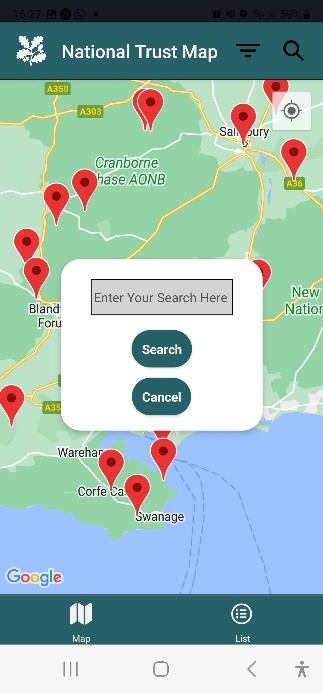
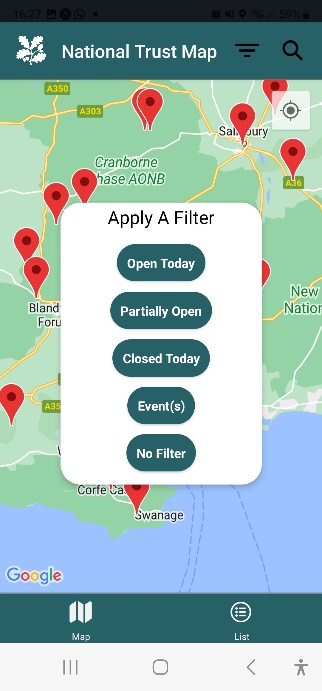
Assessment 1: Ubiquitous Computing - Application Redesign

# PROTOTYPE Overview

My prototype was written using React Native. Upon first opening the redesign you are greeted with a bottom tab navigation with 2 options “Map” and “List”.

Map

Description automatically generated“Map” is automatically selected by default and consists of a map view provided by Google Maps with marked locations that indicate National Trust locations.



By pressing the marker you are greeted with a custom callout very similar to the original app.

Once the callout is pressed you are greeted with the “Details” screen (more on this later).

Also at the top right are two pressable buttons: “Filter” and “Search”, both of which create individual Modals when pressed that behave as expected.

Figure 1. App screenshot to show “Map” screen

Graphical user interface

Description automatically generatedA close up of a calculator

Description automatically generated with medium confidenceGraphical user interface, text, application, chat or text message

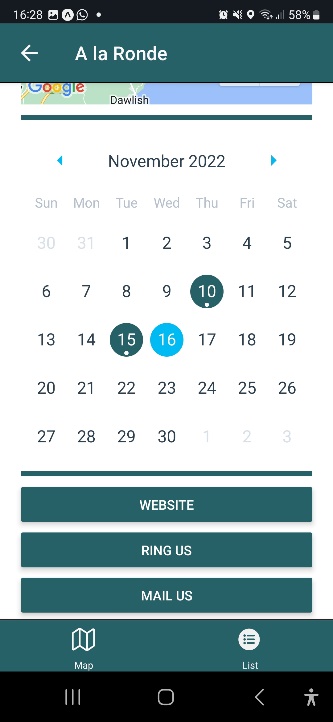
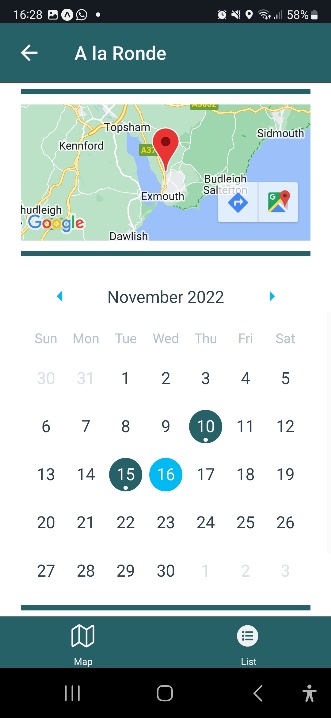
Description automatically generated

The “List” screen displays a list of all National Trust locations just like the list in the original, except combined with the “What’s On” section. It’s a scrollable list of all locations and some information such as the title, an image and brief description with the same buttons “Filter” and “Search” from the Map screen. Pressing one of the locations in this list will send the user to a “Details” screen.

Figure 2. App screenshots to show “List” screen

Diagram

Description automatically generated with medium confidence



The ”Details” screen, just like the original, provides a more informative description of the selected location. It’s a scrollable page with a large image, distance, title, description, events, and navigation buttons. Events are shown on the calendar and once pressed display an alert with event information for that day.

Figure 3. App screenshots to show “Details” screen

# design decisions

During this re-design I made sure to carefully decide my design principles from the beginning. The first thing I did was to use and understand the original app: finding what should be changed, what’s bad, what’s good, and then work from there. Initially I noticed that it had all the right information, but lacked good design and navigation. Drawer navigation made it feel like a larger app than it needed to be, and the use of separate screens for “Discover” and “What’s On” (with both map and list-views for each) made it an unnecessarily complicated implementation of an app that should be very straightforward.

In general, I made sure to let Donald A. Norman’s advice in “The Design Of Everyday Things” (Norman, D., 1988) to guide my design. Below is a list of some of the design principles how I implemented them.

* Visibility / Discoverability
  + Buttons are large and kept in the corners where sensible. This is in line with Fitt’s Law which states bigger targets are easier and quicker to reach, and the most quickly accessed targets on any screen are the four corners. (Fitts, P.M., 1954)
* Feedback
  + When changing screens via the bottom tab navigation, the currently selected screen will have a bold icon to indicate that’s the one currently selected. Shown in Figure 1 Vs. Figure 2
* Affordance
  + Great examples of affordance come from my use of the filter, search and back buttons. The icons of these buttons are the standard icon representation that is typical to a mobile app, and the icons are very relevant to the task they perform.

More specifically, the use of drawer navigation is typically used in apps designed to keep you engaged for long periods of time. This is known as the product posture. Author Alan Cooper speaks about this in his book “About Face: The Essentials of Interaction Design”. It refers to “A behavioral stance that helps to define the context of user interaction with a product, both as a whole or through individual features.” (Cooper, A., 2014). Apps with a sovereign product posture have the goal of monopolizing user attention, so will use components like a drawer navigation. Think of apps like Google Mail, or word processing apps, or even Facebook. Whereas a transient posture would typically present a single function and users would exit the transient workflow once they’ve accomplished the goal they set out to complete. Shifting into the use of bottom tab navigation made the app more simplistic with fewer interactions, therefore easier and quicker to use, thus making the app more transient which is very relevant as the app does only have a single purpose: to locate and provide information about the National Trust locations. The use of bottom-tab navigation also means the buttons will be placed at the bottom of the screen. This is great as the bottom of the screen tends to be within thumb’s reach of a user so it becomes almost effortless to use.

It's also important to mention I combined the “What’s On” and “Discover” sections from the original app, and made a single “Map” and “List”. According to Apple’s Human Interface Guidelines, one of their best practices states “Help people focus on primary tasks and content by limiting the number of onscreen controls while making secondary details and actions discoverable with minimal interaction.” (Apple, 2022). And this is exactly what I’ve done here. The primary task here is swapping to and from the list or map screens and the secondary activities are discoverable with minimal interaction, such as pressing a marker on the map. This principle helped me make navigation more intuitive without cluttering design and reducing amount of information displayed from the app.

Text

Description automatically generated

# IMPLEMENTATION DETAILS

In general, I made sure to apply my knowledge of coding standards. File structure is coherent, with good separation of screens, components, and constants. Variable names and docstrings are sensible. I also made sure to take advantage of features specific to React Native. I’ve implemented state variables on multiple occasions and used the use of hooks through Use Effect. I even installed third party libraries such as react-native-calendars to implement a custom components, and

Figure 4. Shows old implementation of fetching data

made my own.

A screenshot of a phone

Description automatically generated with medium confidenceOne big change I made midway through the development was my app fetching the json data with an HTTP request. Unfortunately, the API endpoint was taken down so as an alternative I now read the data locally. This is a fine alternative for now but if this prototype were to become official it would need to be able to pull live data somehow.

Figure 5. Shows the design of the old callout

Another issue I came across was not being able to place an image on my custom callouts for the map markers. Instead, I opted to omit the image as a workaround.

# references

Norman, D., 1988. *The Design Of Everyday Things* [online]. US: Basic Books.

Fitts, P.M., 1954. *The information capacity of the human motor system in controlling the amplitude of movement* [online].

Cooper, A., 2014. *About Face: The Essentials Of Interaction Design* [online]. 4th Edition. Wiley

Apple, 2022. *Designing for iOS* [online]. Available from: https://developer.apple.com/design/human-interface-guidelines/platforms/designing-for-ios [Accessed 17th November 2019].