GOPHER
INDUSTRIES

Table of Contents

Background ***3***

Usage Guide ***4***

Component Usage ***7***

Bottom Menu Bar **7**

Food Item with Add Button **7**

Preference Page Item Row **7**

Meal Item with Add Button **7**

Background

The Figma wireframes provided by previous trimesters are thorough and in-depth. It is common for developers to spend large periods of time working on coding the design elements that are shown in Figma.

The Trimester 3 2022 team researched and found a Figma plugin known as Locofy. Locofy is a design-to-code software that worked in with our current tech stack in terms of converting Figma wireframes to React Native code.

Whilst the code provided by Locofy is in fact a standalone React app in and of itself, the decision was made to continue with our current code base, and cherry-pick elements from the Locofy-generated code to insert into our pre-existing React pages for each screen. This ensured that content created by previous teams as well as by the Trimester 3 2022 team was not discarded, whilst still assisting with the front-end design element of the code.

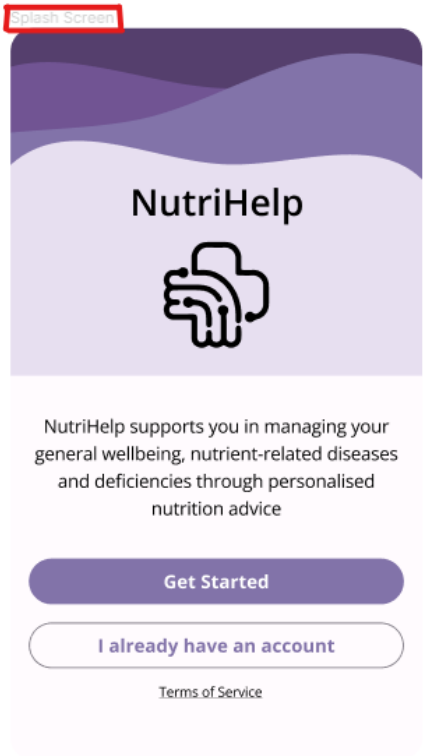
Some screens were being developed by team members, however due to the scope of the project and the length of time available, there are elements of the Figma wireframe that were not completed by the team in regards to functionality and final design touches.

The purpose of the 'NutriHelp Formatting .js files' folder (found in T3 2022 > Development > Front-End) is to assist future team members as they begin to work on the remaining screens that require development.

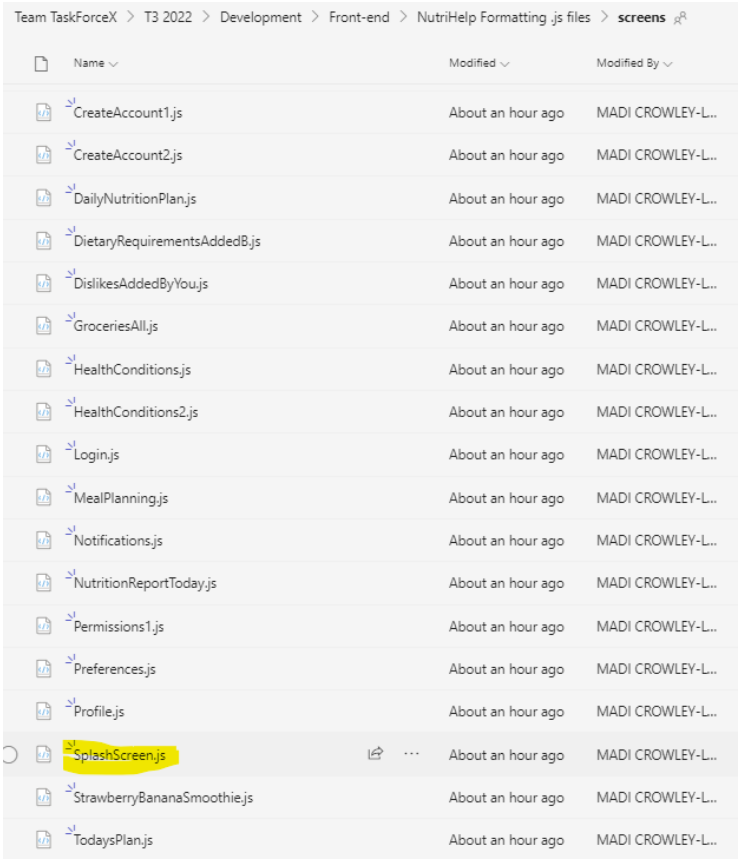
Usage Guide

When a team member assigns themselves a Trello ticket related to changing code for a screen within the app, they should follow the below:

1. Check the *Taskforce X | Wireframe Iteration 1* to find the screen linked to the assigned Trello ticket
2. Check the name of the frame that matches your assigned ticket, found on the top right of the frame (see figure below)



3. Within the 'NutriHelp Formatting .js files' folder, in the 'screens' folder will be a corresponding .js file that is the Figma wireframe converted into React Native.



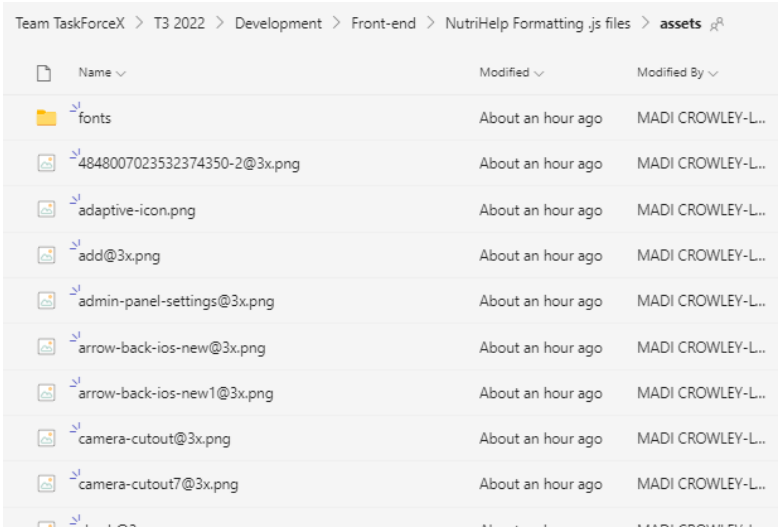
4. The first part of each file is the various elements as tagged and converted to a functional screen, whilst the second part is the CSS formatting. Each tagged and

converted element of the wireframe will have a corresponding CSS section, as shown below:

```
<Text style={styles.termsOfService}>Terms of Service</Text>
<Pressable
  style={styles.getStartedButton}
  onPress={() => navigation.navigate("WelcomeScreen1")}
>
  <Text style={styles.labelText}>Get Started</Text>
</Pressable>
<Pressable
  style={styles.haveAnAccountButton}
  onPress={() => navigation.navigate("Login")}
>
```

```
termsOfService: {
  position: "absolute",
  top: 571,
  left: 119,
  fontSize: 12,
  textDecoration: "underline",
  letterSpacing: -0.1,
  lineHeight: 18,
  fontFamily: "Open Sans",
  color: "#000",
  textAlign: "center",
  width: 112,
  height: 21,
},
labelText: {
  position: "relative",
  fontSize: 16,
  letterSpacing: 0,
  lineHeight: 20,
  fontWeight: "700",
  fontFamily: "Open Sans",
  color: "#fff",
  textAlign: "center",
},
getStartedButton: {
  position: "absolute",
  top: 464,
  left: 16,
  borderRadius: 100,
  backgroundColor: "#8273a9",
  width: 328,
  overflow: "hidden",
  flexDirection: "column",
  paddingHorizontal: 24,
  paddingVertical: 10,
  alignItems: "center",
  justifyContent: "center",
},
```

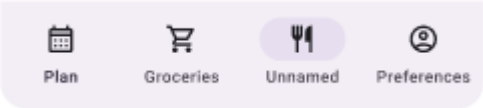
5. With the majority of elements having clearly identifiable names, team members can select the relevant code snippets and insert them into the pre-existing code blocks without having to significantly readjust any front-end or back-end functionality.
6. It should be noted that a large portion of the screens do have screen-changing functionality built in e.g. in the above example, the buttons have been set up to then take the user to the linked screen. Care should be taken if porting these aspects, to ensure that the screens being linked have been added to the App.js file, or errors may occur.
7. If the element being selected involves an image file, these can be found in the ‘assets’ folder as shown below. These images should be taken over to the main image asset location in the working code project and any content referencing these images should be updated accordingly to avoid errors.



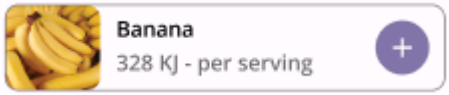
Component Usage

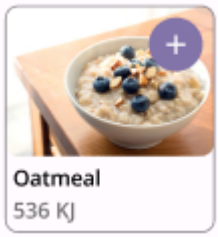
There are a number of reusable components in the Figma wireframe designs. These have been saved as components in the Components folder, and are shown below. Should a future design or front-end development utilise one of these components, the code should be ready-made and able to be extracted or referenced in .js files.

Bottom Menu Bar



Food Item with Add Button



Preference Page Item Row

Meal Item with Add Button