



Pay As You Drive

Design Manual

Rhyder Quinlan
C00223030

Abstract

The purpose of the Pay As You Drive project is to develop a mobile application for the Android & iOS platforms that will, first and foremost, track driving habits of a user whilst travelling on a journey. Driving habits are recorded and used for motor insurance purposes, whereby a user pays a monthly insurance price based solely on how much and how well/poorly they drive. The application allows the user to view their past journey, control their profile details and vehicles through the app, and finally create and monitor insurance claims within the application.

Pay As You Drive is designed with a Peer 2 Peer business approach, whereby users are nodes in a network. Each month the nodes are billed for their driving, and this money is gathered into one central money pool. In the event of an accident, fire or theft to their motor vehicle an insurance claim can be made, and if approved, a transfer takes place from the money pool to the users wallet.

Table Of Contents

Abstract	1
Table of Figures	3
Section 1 - Introduction	4
Section 2 - Technology Toolchain	5
2.1 - React Native	5
2.2 - Firebase	6
2.3 - Cloud Storage	7
Section 3 - Database Schema	8
Section 4 - Class Diagram	9
Section 4 - UI Design Prototypes	10
4.1 - Colour Palette	10
4.2 - Screen Prototypes	11
Section 5 - Bibliography	15

Table of Figures

Fig. 1 Colour Palette	9
Fig. 2 Welcome Screen prototype	10
Fig. 3 Sign in screen prototype	11
Fig. 4 Sign up screen prototype	11
Fig. 5 Home screen prototype	12
Fig. 6 Journey screen prototype	12
Fig. 7 Summary screen prototype	13
Fig. 8 View past journeys screen prototype	13

Section 1 - Introduction

The following document will describe the design of the Pay As You Drive project. This document comes after the research manual and functional specification are defined so that there can be a better understanding of the design requirements needed to meet the functional specification. The research manual covers possible technologies, as well as UI/UX design inspirations from existing/similar applications. This Design manual uses this information to now lay out exactly what technologies this project will be developed with, as well as 'prototypes' of potential screen designs, although this will potentially differ greatly once development and testing commences.

The following section covers the technology toolchain explaining where the chosen technologies fit into the project, this comes after research into potential technologies to use, and so discusses the benefits and suitability of using the technology. Firstly the section will cover the use of react native and potential react native libraries that will be used, weighing up why it has been chosen and how this choice shapes the development environment.

Section 3 covers the database schema of the firebase realtime database. This is laid out in a JSON format to suit the firebase layout.

The next section covers screen prototypes. The screens show the main functionality and features that will be included. The designs are likely to change as development viability shapes the final product more and more.

Section 2 - Technology Toolchain

2.1 - React Native

React Native is a Javascript based programming language maintained by Facebook since 2015. The biggest benefit using React Native for this project will be that it allows cross-platform development, which in the case of Pay As You Drive would be for iOS and Android. What this means is that development would be written with one code base, and when the final product is submitted it would be compatible for publishing on both Android and iOS simultaneously.

“React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces.” (reactnative, 2020)

This project will use the development environment recommended by React Native called Expo. Expo is an open-source project that allows developers to build apps in a very fluid dev environment, where changes and iterations made to code can be instantly seen on emulators and via usb debugging. Expo can also manage libraries, structure the project and publish with ease. This project will also use Node.js and NPM to incorporate any libraries used within the project code.

“With Expo tools, services, and React, you can build, deploy, and quickly iterate on native Android, iOS, and web apps from the same JavaScript codebase.” (expo.io, 2020)

This project will also make use of the very active community backing React Native, there are thousands of NPM libraries built by the community, and more than a few that would be beneficial to this project. Below are some libraries that will be used, but with development more libraries will be adopted:

dotenv - This project will use this library for security reasons. Dotenv allows developers to keep sensitive information such as API keys as variables in a .env file that is not exposed to the public repo.

gesture-handler - This allows React Native to display scrollable views for dynamic lists etc.

geocoding - This is a google maintained library that will be used for geocoding the users location into a human readable format for aesthetic purposes.

geolocation - As described in the research manual, this library watches the position of the user and returns a position array in the event of location changes. This library will be called on to track location data such as distance, heading, longitude and latitude.

accelerometer - the accelerometer and gyroscope libraries are a way to call on the readings made by the inbuilt accelerometer and gyroscope within every phone. This will be used to measure acceleration and aggressive driving behaviour for the project.

2.2 - Firebase

Firebase is a very powerful software that is best described as Backend as a Software (BaaS). Originally developed by Firebase in 2011 and acquired in 2014 by Google. The aim of Firebase is to supply all the Backend tools needed to quickly and securely develop high-quality apps.

For this project it will specifically use Firebase's realtime database functionality as the database solution of choice. React Native communicates with the realtime database by using the firebase npm library.

2.3 - Cloud Storage

Google's Cloud Storage allows the uploading and downloading of images in a 'storage bucket'. The library for React Native works very similarly to Firebase, and will be used when a user needs to upload an image related to a claim.

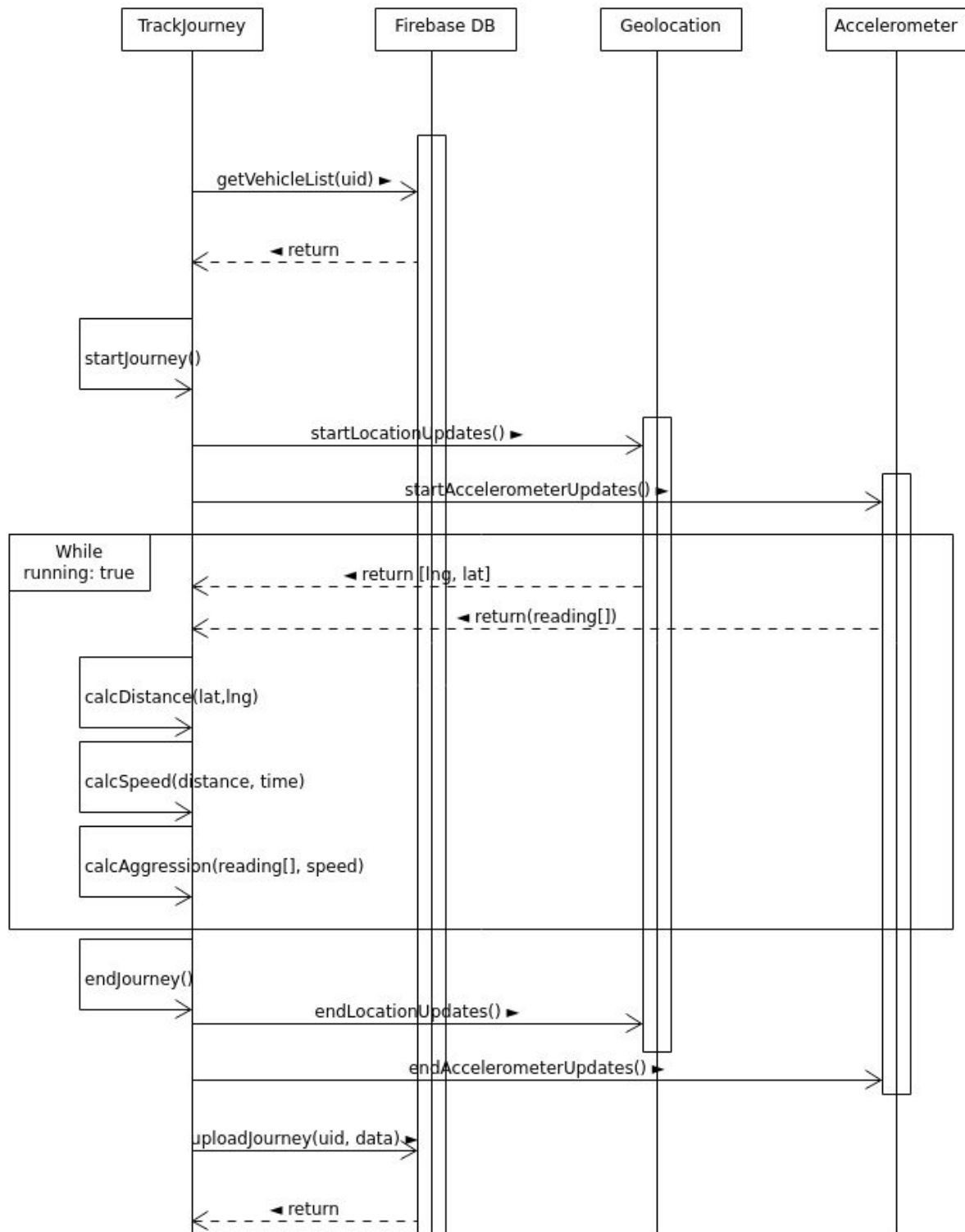
Section 3 - Database Schema

As Firebase uses a JSON format, this diagram will use a similar format to best show the database layout.

Pay-as-you-drive

- **Algorithm Variables**
 - Distance Multiplier
 - Class Additional
 - Licence Additional
 - ...
- **Users [**
 - <\$UID>**
 - Email
 - First name
 - Last name
 - Licence
 - Journeys [
 - <Journey ID>**
 - Address
 - Month
 - Cost
 - Date
 - Distance
 - Duration
 - Vehicle_Key
 - Vehicles [
 - <Vehicle ID>**
 - Name
 - Type
 - Year
 - Billing_History [
 - \$Month_Year**
 - Amount

Section 4 - Track Journey Sequence Diagram



Section 4 - UI Design Prototypes

4.1 - Colour Palette

Below is the chosen colour palette for the mobile application and its various elements.

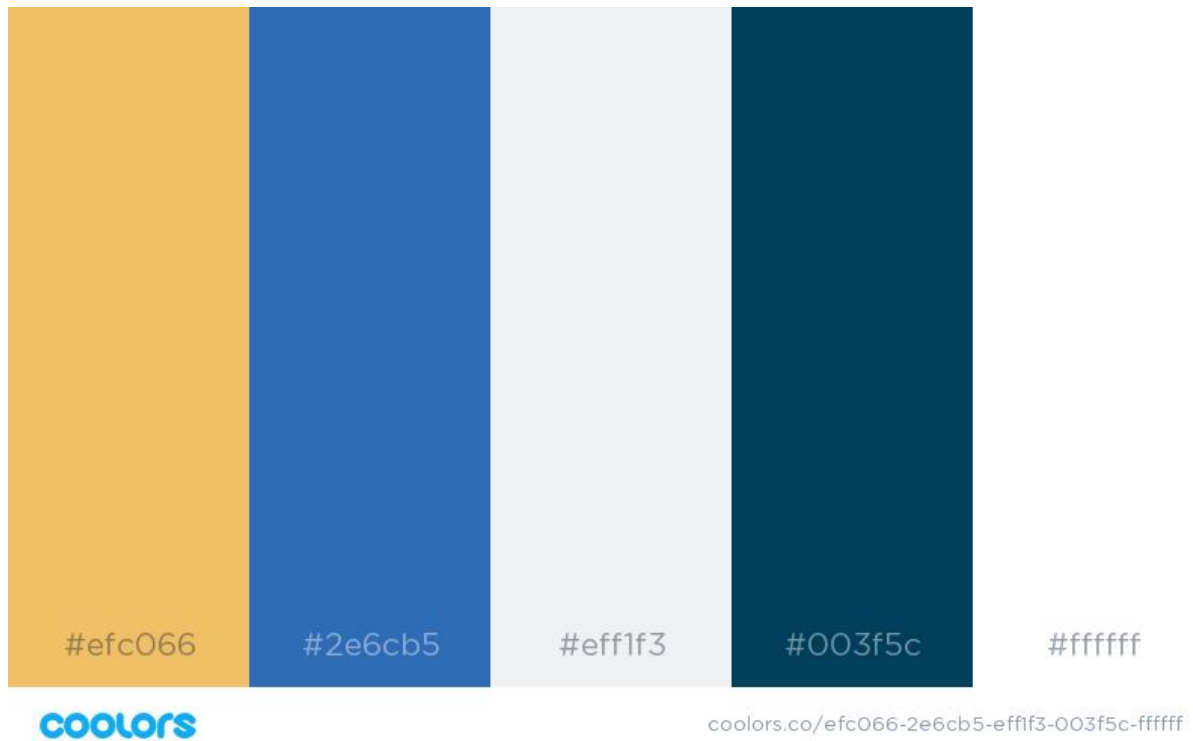
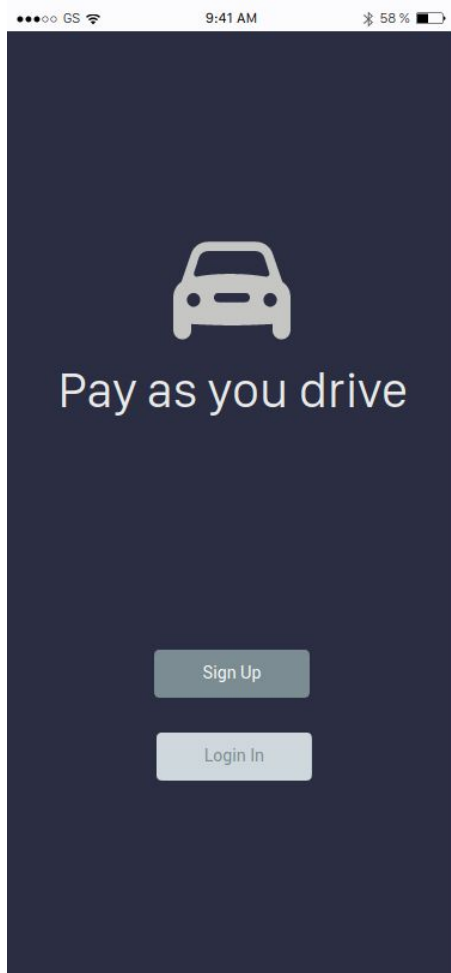


Fig.1 Colour Palette
[coolors.co colour generator]

4.2 - Screen Prototypes

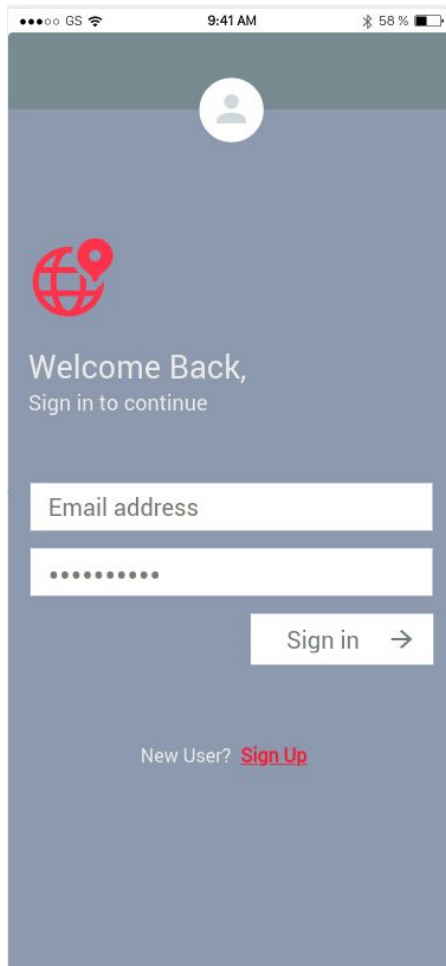
This section shows screen prototypes that will be used as blueprints for the project, the final product may vary. The colours used on the screens are only placeholders and will change throughout development.



Welcome Screen

This will be the opening screen for the application displaying a logo and title.

Fig.2 Welcome screen prototype



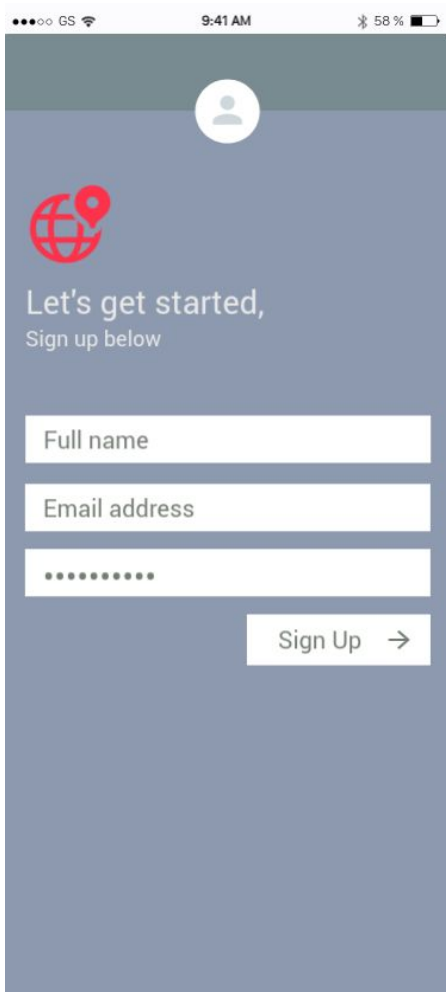
Sign In Screen

The Sign In screen will be the screen the user will see after the Welcome Screen. This is the first screen the user can interact with.

The user must enter their Email Address and Password in order to Sign In. The mobile app will make a comparison between the inputted information and the backend database to validate if authorisation is successful. If the authorisation is invalid then a message on screen will alert the user and another attempt can be made.

If the user is a new user to the application and has not made an account then the 'Sign Up' button is clicked and the user is directed to the Sign Up screen.

Fig.3 Sign in screen prototype



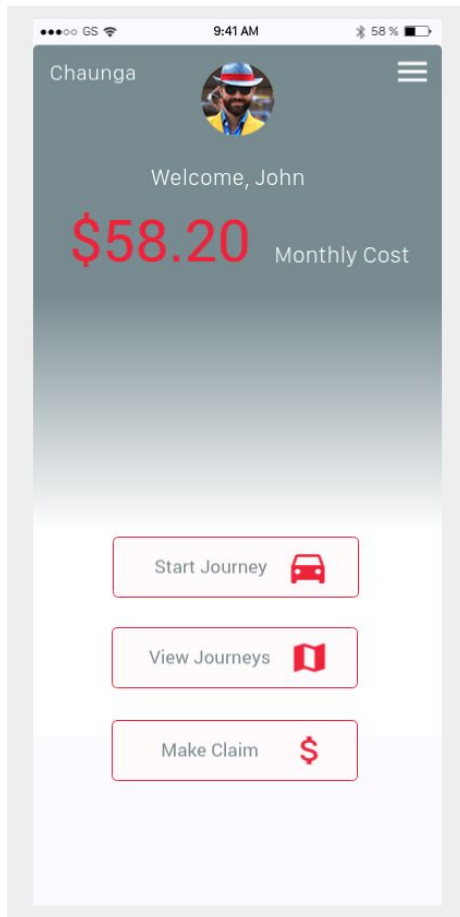
Sign Up Screen

The Sign Up screen is for a new user to create an account to be added to the database.

The user is prompted to enter their Full name, Email address and Password.

There will be more requirements needed for motor insurance purposes, and will make the final sign up screen differ.

Fig.4 Sign up screen prototype



Home Screen

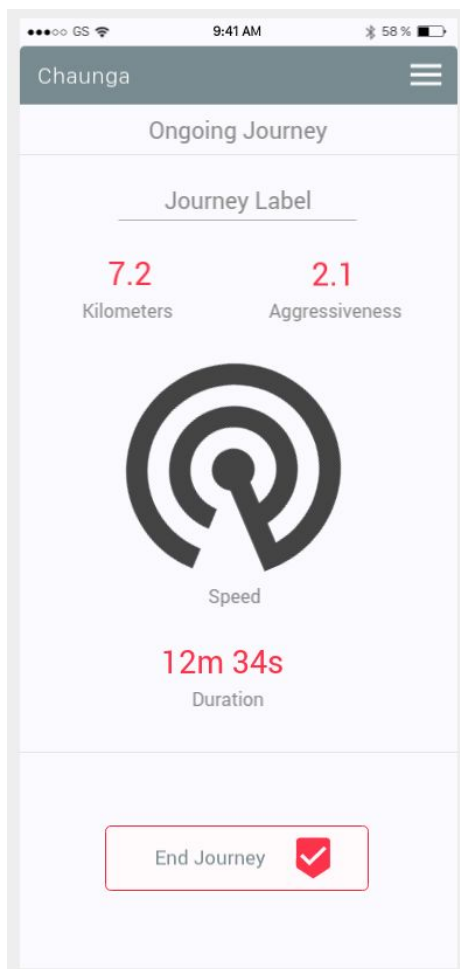
The Home Screen is the main screen the user will interact with.

The user can view their current monthly cost they will be billed at the end of the month.

The user can also navigate from the Home Screen to 3 use case:

- **Start Journey**
- **View Journeys**
- **Make Claim**

Fig.5 Home screen prototype



Journey Screen

The User will be directed to this screen from the Home Screen after selecting 'Start Journey'.

The user can edit the name of the journey by clicking 'Journey Label'.

This screen shows:

- Mileage travelled
- Aggressiveness score
- Duration of Journey
- Current Speed

Selecting 'End Journey' will direct the user to the Journey Summary Screen.

Fig.6 Journey screen prototype

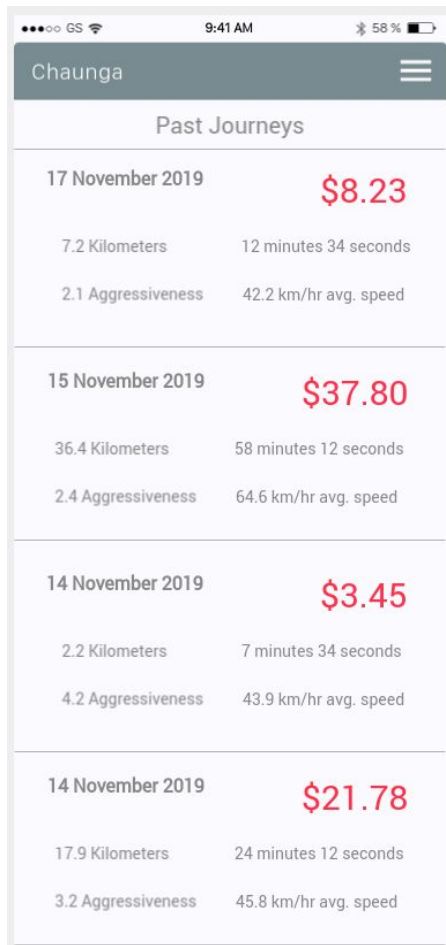


Journey Summary Screen

The Journey Summary screen is a break down of the details of the ended journey for the user to view.

A calculation of the total cost of the journey is displayed near the bottom.

Fig.7 Summary screen prototype



View Past Journeys

This screen shows summaries of past journeys in a list format, highlighting the cost of each journey and the date it was taken.

Fig.8 View Past Journeys screen prototype

Section 5 - Bibliography

reactnative, 2020. *React Native · A Framework For Building Native Apps Using React*. [online] Reactnative.dev. Available at: <<https://reactnative.dev/>> [Accessed 13 April 2020].

expo.io, 2020. [online] Expo.io. Available at: <<https://expo.io/>> [Accessed 13 April 2020].