MOB306 - React Native

Slide 5 - Lab 5

1. Lý thuyết
   - Giới thiệu về Navigation Document
   - Stack Navigation
   - Bottom Tab Navigation
   - Sử dụng axios để gọi API
   - Sử dụng AsyncStorage
2. Lab 5
   - Làm app như thiết kế Figma,
   - Sinh viên thiết kế 2 màn hình Đăng nhập và Đăng ký, sử dụng Stack navigation để chuyển qua lại
   - Sinh viên thiết kế màn hình Trang chủ, Tìm kiếm, Thông tin cá nhân, sử dụng Bottom Tab navigation để chuyển màn hình
   - Có dùng API để đăng nhập, đăng ký
   - Có dùng AsyncStorage để lưu dữ liệu

**Cài đặt thư viện:*

npm install @react-navigation/native

**Cài thêm dependencies*

- Expo: **npx expo install react-native-screens react-native-safe-area-context**
- Base: **npm install react-native-screens react-native-safe-area-context**

Stack Navigation

*Cài đặt thư viện

npm install @react-navigation/native-stack

1. Khai báo

Trong file **App.js** cần khai báo các Screen cần chuyển. Theo mặc định Screen nào được khai báo **ĐẦU TIÊN** sẽ được chạy lên đầu tiên khi run app

```
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import React from 'react';
import {
  SafeAreaView,
  Text,
  View
} from 'react-native';
import Screen1 from './src/components/Screen1';
import Screen2 from './src/components/Screen2';

const Stack = createNativeStackNavigator();

const App = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Screen1" component={Screen1} />
        <Stack.Screen name="Screen2" component={Screen2} />
      </Stack.Navigator>
    </NavigationContainer>
  );
};

export default App;
```

Ta có thể quy định Screen nào được chạy lên **ĐẦU TIÊN** thông qua thuộc tính **initialRouteName**

```
    <NavigationContainer>
      <Stack.Navigator initialRouteName='Screen1'>
        <Stack.Screen name="Screen1" component={Screen1} />
        <Stack.Screen name="Screen2" component={Screen2} />
      </Stack.Navigator>
    </NavigationContainer>
```

2. Chuyển từ màn hình này sang màn hình kia

```
const Screen1 = (props) => {
    //khi khai báo các Screen trong navigation ở file App.js, mặc định trong props của mỗi Screen sẽ có biến navigation
    const {navigation} = props;

    const ClickNe = () =>{
        //Truyền tên Screen muốn chuyển qua vào navigate
        navigation.navigate('Screen2');
    }

  return (
    <View>
      <Text>Screen 1</Text>
      <Pressable onPress={ClickNe}>
        <Text>Chuyển qua màn hình 2</Text>
      </Pressable>
    </View>
  )
}
```

3. Trở về màn hình trước

```
const Screen2 = (props) => {
  const {navigation} = props;

  const ClickTroVeNe = () =>{
    //Cách 1
    navigation.navigate('Screen1');
    //Cách 2
    navigation.goBack();
  }

  return (
    <View>
      <Text>Screen 2</Text>
      <Pressable onPress={ClickTroVeNe}>
        <Text>Trở về màn hình 1</Text>
      </Pressable>
    </View>
  )
}
```

4. Truyền dữ liệu từ màn hình này sang màn hình kia

*Gửi dữ liệu

```
const Screen1 = (props) => {
    const {navigation} = props;

    const ClickNe = () =>{
        navigation.navigate('Screen2', {name: 'Nguyễn Văn A', old: 26});
    }

  return (
    <View>
      <Text>Screen 1</Text>
      <Pressable onPress={ClickNe}>
        <Text>Chuyển và truyền dữ liệu qua màn hình 2</Text>
      </Pressable>
    </View>
  )
}
```

*Nhận dữ liệu

```
const Screen2 = (props) => {
  const {navigation, route} = props;
  const {params} = route;

  return (
    <View>
      <Text>Họ tên: {params.name}</Text>
      <Text>Tuổi: {params.old}</Text>
    </View>
  )
}
```

Tips:

Trong trường hợp ở đoạn code trên, giả sử như dữ liệu truyền qua không có hoặc không nhận được sẽ dẫn đến tình trạng lỗi app. Vì vậy để tránh tình trạng trên chúng ta có thể kiểm tra dữ liệu có tồn tại hay không bằng cách thêm dấu "?" vào sau chữ params. Nếu dữ liệu tồn tại sẽ hiển thị bình thường, ngược lại sẽ không hiển thị.

```
const Screen2 = (props) => {
  const {navigation, route} = props;
  const {params} = route;

  return (
    <View>
      <Text>Họ tên: {params?.name}</Text>
      <Text>Tuổi: {params?.old}</Text>
    </View>
  )
}
```