Assessment 2: Ubiquitous Computing – Showcase Project

# PROTOTYPE Overview

My project was to design and implement an application that could easily look at the current data of the pandemic. The application will use a minimalist design, a series of APIs and a database to obtain information. It will also use the GPS sensor on the user’s device.

# design decisions

To conform with my idea of minimalist design, I chose to go with a simple navigation with three tabs at the bottom of the application. These tabs are static so they will be showed at all times. The reasoning for them being at the bottom is because it promotes reachability therefore on devices, such as mobiles, it could be reached one handed with ease. Each tab has text and a clear icon to represent their identity. Sharp et al (2019) agrees that users should have shortcuts and I have done this with the three-tab navigation system. To follow up on minimalist design, I have not included a homepage. This is because it is not necessary and all information that’s required will be included in the respective tab. Norman (2013) details that limitless possibilities often leave the user confused which is why my design decisions tend to be minimalistic.

I wanted to implement an interactive map for the user. My goal was to make it easy for the user to receive the latest coronavirus information in several different ways whether it was searching for a location or finding a location on the map themselves. I did it this way because with the use of the GPS, the device used will be located on the map then all around you is provided data regarding the pandemic so overall this provides options to how the user can easily find the information they seek. Norman (2013) suggests that when something is out of sight, it’s difficult to know about and use. This backs my design decision to make the map almost the full height of the display which enables good visibility.

I opted to go for a dedicated search tab. This is because searching is the quickest and easiest way to get information. The fact that it has its own tab means that it’s easy to identify from loading up the app and the search functionality enables users to quickly filter the location they want to see. Norman (2013) discusses the principle of making it clear to the user what action has been taken. I have achieved this in several ways. My search tab displays the locations that are interactive, and they give a clear response by navigating to that area on the map and displaying the coronavirus data.

As my application is about providing information, one of the tabs shows key details about the pandemic. All this information comes from the API, so it is updated when necessary. The idea here to keep it simple but one alternative I did explore was to render the mobile website but decided against that as it seriously affected the performance of the application overall.

# IMPLEMENTATION DETAILS

A cell phone with a map on the screen

Description automatically generated with low confidenceAs previously stated, I wanted to keep a minimalist design. So, to fit all of my features in I would need a minimum of three tabs. I used React Native Navigation to do so and chose the tabs concept as seen in Figure 1.

Figure 1 also shows the launch screen. On launch, you are taken to the map where there is an overlay that tells the user that the GPS and API’s aren’t ready. This is good for feedback because the map will not be ready to interact with until these two features are implemented properly. Sharp et al (2019) illustrates that you should offer informative feedback, so I have done this by telling the user what it currently happening in the application.

One key React Native feature I have used here is the Map View component. This is responsible for loading the map, making it interactive and displaying the area circles. Another package I installed was the native location API. This allowed me to use the devices location at ease. I chose to use this because it would be good to use with the map where an additional marker is plotted. I wanted to include animations in my approach. I created an overlay for the map to display information about the locations. For example, when you pick a location from search or click a circle you will animate to that region and A picture containing text, electronics, monitor, display

Description automatically generatedview will display the correct data with a slick animation. To add the animations, I had to install a package because this allowed me to do them easily.

Figure 1 - Launch screen showing navigation design and APIs working

Figure 2 - Data has been pulled from the database and the NHS API is in action

Data storage was required for what I wanted to achieve. The decision I made was the implementation of Google Firestore. This is a NoSQL document database that lets you easily store and query data for my application. I used this to store and retrieve the country’s location information such as latitude, longitudes and cities. To set this up I had to go to the web portal and create a new project. Once complete, I could get the configurations to put into my project.

Instead of using hardcoded data as seen in assignment one, I wanted to get real data that is updated without me having to do anything. A decision I made was to use the official NHS API that would get me all the information I needed to provide a good user experience. With each interaction on the map or search views, a components state would be updated using the API.

# Graphical user interface, text, application Description automatically generated

Figure 3 - Firestore console showing storage of documents (locations)

# References

Norman, D 2013, The Design of Everyday Things: Revised and Expanded Edition, Basic Books, Boulder. Available from: ProQuest Ebook Central. [28 January 2021].

Sharp, H, Preece, J, & Rogers, Y 2019, Interaction Design: Beyond Human-Computer Interaction, John Wiley & Sons, Incorporated, Newark. Available from: ProQuest Ebook Central. [28 January 2021].

Firebase. 2021. *Cloud Firestore  |  Firebase*. [online] Available at: https://firebase.google.com/docs/firestore [Accessed 29 January 2021].

Coronavirus.data.gov.uk. 2021. *Official UK Coronavirus Dashboard*. [online] Available at: https://coronavirus.data.gov.uk/details/developers-guide [Accessed 29 January 2021].

Reactnative.dev. 2021. *Introduction · React Native*. [online] Available at: https://reactnative.dev/docs/getting-started [Accessed 29 January 2021].

GitHub. 2021. *react-native-maps/react-native-maps*. [online] Available at: https://github.com/react-native-maps/react-native-maps [Accessed 29 January 2021].

Reactnavigation.org. 2020. React Navigation. [online] Available at: https://reactnavigation.org/docs/material-bottom-tab-navigator/ [Accessed 11 November 2020].

GitHub. 2020. Oblador/React-Native-Animatable. [online] Available at: https://github.com/oblador/react-native-animatable [Accessed 11 November 2020].