# **Implementation (2,500 words)**

With the previous chapter discussing how the app should be designed, this chapter focuses on how the author implemented this meaningful design.

**Tools Used**

First and foremost, React Native was the development language of choice for the app. Additional tools such as XCode and Android Studio, the Integrated Development Environments (IDEs) for iOS and Android respectively, were needed to create emulators to test the app on these platforms. Additionally, GitHub was the tool used for version control to ensure any progress was not lost and that had the author ran into any unforeseen circumstances where there was some issue with the app, he could simply continue from a previously working version of the code.

**React Native**

React Native was the development language of choice. Chose this over Swift and Java/Kotlin because React Native is cross-platform whereas Swift and Java/Kotlin are native, only working on either iOS or Android respectively. Flutter is an alternative cross-platform development language which was considered, but ultimately the author favoured implementing through React Native due to extensive online documentation and transferable skills whereby React Native is very similar to React, which is a development language used for building websites.

**XCode**

To-do

**Android Studio**

To-do

**GitHub**

To-do

Attach screenshot of GitHub

**Architecture**

* React Native – Front-end
* Firebase – back-end
* Figma – prototyping
* VSCode – Integrated Development Environment (IDE)

Add a diagram here and explain it in above paragraphs

## **User Interface (Front End)**

Insert screenshots below for each screen on a newer iPhone emulator.

**Login**

Graphical user interface, text, application, chat or text message

Description automatically generated

**Homepage**

2 screenshots here

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Similar items are grouped together on the homepage, where food logs appear together and transport logs appear together, making it easier for users to find and explore logs, reducing complexity for users.

**Log Emission Button**

A picture containing text, first-aid kit, clipart

Description automatically generated

The log emission button is easily accessible from anywhere on the home screen.

Graphical user interface

Description automatically generated**Log Food Emission**

Clicking save will lead to the confirmation alert illustrated below.

**Log Transport Emission**

Graphical user interface, text

Description automatically generated

Clicking save will lead to the confirmation alert illustrated below.

**View Individual Leaderboard**

Graphical user interface, text, application

Description automatically generated

The logged in user’s score is highlighted in green to speed up and reduce the complexity of locating their position in the leaderboard.

Graphical user interface, text, application

Description automatically generated**View Team Leaderboard**

Again, the logged in user’s team’s score is highlighted in green to speed up and reduce the complexity of locating their position in the leaderboard. If a user is constantly winning or losing in the individual leaderboard, their teammate’s (only one teammate) score may alter the balance, providing a more competitive landscape to compete in.

**View Individual History of Scores**

A picture containing graphical user interface

Description automatically generated

This screen sustains motivation, whereby users who are either winning or losing all of the time, can find competition against themselves, targeting the competence aspect of self-determination theory where they can strive to master their performance.

**Navigation Bar to Speed Up Frequent Tasks**

Graphical user interface, text, application, chat or text message

Description automatically generated

The navigation bar means these frequents tasks are only 1 click away for users, increasing speed and ease of use.

Graphical user interface, text, application, chat or text message

Description automatically generated**Confirmation Button to Reduce Likelihood of Errors**

The field of human computer interaction emphasises the importance of reducing the likelihood of errors as opposed to dealing with realised errors.

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, application

Description automatically generated**Reducing Complexity Through Variety of Units of Measurement**

Graphical user interface, text, application, chat or text message

Description automatically generated

Users who are used to different units of measurement or who do not know the size of their car are catered for, thus reducing complexity and cognitive load placed on users.

## **Database (Back End)**

**Storing authenticated user accounts on Firebase**

Table

Description automatically generated

Design of database was to follow the rules of avoiding redundancy, and following the noSQL golden rule of make collections large and documents small, whereby instead of storing, for example, a user’s emission logs as an array inside of a user document, the emission logs were stored as a separate collection with a reference link to the corresponding user. This is because the number of emission logs per user is infinite, so with each new log, the user document would have kept expanding, eventually exceeding the 2MB capacity limit imposed by Firebase. Instead, by storing emission logs as a separate collection, this error was avoided.

## **Implementation Issues (May have these…)**