Essay

Word Locator

Overall, I really enjoyed this mobile project. I learnt a great deal from the app development process; the primary point learnt being that it is an iterative app development process between planning, design, and implementation.

Some of the early stages of planning were straightforward (due to them being the initial design decisions), eg. deciding on the programming tools(and development platform) to use was a case of reflecting from my experience of these tools(eg. React(+VS code) and Android Studio), listing the pros and cons of each, then prioritising the tools based on personal preference (for myself it was the ease of build and deployment; which consequently led me to select using React Native (and Expo to quickly get started on building a React Native app) as well as Firebase).

However, other early stages of planning like choosing the app design and features, this was iterative. Because, although I had an idea in mind, after comparing competitor apps and understanding its issues, as well as popular features, this unfortunately had to be refined and re-done.

During the initial storyboard sketch that I had drawn, I realised that I had been attempting to plan to implement one too many features, that would not have been feasible given the time period for completion. Therefore, needed to limit the number of features before implementation (however, I was unsure of which features to implement and which to not). I continued to complete the font board and business case of my app, but before long I understood that I would need to back-track and redo this, due to not knowing what features I would be implementing. When I reached the user stories and personas however, this provided me with the insight I needed to decide on the app' core features, because I was able to put myself in the seat of what the users of my app would need and expect (i.e. primarily a search feature to look up and display word definitions and similar words).

Of which, then I was able to re-do parts of my business case and font-board to reflect this. It was unfortunate that I had to do this, but was also inevitable and lucky, because this occurred before creating a new storyboard sketch and digital version in PowerPoint, which possibly could have saved me hours of time.

I was not sure how to approach creating a basic React Native mobile app, as I had no previous experience in doing so. Therefore, this prompted me to search for tutorials and come past Expo and React Navigation. Expo was easy to install, and incredibly useful as it allowed me to build and deploy a basic React Native app/project out of the box on either my android phone, or through an emulator, or web browser, for quick testing of various frontend UI and backend components.

React Navigation was similarly also easy to setup, installing the latest version to create a basic stack navigation between different mobile screens, using React Native functions (following the official documentation). Unfortunately, later, further into the project, I had issues implementing the search functionality of the app, due to discovering that when installing the latest version of React Navigation, I could only navigate to different screens through using functions. Of which up until this point, was not an issue, but unfortunately, I required to use a React Native class, due to complexity of the search functionality.

This incredibly frustrated me, as I spent hours searching online, trying alternative methods to implement the navigation using React Native classes. However, most solutions either resulted in

changing the versioning of the React Navigation or creating my own navigation template. Both solutions I was sceptical of carrying out because it might have been prone errors and would have resulted in myself having to heavily adjust existing implementation. I finally ended up changing my react navigation versioning (in the package.json file) and adjusting my implementation. However, I believe it would have saved me a lot of time before this in testing and knowing (whilst reading the official React Navigation documentation) that I could not use React Native classes for navigation. The React Navigation package did save me time in not having to implement my own navigator, but the official documentation was incredibly misleading. I would use it again; however next time would have needed to have tested React Navigation in React Native classes earlier on.

A few other minor issues I had encountered was also a case of either not understanding the scope, or syntax of installed packages. For example, in using the React Native Elements checkbox (for enabling 'word of the day' notifications), I thought that the checkbox state could be changed by implementing a simple method as to what I had previously used in React.js for a web application. However, understood after searching online, that this would not work, since the checkbox state could only be changed by using certain parameters (as outlined in the React Native Elements documentation).

Throughout this project, It was a case of trialling components and understanding from bugs (or errors as outlined from Expo debugging tools) what was the issue, reviewing the code to discover and correct the issue, and if this did not work, then against possible solutions online. Sometimes, for instance when I utilised the expo-av package to try and load and play audio pronunciations of different words, I had issues. Because, I followed the Expo documentation blindly. But, usually solved this by fiddling around within areas of my codebase to understand exactly what my code was trying to achieve.

In improving my understanding of the tools that I had been using, I became more comfortable in being able to solve bugs and issues as they arose. Which is why I would consider using them again in the future.

In summary, I am very happy with the outcome of my app. However, I do believe it needs further development, plus further additional features(eg. favourites), to be able to be taken to market, and be on a higher level than its competition. For example, I did not complete the backend (Firebase) implementation on in app-messaging/notifications for the 'word of the day', and I was also unable to optimise search to include the entire English dictionary, as currently the app only holds 500 words.