# Introduction

## Project Overview

This report outlines the approach that was taken when developing a mobile application which's design is to assist users to keep track of their mood over the month of January. The idea being that once a user downloads, installs and creates an account through the application they can then leave daily notes regarding their day .as well as rate them on a scale from “horrible” to “great”. By the end of each month they will have a graphic that is uniquely generated from the data that they have been inputting throughout the month.

## Aims and Objectives

With this application it is being targeted the identification of bad habits as well as the generation of a graphical form to display the user’s moods over the period of a month, for upcoming months, it is expected to continue the maintenance of the app by releasing updates with new monthly art to represent every new month.

However, for the aims and objectives of this project it was decided to stick with what is currently attainable and achievable by the given deadline, which was to deliver a working system with at least registration and login, graphic monthly art for the current month and finally the ability to enter data and dynamically change the graphic art based on the data entered by the user.

Beyond the deadline, some of the future aims of this project include the development of the platform and integration of subscription services to fund a specialist team that could provide feedback to the users where a given specialist would target each user individually and through in app messages or campaign emails would then provide them easily with digestible feedback that would aim to help improve their life.

## Justification of selection of Particular Technologies or Techniques

By looking at the job market it can be certainly observed that JavaScript is a widely sought after skill as a programming language that a software engineer should know. This is easily justifiable as its versatility allows for its use on web, console, desktop and mobile applications. Taking this into consideration, it seemed most useful to pick it as the main programming language for developing the mobile app as this would make learning the language, a fair amount more purposeful and enforcing.

However, JavaScript is not natively supported by Android or IOS smartphones, it so therefore requires a framework that enables this support, and for that purpose it was chosen to use React Native, which is written and maintained as an open sourced project by Facebook. The main advantages of using React Native are that, as a JavaScript framework, it facilitates the use of the same code for both Android and IOS, especially where it comes to graphical user interfaces, networking and a few other functions that do not require native support, in which case these may have to be written individually for both Android and/or IOS if the React Native framework does not support inter-compatibility right out of the gate, an example of this it could be to access the device’s storage media, Camera or GPS.

While these few functionalities that have to be written individually based on the operating system, using Java or Switft (i.e.) the vast majority of the code is shared, which makes the application easier to maintain and facilitates the scalability.

For the storage and authentication API solutions, it was chosen the use of Firebase, which is a platform developed by Google for creating web and mobile applications, one of its storage solutions is a real time NoSQL database, which is a service offered freely with a very reasonable limit, beyond this limit the costs are also fairly reasonable as they scale proportionally to the usage of the service, and on that note, since it is a service provided by Google, its scalability can accommodate just about any amount of unpredictable growth. This makes firebase the ideal candidate for start-up applications as opposed, for example, to self-hosting of services where it is required an advance investment in resources.

NoSQL databases are extremely easy to manage, maintain and scale as new attributes can be added at any time without affecting the existing data as opposed to SQL databases where in a lot of cases it would be necessary to completely modify the existing data model. This aspect is particularly important for the given project as during the inexperienced ongoing development of the app, it may be the case that new storage requirements might pop up which in a SQL database would translate into significant amounts of work and time being spent creating and optimizing the new structure of the model instead of working on more important aspects such as app features that would more likely impact the success of the app.

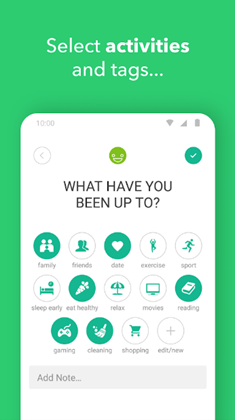
Lastly, the technology choices made were highly biased towards new high in-demand skills sought by employers, new open-sourced technologies that are highly regarded by the software engineering industry and towards delivering a project that in the future could easily also be adapted for iOS systems. This meant that, it was overlooked the use of old proprietary technologies such as Adobe PhoneGap as suggested by the assignment brief. Valid alternatives for the mobile framework included: Flutter, Ionic and NativeScript.

# Secondary Market Research

In the following paragraph we are going to look at some of the trending applications (according to google) that resemble with ours and can be found on the Google Play store. Althought there are a few similarities in between them, the end goal completely differs, as well as the way they are monetized. The majority of them ask for paid subscriptions or have to be bought in order to be used, whilst for ours we don’t. Insteal of doing that we provided with a donation functionality, highlighting the fact that this is a volunteer work and all the money obtained from the donations will futher be invested in the application itself.

## Daylio

Daylio is a mood/habit tracker with a neat, modern-looking design and a ton of features. The app has a mood tracker which simply asks the user to pick one of the five smiley faces representing moods. The mood data can then be transferred into a graphical representation, from simple line graphs to colored dots at the end of each month. The other important feature is the habit tracker, where the users pick from a pre-made selection of habits or tasks. Both the mood tracker and the habit tracker can then evaluate the data and display statistics, for example how many times did the user go to sleep early in that given month and how it correlates with their moods. A diary and a journal are part of this app, however these functions are often found on pre-installed software on any mobile device and so are not considered a plus.



The monetizing strategy for this app is free to download and buy to customize. There seems to be a large variety of customization, however, all of it is in line with the static modernist design of the app, so the user only customizes icons, colors and can add their own activities or habits to keep track of. The functionality of this app is its main selling point, but its design is rather unattractive, boring and unoriginal.

**Pros**: large variety of graphs and statistics, easy to use interface

**Cons**: general and impersonal design, very limited customization options, in app purchases

### Moodflow

This app is almost identical to the previous app in terms of functionality. The only noticeable difference is in the design of the app, which tends to use more vivid colors and defined shapes. Additionally, this app also markets itself as a “self-improvement” app, emphasizing the planning and organizational utility of the app.

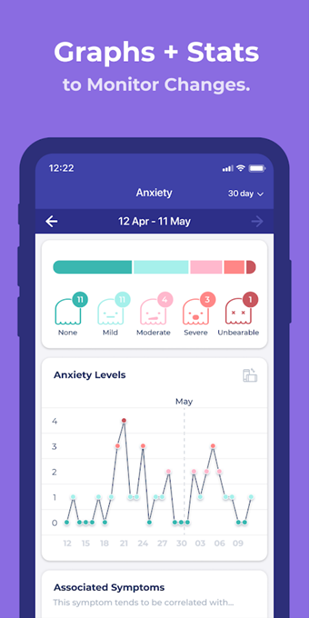
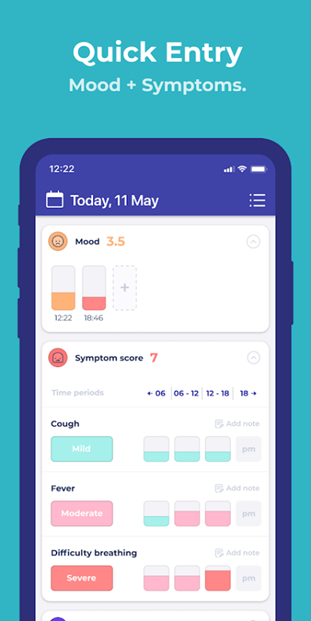


**Pros**: Customizable charts and graphs

**Cons**: the interface is not very user-friendly, the customization process is not well explained , a large amount of ads

### Bearable

The biggest difference and a selling point of Bearable is its connection to **health**. In Bearable, users can keep track not only of their moods and habits, but also of their symptoms if they are ill. This is done via a friendly-looking UI where the user can pick, for example, whether their fever is “moderate” or “severe”, and their current mood. This input is then transferred into statistics and the user can find out how impactful are their symptoms on their mood.



**Pros**: tracking symptoms, can gather data about sleep or step count from the device, connects the mood data with the symptoms data

**Cons**: design is static and tied to the name/brand, some features not that useful (reminders or alarms), generic-looking graphs

Project Constrains and Risks

Living in a technological era that facilitates human interactions, more than ever we have all those alternatives to communicate with the loved ones. The same technology that brought us together made it overly difficult for people to express their emotions. Demographically speaking over the past years there can be seen a huge spike in the cases of depression, anxiety and people that suffer of loneliness or generally have a negative mindset.  
This application comes as a collective effort, whereas people from completely different indistries decided to collaborate and take action on that regard. The whole ecosystem is based on volunteer work, where students that need work experience decided to join the team.

Our team is composed by artists who are in charge of the monthly themes for the application. Each month having a completely unique design. The user input is then processed by the psychology students and the application is maintained by the other voluteers. Other than gaining real world working experience that can be added up on their CV’s the people involved in the project get the benefit of working on something that can make a difference.

The main risk of the program is that the demand will be extremely high that the voluteers cannot deal with every single individual that opted to use the application. This may will result in less particular oriented advices, keeping the feedback as general as possible so it fits a wider variety of people. Clearly not a huge game changer, but when dealing with persons individually the healing process is way faster.

Scope

For the time being we decided to prioritize the application itself. Assurring that we have a fully fladged application on the marketstore which users can download and use. It consists of the login, register pages the main page and the sub main page where the user leaves notes about his day. Once we gather some substantial data the next step will be to look aftere volunteers who are willing to parse it and write a few words for each individual of how they could improve their lifestyle and make the best out of it. Currently the applciation uses the mountain theme, but in the near future we’re expecting to have huge improvements on the front end side.

Resources

* Software
* Hardware
* Graphics

Methodology

An agile approach will be used for this project. Based on the user reviews and the data we collect, constant improvements will be made to assure a great user experience. The main idea is to create a prototype of the application and launch as soon as possible. Acknowledging the fact that the users might demand certain futures we developed the application in a fashion that it permits continuous integration and assures scalability.

Stakeholders

University of Bedfordshire students

University lecturers

Students from other Universities/Campuses

Functional Requirements

Login

Register

Signout

Persisten authentification

Persisten storage

Rate your day page

Autopopulate the SVG with user data

Database connectivity

Form validation

Functionality to rate your day

Main page design

* Operational Requirements
* Usability Requirements
  + User-centric design
  + The design should demonstrate evidence of a good understanding of mobile app interface design issues – for example, a consistent design for each page, layout of content, use of colour schemes and images, navigational methods, usability when viewed at various device resolutions and various device dimensions.
  + Simple and short forms for entering details
  + Content can be added and updated easily and quickly
  + Clear and intuitive navigation
  + Complies with WW3 Web Accessibility Standards (WCAG)
  + Text easy to read and resizable  
    Language style should be appropriate with absence of grammar / spelling errors
  + There should be a clear layout which remains consistent throughout the site. Style, layout and content should be appropriate for the purpose of the app.

Identify target users

The application is addressed to the people that encounter difficulties coping with life. With the aid of our specialist we are hoping that as long as their willing to invest in themselves and follow along their advices significant improvements can occur.

Although they represent our main audience, the application can be used by anyone who wishes to keep track of their evolution over time. Our psychologists will only address to the individuals that have clear signs or anxiety, depression or a negative mindset.

Primary market research

come up with a questionaire

Personas, Scenarios and Uses Cases

App Prototype (sketches/screenshots, etc.)

//Use Balsamiq for this

//Annotate your prototype sketches/screenshots based on the design guidelines/principles, lessons learnt from the secondary market research and the preferences of your personas from the primary market research.

Mobile app development

Since the assignment brief was extremely permissive in terms of what technologies can be used and what the application should be about we started with a brainstorming session. After we had a clear idea of the application should be like we done some diligence on the technologies that can be used to create it. We analyzed each suggestion carefully and picked the one we consider to be the best to fit our needs.

The next step was to break down the application from a functional point of view so each member of the team can work on his tasks individually, without merging conflicts. The database structure was defined from the very beginning to avoid any inconveniences and assure that we’re all on the same page.

Once the tasks were assigned we defined a set of rules which cover:

- how each party should push his code

- the branching system

- when to merge to the main branch

- how to avoid overriding each others code

We heavily relied on github as a version control platform, having branches per developer minimizing the potential merge conflicts. Once a future was marked as completede we would merge to the main branch then pull on the individual branches. Assuring that we are all working on the same codebase.

At the very beginning we looked up all the node modules that are required in order to create the application, defined the folder structure and the environment the application should run on.

One of the biggest problems we encountered was due to the fact that initially we managed the node modules using npm, where the firebase module was outdated at that time. Making it impossible to retrieve data from the database. To overcome that we had to switch to yarn, a different javascript node module manager which had that library up to date. Sometimes the virtual machine wouldn’t update the application. This resulted in a lot of odd behaviours where the changes that were made wouldnt make effect. Initially we thought it was due to a malicious code block or something similar, but later we discovered that we had to restart the emulator to get the latest version of the application. It was extremely time demanding to write the application due to that. It would up to minutes sometimes to power up the emulator. Other than this there were not any noticeable issues.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

<https://firebase.google.com/docs/auth>

The firebase library manages the data layer and keeps the application on sync with the information from the database. It also supplies with a secure environment for user authentification and registration which saved us a lot of time, since we didnt had to code that functionality ourselves while instead we could focuss on the other parts of the system.

React wise we opted for functional components for a number of reasons, first being the simplicity of the code. Compared to class based components which can achieve the same with the double lines of code. The less code the better, since it will be easier to maintain, scale and test. Another reason why we choose to opt for functional components being that according to the React maintainers they offer a performance boost.

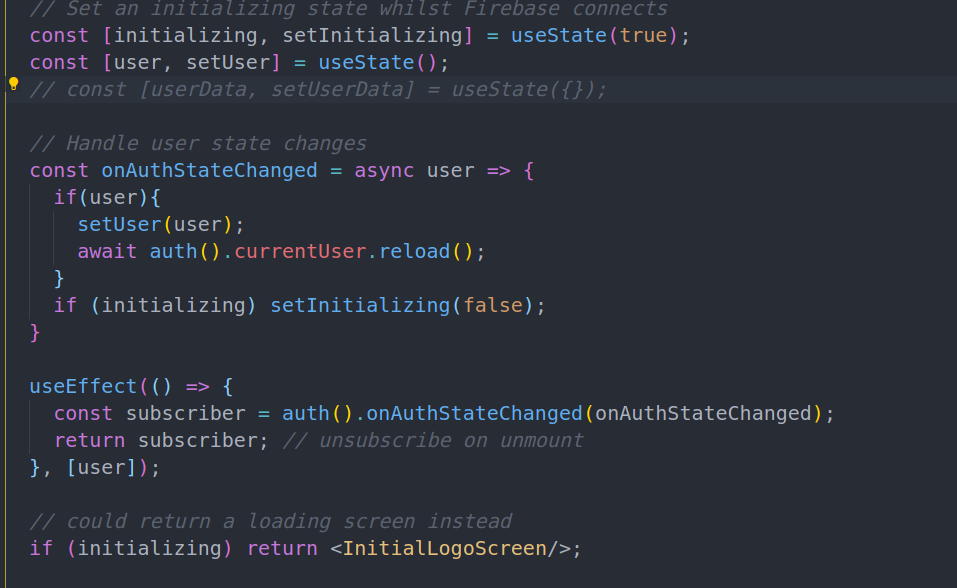
In the development process we havely relied on React Hooks.

React Hooks are functions that let us hook into the React state and lifecycle features from function components.

By this, we mean that hooks allow us to easily manipulate the state of our functional component without needing to convert them into class components.

Hooks don’t work inside classes(because they let you use React without classes). By using them, we can totally avoid using lifecycle methods, such as componentDidMount, componentDidUpdate, componentWillUnmount. Instead, we will use built-in hooks like useEffect .

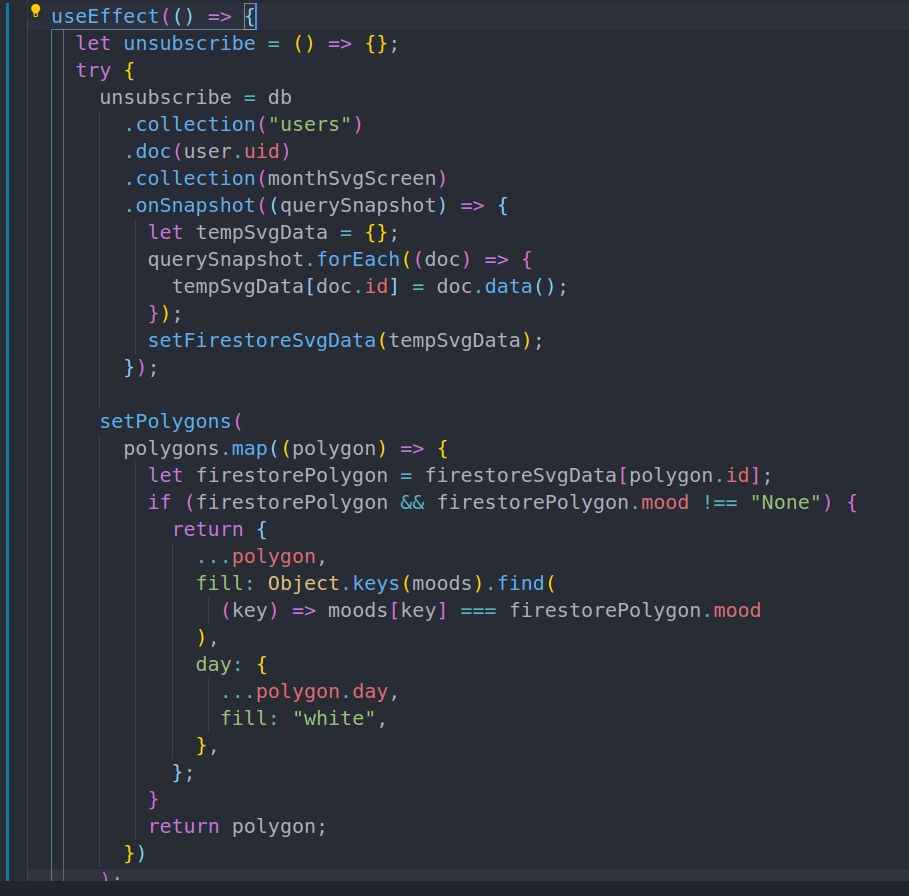
https://hackernoon.com/react-hooks-usestate-using-the-state-hook-89ec55b84f8c



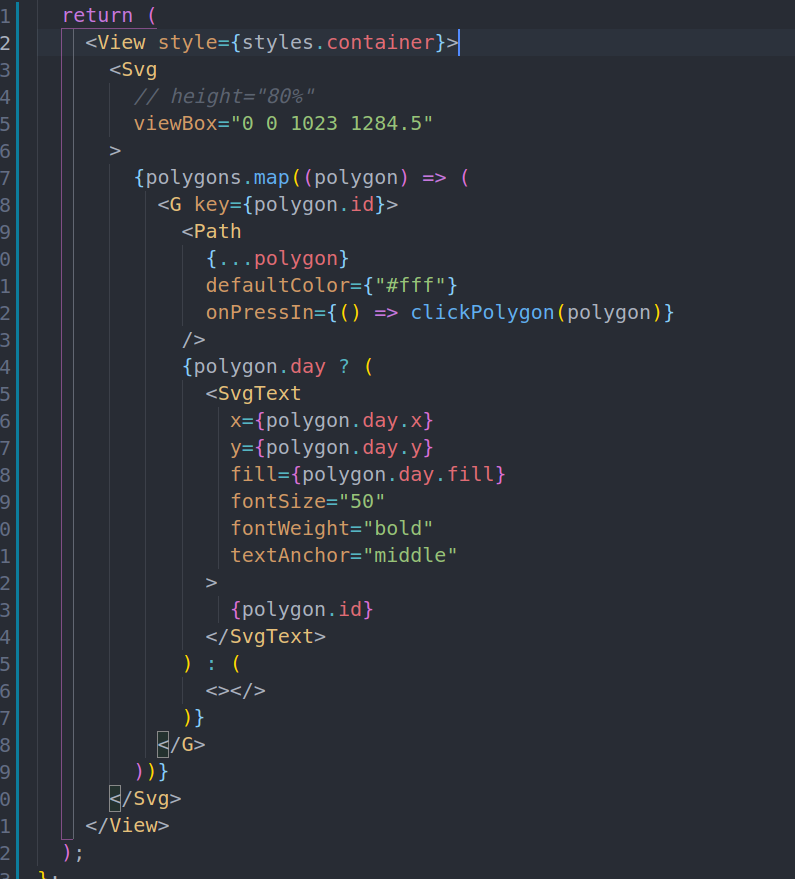
This code snippet attached above can be found in HomeScreen.js and verifies if the user is logged in or not. Based on that he will be prompted to the login/register screen or navigate to his personal page.



Above can be seen the data structure for each polygon rendered in the mood page. Where they have a unique id, the x, y coordonates which dictate where each should go in the view. The stroke, strokeLineCap and the other attributes are used for design purposes.



By default all polygons come in a white filling, but if the user left notes or a review we retrieve the data from the database and render the page accordingly. That’s achieved by using useEffect, a react hook.

To render the polygons we interate through each one of them (the polygons array) using the map function, passing down the props mentioned above.

Once the user click on them they will be prompted to a new page, where they get to write a few words about their day and leave a rating.