ชื่อ.....

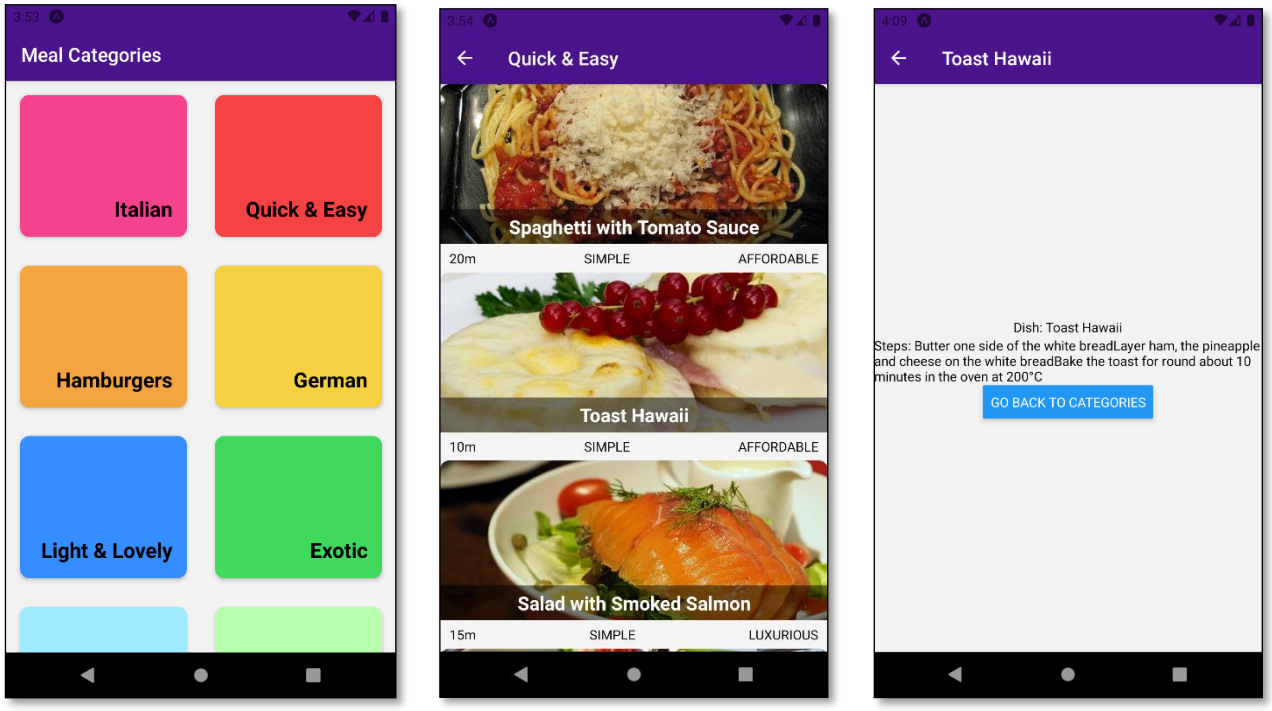
รหัสนักศึกษา.....

Lab 5 : Navigation (Part 1)

จงเขียนโปรแกรม MealApp ที่เป็นแอปพลิเคชันที่แสดงเมนูอาหารประเภทต่างๆ โดยมีหน้าจอทั้งหมด 3 หน้าจอ ได้แก่

- หน้าจอ Categories Screen เป็นหน้าจอแสดงประเภทอาหาร เช่น Italian, Quick & Easy, Hamburgers เป็นต้น
- หน้าจอ Category Meals Screen เป็นหน้าจอแสดงเมนูอาหารตามประเภทอาหารที่ผู้ใช้เลือก เช่น หากผู้ใช้เลือกอาหารประเภท Quick & Easy หน้าจอนี้จะแสดงเมนู Spaghetti with Tomato Sauce, Toast Hawaii เป็นต้น
- หน้าจอ Meal Detail Screen เป็นหน้าจอแสดงรายละเอียดของเมนูที่เลือก เช่น แสดงชื่อเมนู และวิธีการทำ เป็นต้น

ตัวอย่างหน้าจอ (Stack Navigation)



Categories Screen —(categoryId, categoryTitle)→ Category Meals Screen —(mealId, mealTitle, mealSteps)→ Meal Detail Screen

ชื่อ.....

รหัสนักศึกษา.....

โครงสร้างของโปรแกรม

- components
  - CategoryGridTile.js
  - MealItem.js
- data
  - dummy-data.js
- models
  - category.js
  - meals.js
- screens
  - CategoriesScreen.js
  - CategoryMealsScreen.js
  - MealDetailScreen.js
- App.js

หมายเหตุ นักศึกษาสามารถแก้ไขโค้ดโปรแกรมที่ได้เตรียมไว้ให้ (OnLearn) ได้ตามความเหมาะสม

ขั้นตอนการปฏิบัติการ

1. ทำการสร้างโปรเจคใหม่ และคัดลอกไฟล์โปรแกรมตามโครงสร้างข้างต้น
2. ติดตั้ง library และ dependencies เพื่อใช้ในการทำ Navigation ให้ครบถ้วน
   - expo install @react-navigation/native
   - expo install react-native-screens react-native-safe-area-context
   - expo install @react-navigation/native-stack
3. ไฟล์ต่างๆ มีรายละเอียดดังนี้
   - CategoryGridTile.js เป็นคอมโพเนนต์ที่ใช้แสดงประเภทอาหารแต่ละประเภท ในหน้า Categories Screen (แทน grid 1 ช่อง ที่บอกประเภทอาหาร)
   - MealItem.js เป็นคอมโพเนนต์ที่ใช้แสดงเมนูอาหารแต่ละเมนู ในหน้า Category Meals Screen (แทนเมนูอาหาร 1 เมนู)
   - dummy-data.js เป็นส่วนของข้อมูลประเภทอาหาร และเมนูอาหารทั้งหมด โดยเก็บเป็นอะเรย์ขององ็อบเจ็คประเภทอาหารและเมนูอาหาร

- category.js เป็นคลาสของประเภทอาหาร
- meals.js เป็นคลาสของเมนูอาหาร
- CategoriesScreen.js ทำหน้าที่แสดงประเภทอาหาร ซึ่งผู้ใช้สามารถเลือกประเภทอาหารดูได้
- CategoryMealsScreen.js ทำหน้าที่แสดงเมนูอาหารตามประเภทอาหารที่ผู้ใช้เลือกจากหน้า CategoriesScreen
- MealDetailScreen.js ทำหน้าที่แสดงรายละเอียดของเมนูที่เลือกจากหน้า CategoryMealsScreen
- App.js เป็นส่วนคอมโพเนนต์หลักที่เป็นส่วนกำหนดการทำ Navigation ของโปรแกรมมาทำการจัดการการเปลี่ยนหน้าจอ 3 หน้าจอข้างต้น

สร้างส่วนกำหนดการทำ Navigation (App.js)

ปรับปรุงโปรแกรม App.js เพิ่มเติม ดังนี้

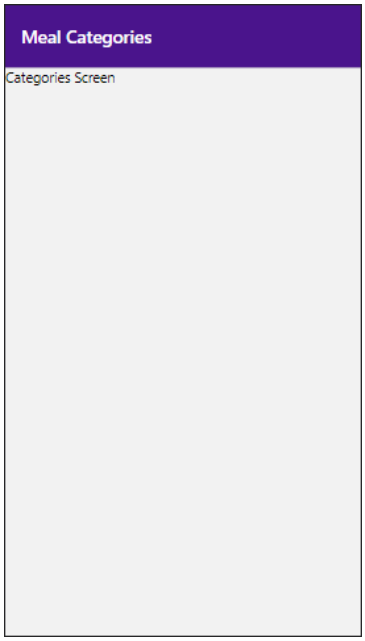
1. import คอมโพเนนต์ที่จำเป็นต่างๆ ในการสร้าง stack navigator รวมถึงหน้าจอต่างๆ เช่น NavigationContainer, createNativeStackNavigator, CategoriesScreen, CategoryMealsScreen และ MealDetailScreen เป็นต้น (Slide 13)
2. กำหนดการตั้งค่า Route ภายใต้คอมโพเนนต์ NavigationContainer โดยกำหนดให้ (Slide 17)
   - หน้าจอ CategoriesScreen มี Route name เป็น Categories
   - หน้าจอ CategoryMealsScreen มี Route name เป็น CategoryMeals
   - หน้าจอ MealDetailScreen มี Route name เป็น MealDetail
3. กำหนด title ของหน้าจอ CategoriesScreen ให้เป็น Meal Categories ผ่านการตั้งค่า options (Slide 23-24)
4. กำหนดสไตล์ของส่วนเฮดเดอร์ของแอปพลิเคชันในทุกหน้าจอให้เหมือนกัน ด้วยการกำหนดค่า screenOptions ในคอมโพเนนต์ Navigator ดังนี้ (Slide 25-26)

   ```
   headerStyle: { backgroundColor: "#4a148c", },
   headerTintColor: "white",
   ```

   ซึ่งเป็นการกำหนดสีเฮดเดอร์เป็นสี #4a148c และตัวอักษรเป็นสีขาว
5. ทดลองรันโปรแกรมจะพบว่าหน้าจอ CategoriesScreen จะถูกแสดงขึ้นดังนี้

ชื่อ.....

รหัสนักศึกษา.....



สังเกตว่า จะมีการแสดงข้อความว่า “Categories Screen” เท่านั้น ซึ่งนักศึกษาต้องทำการดึงข้อมูลจาก dummy-data.js มาแสดงต่อไป นอกจากนี้ ส่วนเฮดเดอร์มีหัวข้อ “Meal Categories” และมีการแสดงสีตาม screenOptions ที่ได้กำหนดใน MealsNavigator.Navigator

เขียนโปรแกรมเพื่อแสดงหน้า CategoriesScreen (CategoriesScreen.js)

1. ศึกษาและทดลองใช้ FlatList ในการแสดงประเภทอาหารทั้งหมดที่เก็บไว้ในตัวแปร CATEGORIES ในไฟล์ dummy-data.js (สังเกตว่ามีการ import CATEGORIES มาใช้งาน)
    - CATEGORIES เป็นอะเรย์ของอ็อบเจ็ค Category ซึ่งมี id, title และ color เป็นคุณลักษณะ (ดูไฟล์ /models/category.js ประกอบ)
    - Property data ใน FlatList เป็นการกำหนดข้อมูลที่ต้องการนำมาแสดงใน FlatList ในที่นี้ คือ CATEGORIES นั่นเอง
    - Property renderItem เป็นการกำหนดเมธอดที่ทำหน้าที่แสดงข้อมูลแต่ละตัวใน data ในที่นี้โปรแกรมจะทำการเรียกเมธอด renderGridItem ขึ้นมาทำงาน
    - Property numColumns เป็นการกำหนดจำนวนคอลัมน์ที่จะแสดงใน FlatList
    - พิจารณาเมธอด renderGridItem จะพบว่ามีอาร์กิวเมนต์ itemData หมายถึงข้อมูลแต่ละตัวที่อยู่ใน data นักศึกษาสามารถอ้างอิงข้อมูลนั้นได้ด้วยคำสั่ง itemData.item ซึ่งในที่นี้คือ อ็อบเจ็ค Category ซึ่งสามารถอ้างอิงถึงคุณลักษณะของอ็อบเจ็คได้ เช่น itemData.item.id เป็นต้น

- ใน return() ให้นักศึกษาคอมเมนต์โค้ด <View>...</View> และทำเรียกใช้ <FlatList> แทน
- ทดลองรันโปรแกรม ได้ผลดังนี้

จะสังเกตว่า ได้มีการแสดงประเภทอาหารต่างๆ แล้ว แต่อยู่ในรูปแบบของข้อความเท่านั้น ต่อไปนักศึกษาต้องทำให้มีการแสดงประเภทอาหารตามรูปแบบที่กำหนด และเมื่อกดที่ชื่อรายการอาหารแล้วจะทำการเปลี่ยนหน้าจอไปที่ Category Meals Screen

เขียนโปรแกรมเพื่อปรับปรุงการแสดงผลหน้า Categories Screen (CategoriesScreen.js)

1. ศึกษาการทำงานของ CategoryGridTile.js ซึ่งมีประเด็นสำคัญดังนี้ (นศ.ไม่จำเป็นต้องแก้ไขโค้ดโปรแกรม CategoryGridTile.js)
   - CategoryGridTile เป็นคอมโพเนนต์ที่ใช้แสดงประเภทอาหารแต่ละประเภทในหน้า Categories Screen (การเรียก CategoryGridTile 1 ครั้ง คือ การแสดง grid 1 ช่อง ที่บอกประเภทอาหาร เช่น grid ช่องสีชมพู-Italian เป็นต้น)

   - CategoryGridTile มี property title ซึ่งเป็นชื่อของประเภทอาหาร เช่น Italian
   - CategoryGridTile มี property color ซึ่งเป็นสีสำหรับแสดงในประเภทอาหารนั้น เช่น สีชมพู

- CategoryGridTile มี property onSelect ซึ่งจะทำการเปลี่ยนหน้าจอไปยังหน้า Category Meals Screen
- CategoryGridTile มีการเรียกใช้คอมโพเนนต์ TouchableOpacity ซึ่งเมื่อถูกกดบริเวณนี้จะทำการเรียก property onSelect นอกจากนี้ ก็ได้มีการนำค่าที่รับมาผ่าน props มาใช้ในการแสดงข้อมูล

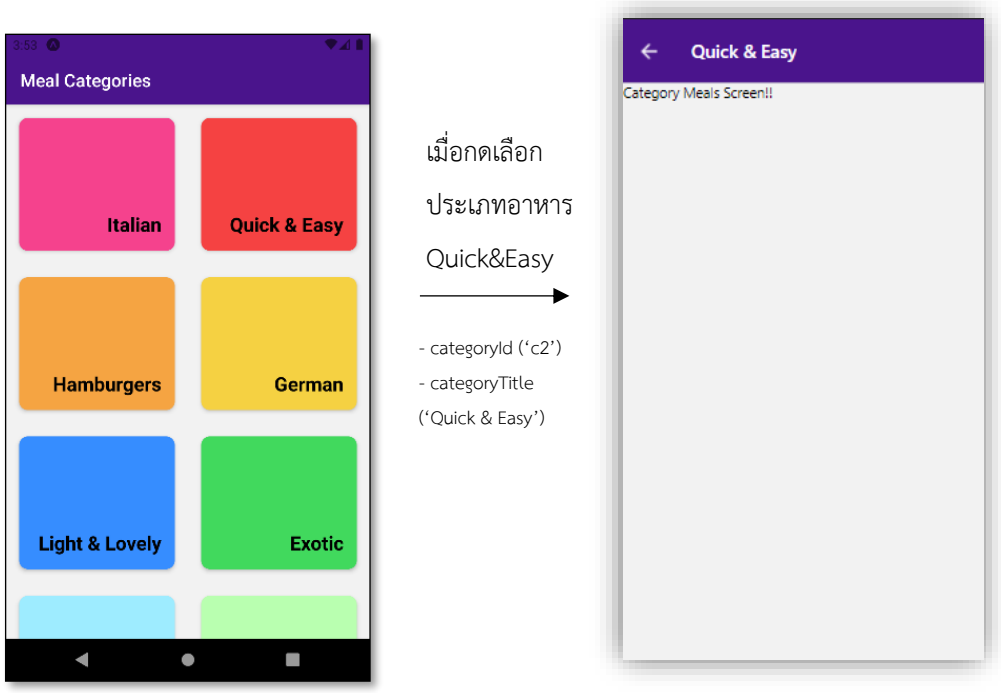
2. ทำการปรับปรุงโปรแกรม CategoriesScreen.js ในส่วนของเมธอด renderGridItem()
   - ให้นักศึกษาคอมเมนต์โค้ด <View>...</View> และทำเรียกใช้ <CategoryGridTile> แทน
   - เมธอด renderGridItem มีอาร์กิวเมนต์ itemData ซึ่งเป็นข้อมูลแต่ละตัวที่จะแสดงใน FlatList ให้นักศึกษาใช้ itemData นี้ ปรับปรุงโปรแกรมใน <CategoryGridTile> ให้สมบูรณ์
   - Property onSelect จะทำการเปลี่ยนหน้าจอไป Category Meals Screen พร้อมกับส่งพารามิเตอร์ชื่อ categoryId ด้วยค่า id ของประเภทอาหารนั้นๆ เช่น ถ้ากดที่อาหาร Italian จะเปลี่ยนหน้าจอ พร้อมกับส่ง categoryId เป็น 'c1' และส่งพารามิเตอร์ชื่อ categoryTitle ด้วยชื่อประเภทของอาหาร เช่น Italian เป็นต้น (Slide 27)
   - ทดลองรันโปรแกรม จะแสดงหน้าตาดังรูป

จะเห็นว่าในส่วนของเฮดเดอร์แสดงข้อความ "CategoryMeals" ตาม Route name ที่กำหนดใน MealsNavigator และในส่วนของเนื้อหาแสดงเพียงข้อความ Category Meals Screen เท่านั้น

3. ปรับปรุงข้อความเฮดเดอร์ให้แสดงชื่อประเภทอาหารที่ถูกเลือกมา โดยแก้ไขไฟล์ App.js ด้วยการกำหนด options ในคอมโพเนนต์ Screen ของหน้าจอ CategoryMealsScreen ให้แสดง title ของหน้าจอ เป็นชื่อประเภทอาหารที่เลือกมา เช่น Italian, Quick & Easy เป็นต้น (Slide 30-31) ตัวอย่างเช่น

ชื่อ.....

รหัสนักศึกษา.....



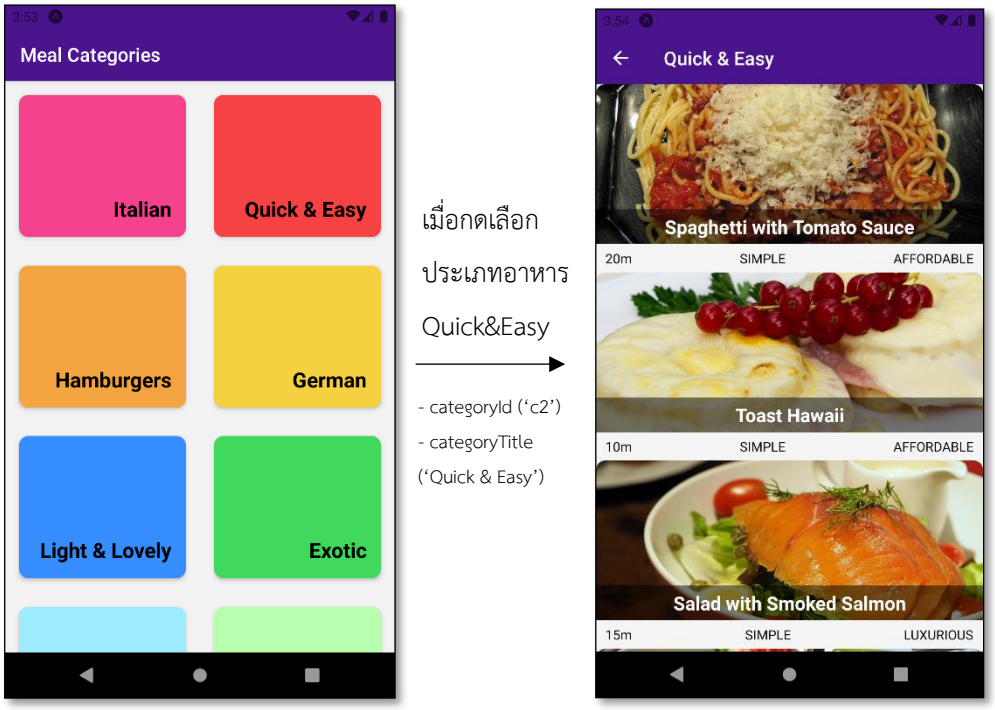
เขียนโปรแกรมเพื่อแสดงหน้า CategoryMealsScreen (CategoryMealsScreen.js)

1. ทำการดึงข้อมูลรายการเมนูอาหารที่อยู่ในประเภทอาหารที่เลือกไว้
    - รับข้อมูล categoryId ของประเภทอาหารที่ส่งมาจากหน้า CategoriesScreen แล้วเก็บในตัวแปร catId ภายในคอมโพเนนต์ CategoryMealsScreen แทน (Slide 28-29)
    - ใช้ filter() ทำการกรองเอาอ็อบเจ็ค Meal (เมนูอาหาร) ที่มี categoryId ตรงกับ catId (ให้สังเกตข้อมูลใน dummy-data.js ว่าเมนูอาหารหนึ่งๆ สามารถถูกจัดอยู่ในอาหารหลายประเภท (category) ได้ (ให้ศึกษาการทำงานและเอาคอมเมนต์ส่วนของโปรแกรมในการกำหนด displayedMeals ออก)
    - ผลลัพธ์จากการทำ filter() จะเป็นอะเรย์ของอ็อบเจ็ค Meal ที่มี categoryId ตรงกับ catId เก็บในตัวแปร displayedMeals
2. ทำการ**แสดงรายการเมนูอาหารที่กรองได้**มาแสดงที่หน้าจอด้วย FlatList (แนวคิดคล้ายกับการแสดงรายการประเภทอาหารที่ได้อธิบายไว้ก่อนหน้านี้)
    - โดยเมื่อมีการกดที่เมนูอาหาร จะมีการเปลี่ยนหน้าจอไปยังหน้า MealDetailScreen พร้อมกับส่งพารามิเตอร์ต่างๆ เพื่อนำไปแสดงรายละเอียดต่อไปตามที่เห็นสมควร (นศ.สามารถกำหนดพารามิเตอร์ได้อย่างอิสระ)

ชื่อ.....

รหัสนักศึกษา.....

3. ทดลองรันโปรแกรม เมื่อเลือกประเภทอาหาร โปรแกรมจะแสดงรายการอาหารตามประเภทอาหารที่เลือกนั้นๆ



เขียนโปรแกรมเพื่อแสดงหน้า MealDetailScreen (MealDetailScreen.js)

1. ปรับปรุงข้อความเฮดเดอร์ให้แสดงชื่อเมนูอาหารที่ถูกเลือกมา โดยแก้ไขไฟล์ App.js ด้วยการกำหนด options ในคอมโพเนนต์ Screen ของหน้าจอ MealDetailScreen ให้แสดง title ของหน้าจอ เป็นชื่อเมนูอาหารที่เลือกมา เช่น Spaghetti with Tomato Sauce เป็นต้น (Slide 30-31)
2. ทำการดึงข้อมูลเกี่ยวกับเมนูอาหารที่ผู้ใช้ที่เลือกไว้ (ส่งมาจาก CategoryMealsScreen) (Slide 28-29)
3. ทำการแสดงชื่อเมนูอาหาร และวิธีการทำอาหารของเมนูอาหารที่ได้เลือกไว้แสดงที่หน้าจอ
4. เมื่อกดปุ่ม Go Back to Categories ให้ไปการแสดงหน้า CategoriesScreen ที่แสดงประเภทอาหารต่างๆ (หน้าแรก)