigation

Documentation: <https://reactnavigation.org/> **Practice Question:** What are the base packages you would need to install to setup React Navigation within your project? Select all that apply. → Answer:

- @react-navigation/native
- react-native-safe-area-context
- react-native-screens

1.4. Video: Using the Stack.Navigator

Practice Question: The name given to a screen within a Stack.Navigator must match the name of the component that it calls. True or false? → Answer: False

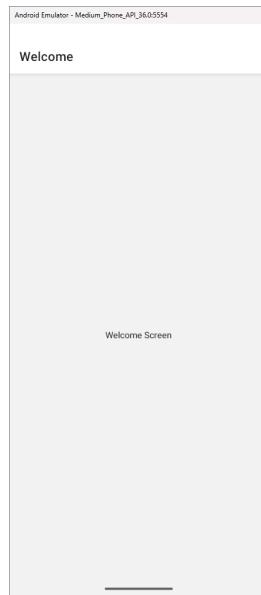


Figure 1.1: Using the Stack.Navigator

1.5. Reading: Creating and Configuring Native Stack Navigator

This part is the same progress of using Stack.Navigator that I have done in the above section.

1.6. Video: Approaches to Passing Props to Screen

- To change the title of the Stack Screen, use in Stack.Screen:

```
options={{ title: '...' }}
```

- To set the root Stack that would be viewed when user open the first time, use in Stack.Navigator:

```
initialRouteName='...'
```

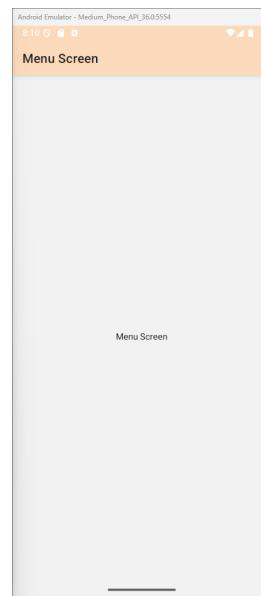


Figure 1.2: Passing prop into Stack

1.7. Reading: Passing props to Screen

This part is the same progress that I have done in the above section.

1.8. Reading: Exercise: Set up Routes

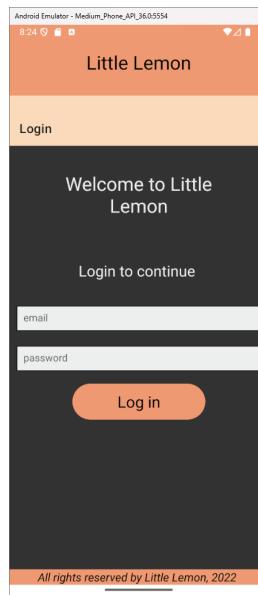


Figure 1.3: Exercise: Set up Routes

1.9. Practice Assignment: Self review: Set up Routes

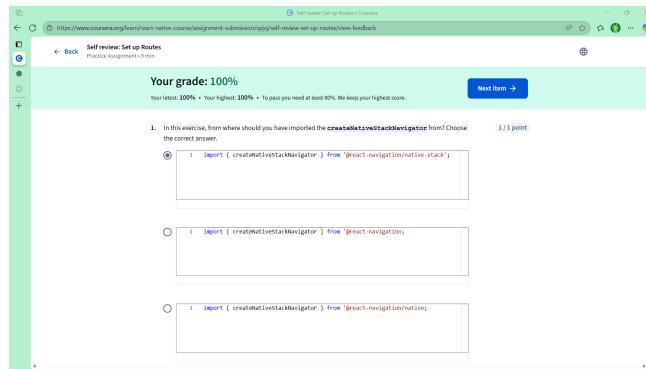


Figure 1.4: Self review: Set up Routes

1.10. Practice Assignment: Knowledge Check: Introduction to React Navigation

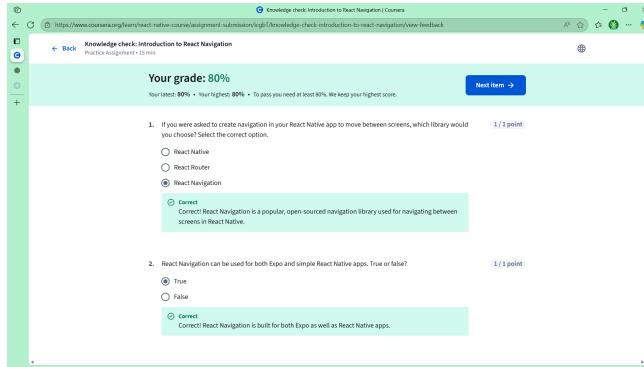


Figure 1.5: Knowledge Check: Introduction to React Navigation

CHAPTER 2

Screen

2.1. Video: Moving to a New Screen

2.1.1. Recall the browser navigation

```
<a href='detail.html'>Go to detail</a>
```

- In React Native, we using navigation props.
- Several function:
 - navigate
 - push
 - goBack
 - popToTop

2.1.2. navigate

- Designates a route for the user to use.
- Example:

```
<Button  
      title='Go to detail'  
      onPress={() => navigation.navigate('details')}>  
</Button>
```

- If we call the navigate function without defining the root:
 - In development build, it will return an error.
 - In production build, it make no effect.

2.1.3. push

- Adds new route to navigate stack.
- Difference between navigate and push:
 - **navigate:** When calling a navigate, it will first try to find an existing route with that name and only pushes a new route if there isn't one on the stack yet.
 - **push:** add another route on the top of the stack regardless of the navigation stack history.

2.1.4. goBack

- Returns to previous screen in navigation history.
- If user go from screen 1 to screen 2 to screen 3 to screen 4. At screen 4 there is a button that call a goBack function, user will go back to screen 3

2.1.5. goToTop

- Retuns to the root screen in navigation history.
- In the above example, if user click the go back button that call goToTop, user will be at the screen 1.

Practice Question: In setting up navigation in your app, you'd like to add a new route to the top of the stack, regardless of what the user's navigation history is. Which method would you call from the navigation prop to do this? → push

2.2. Video: Moving Between Screen

Practice Question: You'd like to set up a Pressable with the onPress method, so that it goes to a screen called "Contact" when the user clicks on it. Which of the following lines of code would you type in the Pressable to do this? → navigation.navigate('contact')

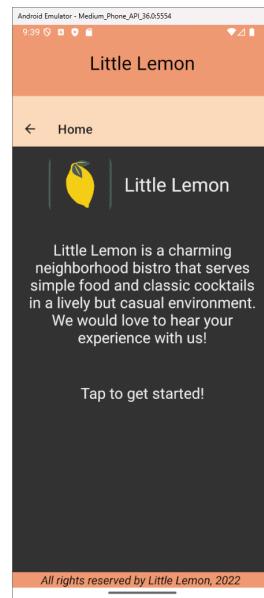


Figure 2.1: Welcome Screen before click to move to Login Screen

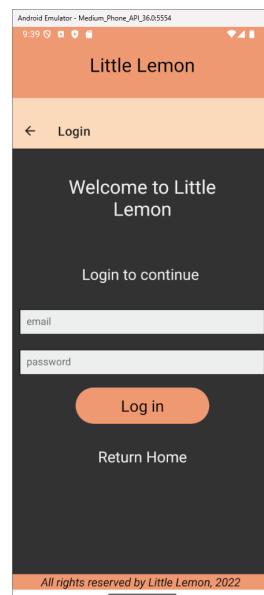


Figure 2.2: After click to move to Login Screen

2.3. Reading: Navigating to a Screen

This part is the same progress that I have made in the previous section to move from one screen to another screen.

2.4. Video: Going Back to Previous Screen

- In addition to back button on the Stack navigator, we can program a button that allow user to go back by out self using goBack props.

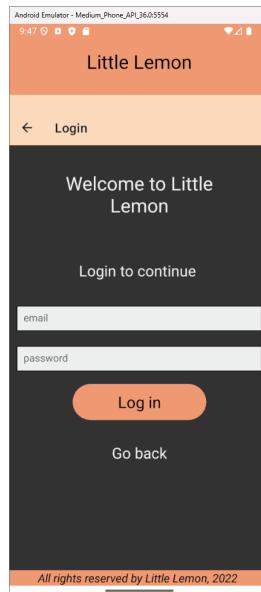


Figure 2.3: Login Screen with the go back button

Practice Question: You need to add a Pressable component on your screen that calls the goBack function, otherwise users have no way of returning to the previous screen in the stack. True or false? → False

- The android have 3 ways to go back:
 - The back button on stack navigator
 - The goBack props
 - The back button on phone navigator.
- The transition of moving between screens is different from iOS and Android.

2.5. Video: Configuring a Header Bar

Practice Question: In your app, you would like to style the header bar, but only for the screen titled “About Us”. To do this, you would apply the header styling options inside of Stack.Screen for “About Us”. True or false? → True

- To display a different header title for a specific header bar, use the headerTitle props inside the option of Stack.Screen like this:

```
<Stack.Screen
    options={{ 
        title: 'Home',
        headerTitle: () => <LogoTitle />
    }}
    name='Home'
    component={WelcomeScreen}
/>
```

- The LogoTitle is customize components.

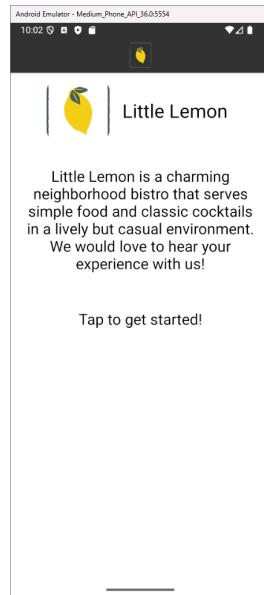


Figure 2.4: Welcome Screen with customize header bar

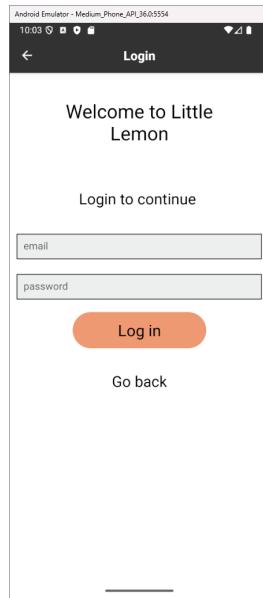


Figure 2.5: The Login Screen's header bar is not affected

2.6. Exercise: Moving between screens

This part is to navigating from Welcome Screen and Login Screen that I have made in the above section.

2.7. Practice Assignment: Self review: Moving between screens

The screenshot shows a browser window with the URL <https://www.coursera.org/learn/react-native-course/assignment-submissions/1952/self-review-moving-between-screens/view/feedback>. The page title is "Self review: Moving between screens".
Question 1: "In this exercise, which prop was passed to the `LoginScreen` to navigate between screens?"
Options:

- `nativere-navigation`
- `navigation`
- `navigate`

Feedback: "Correct! The `navigation` prop is accessible by any screen within the app that needs it. The `LoginScreen` was passed the `navigation` prop." (1 / 1 point)
Question 2: "In this exercise, which method was called, along with the route's name, to move the user to the `Welcome` screen?"
Options:

- `nativere-navigation("Welcome")`
- `navigate("Welcome")`
- `navigation("Welcome")`

Feedback: "Correct! `navigate` is the function on the `navigation` prop that is passed the name of the route you'd like the user to move to." (1 / 1 point)

Figure 2.6: Self review: Moving between screens

2.8. Practice Assignment: Knowledge Check: Moving between screens

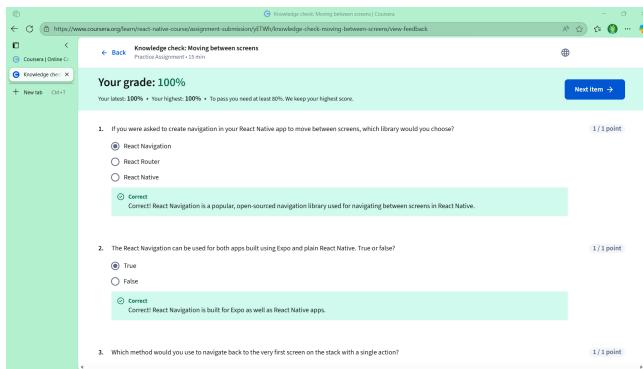


Figure 2.7: Knowledge Check: Moving between screens

CHAPTER 3

Tab Navigation

3.1. Video: What is Tab Navigation

- Its a React Navigation library for setting up tab-based navigation between screens.

3.1.1. Stack Navigation vs Tab Navigation

- Stack navigation can transition between screens and navigation history.
- With Tab navigation, you can contain tabs on the bottom of the screen or on the header.
Clicking on a tab takes you a specific screen. → common design on modern mobile app.

3.1.2. Options

- Bottom tab navigation
- Material design-based bottom tab navigation
- Material design-based top tab navigation
- Within the tab navigator screen options such as icons, active and inactive tint colors are readily available.

Practice Question: You would like to create a tab navigator in your React Native app. Which of the following are possible options? Select all that apply.

- Material-design bottom tab navigation
- Material-design top tab navigation
- Bottom tab navigation

3.2. Video: Tab Navigation Example

Practice Question: You would like to add a Bottom Tab Navigator to your React Native. This is included with the NavigationContainer , so once you've imported that then you are ready to go. True or false? → False

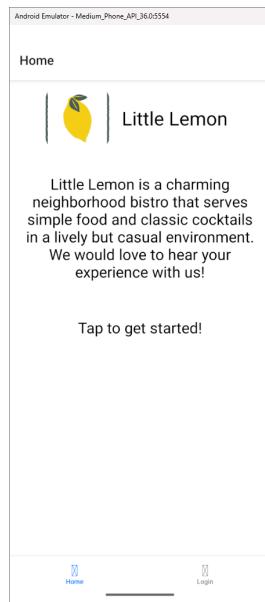


Figure 3.1: Welcome Screen with Tab Navigation



Figure 3.2: Login Screen with Tab Navigation

3.3. Video: Customizing your tab navigator

Practice Question: You would like to customize the tab icons in your Tab Navigator. However, only one icon can be assigned to a tab. True or false? → False

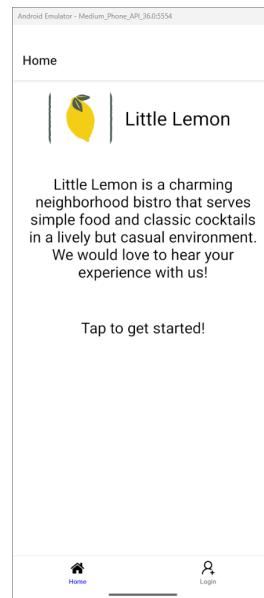


Figure 3.3: Welcome Screen with customize Tab Navigation

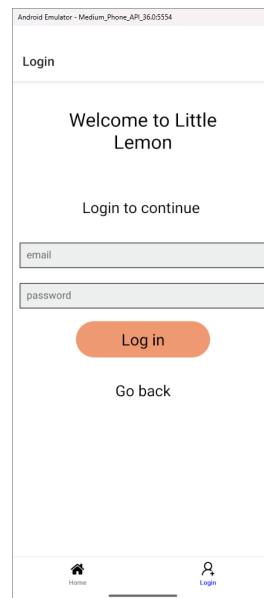


Figure 3.4: Login Screen with customize Tab Navigation

3.4. Reading: Exploring Tab Navigation

This part is the same progress that I have made in the above section, from configuring Tab Navigation to customizing it.

3.5. Reading: Exercise: Configure Tab Navigation

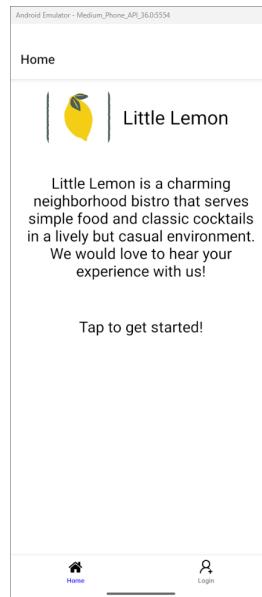


Figure 3.5: Exercise: Configure Tab Navigation - Welcome Screen

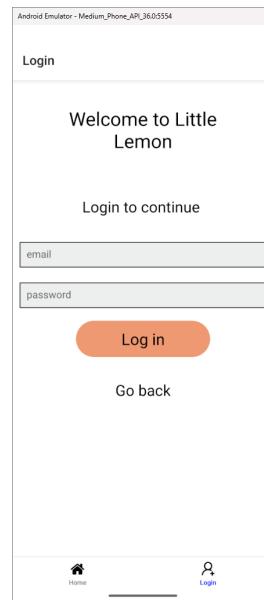


Figure 3.6: Exercise: Configure Tab Navigation - Login Screen

3.6. Practice Assignment: Self review: Configure Tab Navigation

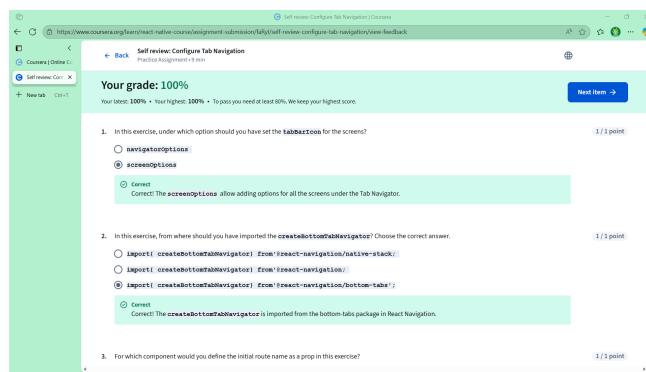


Figure 3.7: Self review: Configure Tab Navigation

3.7. Practice Assignment: Knowledge Check: Tab Navigation

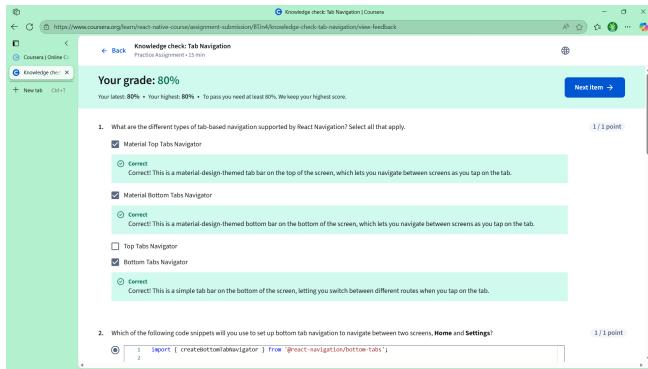


Figure 3.8: Knowledge Check: Tab Navigation

CHAPTER 4

Drawer Navigation

4.1. Video: What is Drawer Navigation

4.1.1. Drawer Navigation

- The drawer appears when the user touches the hamburger icon in the app bar or when the user swipes a finger from the edge of the screen.
- Drawer navigation is, like Tab navigation, a common navigation pattern. It uses a drawer from left to right side for navigating between screens.
- You would use this pattern typically when there are many screens to navigate to.

4.1.2. Syntax

Import

```
import { createDrawerNavigator } from
'@react-navigation/drawer'

const Drawer = createDrawerNavigator();
```

Use in App

```
<Drawer.Navigator>
  <Drawer.Screen name='...' component={...} />
</Drawer.Navigator>
```

- We can add option to open the drawer on the right of the screen.
- We can also add multiple choices for the drawer, for example the navigation in the left drawer, and the contact in the right drawer,...

Practice Question: Which of the following statements are correct with regards to drawer navigation? Select all that apply.

- You can either swipe to open the drawer or simply click on the hamburger menu on both Android and iOS platforms.
- You can customize your app to support multiple drawers; one on the right and one on the left.
- By default you swipe to your right, and the drawer opens on the left.
- You can also customize the drawer position so that the drawer opens on the right with a left swipe.

4.2. Video: Drawer Navigation Example

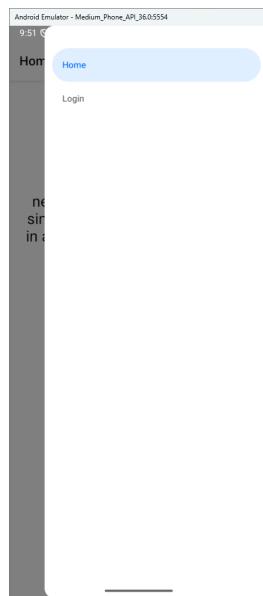


Figure 4.1: Drawer with Home and Login Screen



Figure 4.2: Drawer with Home and Login Screen

Practice Question: You'd like to set a Drawer Navigator for your app. From which directions on the screen can you set a drawer to open? Select all that apply

- Left
- Right

4.3. Reading: Exploring Drawer Navigation

This part is the same progress of installing and using Drawer as the above part.

4.4. Reading: Exercise: Configure Drawer Navigation

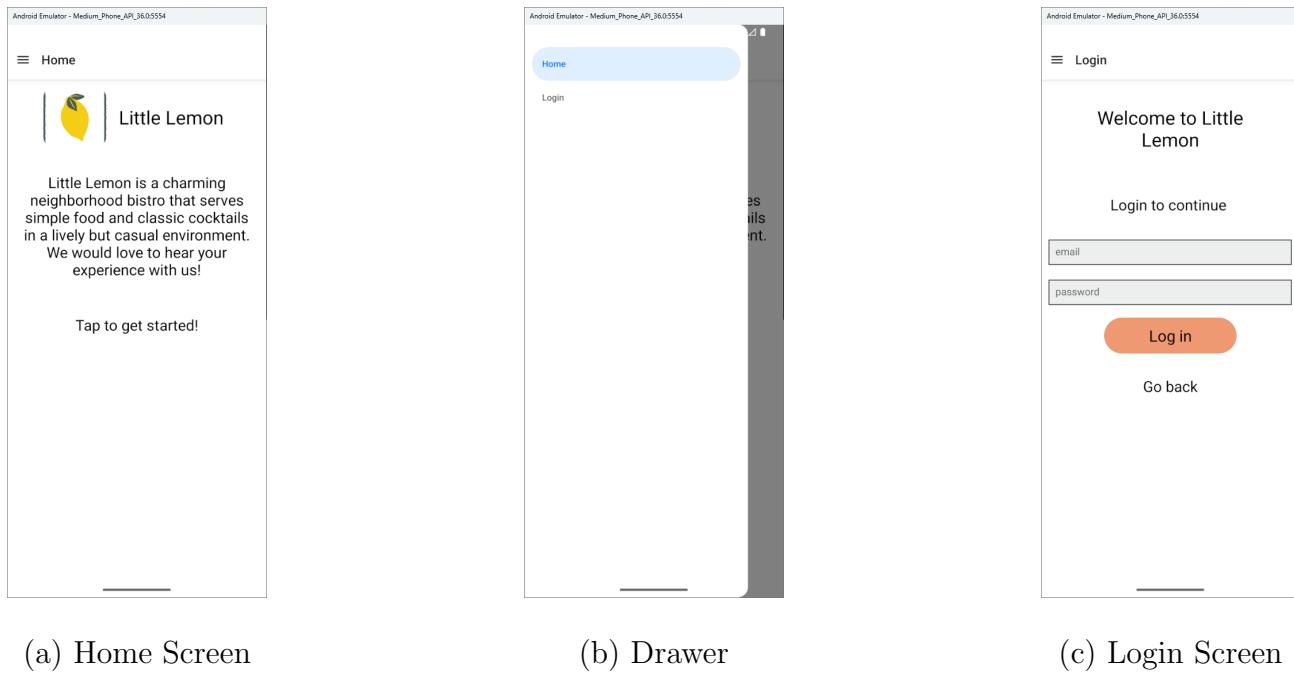


Figure 4.3: Drawer with Home and Login Screens

4.5. Practice Assignment: Self review: Configure Drawer Navigation

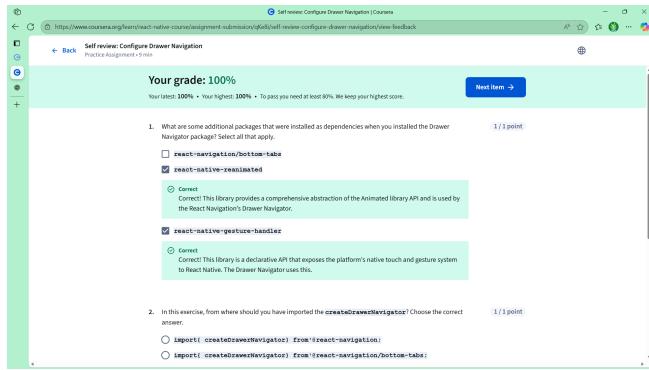


Figure 4.4: Self review: Configure Drawer Navigation

4.6. Practice Assignment: Knowledge check: Drawer Navigation

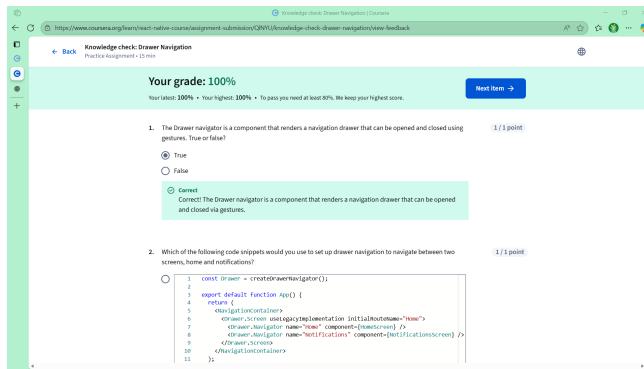


Figure 4.5: Knowledge check: Drawer Navigation

4.7. Video: Module Summary: React Navigation

- Introduction to React Navigation
 - What is React Navigation
 - Transition between screens
 - Benefits and Trade off
 - Install and setup React Navigation
 - Create and configure the Stack Navigator
 - Build Stack Navigator
 - Pass props to screen
 - Setup routes
- Moving between screens
 - navigate
 - push
 - goBack
 - popToTop

- Move between screens in a stack
 - Go back to previous screen
 - Configure a header bar
- Tab Navigation
 - What is Tab Navigation
 - Apply and configure a Tab navigator
 - Create code for tab navigation
 - Customize your tab navigator
 - Drawer Navigation
 - What is Drawer navigation
 - Apply and configure a Drawer navigator

4.8. Graded Assignment: Module Quiz: React Navigation

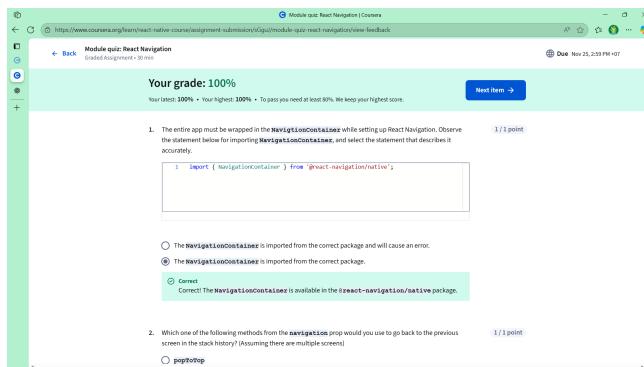


Figure 4.6: Module Quiz: React Navigation