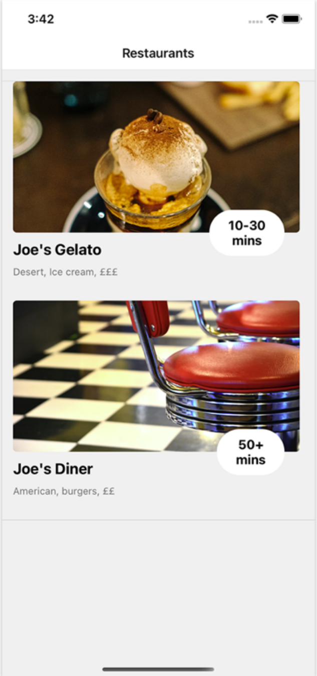
Creating a food app



In this assignment, we will make a food app that displays multiple restaurants, each with unique menus. This task will involve using a few different types of menu cells and app navigation. I suggest following the exact content indicated in this tutorial, including images, and later you can customise it and create your own shops!

For now, all the content is hard-coded, meaning we can’t dynamically load the menus or images from databases or online, but don’t worry, we cover that later in this module. For now, we are just focusing on creating the UI elements.

How do I submit this?

This graded assessment is one of five graded assessments which will make up your midterm assignment submitted in week 10. I strongly advise you to complete it now as it is crucial to expanding your understanding of the subsequent weeks. Once you have completed it, store it somewhere safe and do not share it with others. Instead, you can upload it with the other assignments during week 1

Can I customise this assignment?

Yes! I encourage you to make this food app your own. I provide steps to allow you to build the overall structure of the app, and I suggest you build it using my exact instructions and guides first before adapting and adding new elements so that it feels like your own app. The marking rubric is below, so you can make sure you include the required features.

Steps to complete:

Firstly, we will be using App as a functional component with this demo; this means you don’t need to create a class extending App; instead, we will generate a blank project and start adding to it.

So first, set up a new React Native project using expo. Refer back to the setup instructions earlier in this module if you have forgotten how to do this.

(1) Open the project in your IDE and select the App.js file.

(2) Import required libraries

- Import NavigationContainer from @react-navigation/native
- Import createStackNavigator from @react-navigation/stack
- Import Cell, Section, TableView from react-native-tableview-simple.
- Remember, you will have to import other modules later in these steps, and you might need to install the modules above. See my previous videos for guidance on this.

(3) Setup the navigation container

- Wrap the contents of App() in a NavigationContainer, inside of which should be a Stack.Navigator and two Stack.Screen’s called “Restaurants” and “Menu”. Both *Restaurants* and *Menu* should have components, so make them with blank Views for the time being. Remember, you need to create a new Stack const that calls createStackNavigator(); You can refer back to the navigation section of this course for guidance on this.

(4) The Home Screen component

- Inside of your Home Screen component function, create a ScrollView. Inside of this should be a TableView and a Section. Set the section name to an empty string and use props to hide the separator and change the separator tint colour to #ccc. Look at this list of available props.

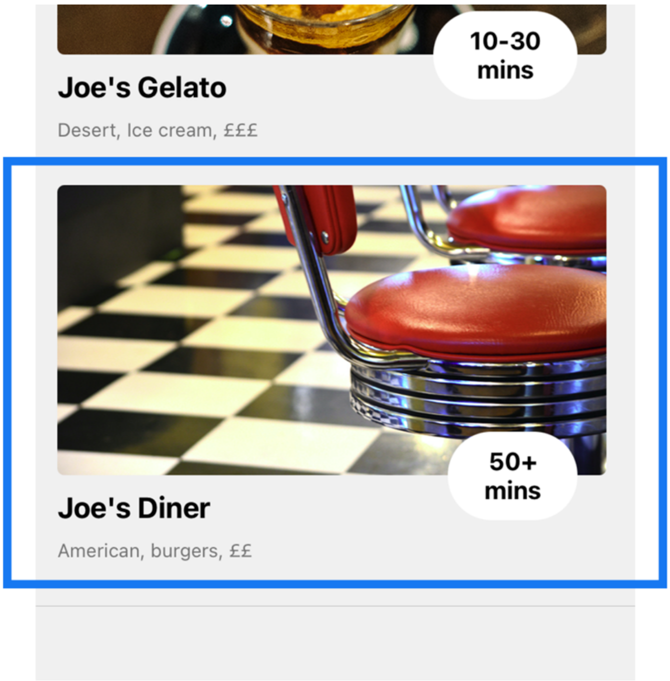
(5) The HomescreenCell

Next, we want to create a custom cell that we can use for the main restaurants UI. Create a new HomescreenCell const. Inside define a standard cell, and we will customise it next.

- Make sure to pass props through to the HomeScreenCell const and embed them within the Cell. Remember, you can use the syntax {...props} for this.
- Create the HomeScreenCell within the HomeScreen Table Section.
- To make editing simple for us, we will pass props through to each custom cell within the HomeScreen Section, and this will allow us to define our names, images and text and have them reflected in the custom cell. Define a set of props within the HomeScreen cell element similar to below:

  ```
  imgUri={require('./images/ice-cream-header.jpg')}
  ```

- Make sure that you download and place the images in the same location. You can use any image, although I recommend finding royalty-free images or your own. Make sure they are also JPEG files.
- Now, let's create the custom cell which will display this information. In this app, we use a new cell for each restaurant (see below):



- In your HomescreenCell, use props to set the height to 290px, background colour to transparent and highlight colour to #ccc.

Inside of the cellContentView add the following:

- An image (for the header). This image should use the imageUri prop as the source and styled to look like the example image.
- Text wrapped in a View containing the eta prop also styled to look like the example image. Hint: You'll need to use absolute positioning for this.
- Two Texts, one for the headline, using the title prop and one for the subtitle, using the tagline prop.
- Once you have finished, you should have a cell that looks like the example image above.

(6) Navigation

We want the app to display a second screen every time we press a restaurant cell.

- Add a prop called *action* to the HomescreenCell within your homescreen function. Within it, pass a function navigating to "Menu". You can do this as you would if we were about to add it to onPress. We are just passing this as a prop so we can call it directly within the custom cell.
- In your HomescreenCell add an onPress which calls the prop.action. Note, you need to wrap the prop in curly brackets. E.g. onPress={prop.action}
- Now when you press the cell, it should navigate to a new page.

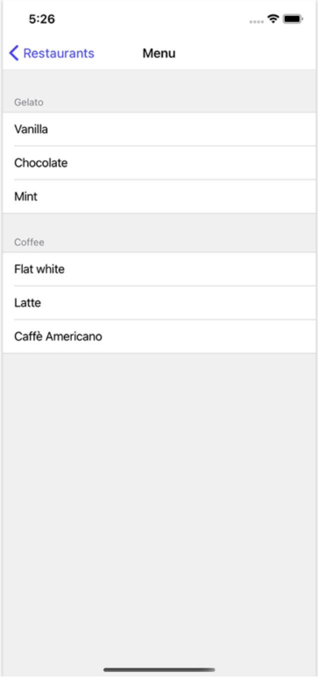
(7) The menu

Next, we are going to make a basic table view menu for items for each restaurant.

- We need to write some information about the menu to pass to navigation and the next page to display it in a table. To do this, add a second parameter to navigate so that it sends an object. I suggest a format like the below:

  ```
  { items: [{ "title":"Gelato", "contents":[{"title":"Vanilla"}] }] }
  ```

- This object contains the sub-items of the menu. When interpreted as a table, it will look like below:



Gelato

Vanilla

Chocolate

Mint

Coffee

Flat white

Latte

Caffè Americano

- Now let's display it as a menu. In your detailsScreen add a ScrollView, inside of which is a TableView.
- Inside the TableView iterate through items, creating a new section each time.
- Inside that iterative section, loop through the *contents* and create a new basic cell. This cell's title should be the content's title.
- That will programmatically create cells within the table based on the object we passed in during navigation. Note that you will need to add more items to the object to look like the screenshot above.
- Add more items to this restaurant, exploring using multiple Sections.
- Add in another restaurant, duplicate the HomescreenCell and make another restaurant with a different image, title, subtitle and menu.
- Customise your app. Feel free to add other customisations. Perhaps you could do the following:
- Add more restaurants to the app.
- Stylise the UI.
- Add images to the menu items, with out of stock disabled options
- Add additional functionality so that something happens when you press a menu item.

Mark scheme

Description	Marks
Navigation	
No attempt made	0
Navigation libraries were imported correctly	1
Navigation was attempted but doesn't function	2
Navigation is working as intended, however is not customised	3
Navigation is working exactly as expected	4
Navigation works with custom table presses	5
Table	
No attempt made	0
The table view is implemented	1
The table view uses basic cell components	2
The table view correctly uses custom components	3
Custom cells use props correctly	4
Table view uses for loops for cell creations, fueled by an object	5
UI	
No attempt made	0
Basic UI elements used, incorrect positioning	1
Somewhat similar UI elements, almost similar to screenshot	2
Correct UI, identical to screenshot	3
Basic improvements have been made to the UI	4
Significant improvements made to the UI	5
Additions	
No improvements have been made	0
Minor improvements have been made	1
More significant improvements have been made to increase functionality	2
Significant improvements throughout the app	3
Improved functionality and styling of a high-quality	4
Drastically improved application	5