React Native Navigation

In this section we will start to see how to create multiple pages and how to move from one to another page. You will be needing react-navigation for that, start by creating a new project and install the following in your project folder:

npm install react-navigation
expo install react-native-gesture-handler react-native-reanimated react-native-screens

This will install the react-navigation that can be used in your application.

Stack Navigation

The first navigation that we will discover is navigation stack. In react, everytime you implement a new type of navigation, you will need to install a new library.

1) Run the following command in CMD:

npm install react-navigation-stack

2) We will start by creating different components / pages for your project. To do that, create 3 components, home, about and reviewDetail. Each of the component will only have a simple text as below:

```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default function About() {
  return (
    <View style={styles.container}>
      <Text>This is About page</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
```

```
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

Example of About file. Repeat the same for Home and reviewDetail pages.

3) Once you have that, we will create a routing page. Create a new file, call it homeStacks.js. This file will contain route definition to all components in your project. Define it as follows:

```
import {createStackNavigator} from 'react-navigation-stack';
import { createAppContainer } from "react-navigation";
import Home from '../screens/home';
import ReviewDetails from '../screens/reviewDetails';
import About from '../screens/about';
const screens = {
  Home: {
    screen: Home
  },
  ReviewDetails: {
    screen:ReviewDetails
  },
  About: {
    screen:About
  }
}
const HomeStack = createStackNavigator(screens);

export default createAppContainer(HomeStack);
```

This is where you will define each component by its name. If you have more screens and routes you will modify and add the definition here.

4) Inside the home page, add a button that will later link us from the first page to the second page.

```
return (
```

```
    <View style={styles.container}>
      <Text>This is Home page</Text>
      <Button onPress={()=>console.log('To do later')} title="Go to about page"/>
    </View>
  );
```

5) We will now create the link from Home page to Review Detail page. There are two things need to be done:
   a) In Home component declaration, you need to modify to be as follows, to add navigation declaration:

```
export default function Home({navigation}) {

}
```

   b) Then remove the console.log and replace it with navigation code as below:

```
<Button onPress={()=> navigation.push('ReviewDetail') title="Go to Review page"/>
```

   c) Replace the code in App.js to be as below:

```
export default function App() {
 return (
  <Navigator/>
 );
}
```

6) Verify that when the button is pressed you will be brought to the ReviewDetail page. You will notice that the navigation bar will be created for you automatically.
7) We will now add another button in the ReviewDetail component, that will be labelled be Return Home. Then you will create a link from ReviewDetail component to Home component, slightly change than previous example:

```
export default function ReviewDetails({navigation}) {
  return (
    <View style={styles.container}>
      <Text>This is Review Details page</Text>
      <Button onPress={()=> navigation.goBack()} title="Go back to main page"/>
    </View>
  );
```

```
}
```

8) We will now see how to pass data from one page to another page. On the Home page, add a TextInput. The input will ask for the user's name and you will store it inside the name state. *(Add note for your answer below)*

9) From the main page, you will pass the name to the second page (Review) page. Add a second parameter in the push function. The argument will be an object with key and value as below:

```
<Button onPress={()=> navigation.push('ReviewDetails',{'name':name})} title="Go to review page"/>
```

10) On the second page, you will retrieve the item passed, edit the review page to be as follows:

```
export default function ReviewDetails({navigation}) {
  return (
    <View style={styles.container}>
      <Text>{navigation.getParam('name')}</Text>
      <Button onPress={()=> navigation.goBack()} title="Go back to main page"/>
    </View>
  );
}
```

11) You may improve the navigation flow, for example by adding colour to the header. Modify the code in homeStack.js to be as follows:

```
const HomeStack = createStackNavigator(screens, {defaultNavigationOptions:{
  headerStyle:{backgroundColor:'#eee'}
}});
```

Stack Navigation (project)

You will change the homepage to show a List of reviews. For example one line of the reviews will be as follows:

Name : Danny
Rating: 3
Review: The class is so so..

Another line will be as follows:

Name : Suzy
Rating: 4
Review: I love how the presenter presents the topic.

Once the item of the review is pressed, it will open second page (Review Detail) that will show only the selected Review.