

ELDER-CARE

A Progressive Web Application (PWA) to Assist the Elderly

Student Name: Craig Lawlor

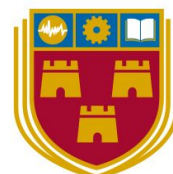
Student Number: C00184465

Supervisor: Hisain Elshaafi

Submission Date: 12/4/2019

Final Report

BSc (Hons) in Software Development



INSTITUTE *of*
TECHNOLOGY

CARLOW

Institiúid Teicneolaíochta Cheatharlach

Table of Contents

1. Abstract	3
2. Introduction	4
3. Project Description	5
3.1 Inspiration	5
3.2 Project Overview	5
3.3 Technologies	6
3.3.1 Frameworks	6
3.3.2 Languages	6
3.3.3 Tools	6
3.4 Application Screenshots	7
3.4.1 Complete Details Form	7
3.4.2 Elderly Person's Dashboard	8
3.4.3 Care Assistant Dashboard	9
3.4.4 Dashboard on Mobile	10
3.4.5 Emergency Contacts	11
3.4.6 Emergency Contacts on Mobile	12
3.4.7 Add Appointment/Event	13
3.4.8 Appointments & Events	14
3.4.9 Add Appointment/Event	15
3.4.10 Elderly Person's Diary	16
3.4.11 Elderly Person's Diary on Mobile	17
3.4.12 Group Diary	18
3.4.13 Group Diary on Mobile	19
3.4.14 Medication	20
3.4.15 Medication on Mobile	21
3.4.16 Update Details	22
4. Conformance to Specification & Design	23
4.1 Early Prototype	23
4.2 Changes made to Original Specification & Design	23
5. Description of Learning	24
5.1 Personal	24
5.2 Technical	24

6. Project Review	26
6.1 What went wrong	26
6.2 What went right	27
6.3 What I would do differently	27
7. Conclusion	28
8. Acknowledgements	28

1. Abstract

Digital technology can enhance our lives in many ways from keeping in contact to finding the information we want on demand. As technology advances it seems the elderly are being left behind. There are many contributing factors such as the fear of making mistakes or tough learning curves to modern applications. From the research carried out it is believed that the elderly, with the help of family and/or their care assistants, can too enhance their lives through the use of a web application.

The aim of this project is to help elderly people to stay connected and organised with the aid of a web application. Through the research it was found that a Progressive Web Application was an ideal way to deliver a smooth user experience across multiple devices. This will ensure that an elderly person who prefers to use a tablet or phone will have a similar experience to an elderly person who prefers to use a desktop or laptop. Multi device accessibility will enable more family members and/or care assistants to access and aid the elderly person with the application.

2. Introduction

The purpose of this document is to outline the process of development for the project which was a Progressive Web App to assist the elderly. In the sections to follow the document will give an overview of the project, the inspiration behind it, the proposed way to achieve the goal and the technologies used. This section will also include screenshots of the completed project.

The next section will outline the beginning of the coding phase of the project and what had to be changed from the original design. To follow this the document will describe the learning outcomes both from a personal and technological perspective. The final section will review the project as a whole outlining the things which went wrong, the things which went right and changes that would be made if starting again.

3. Project Description

3.1 Inspiration

The inspiration for this project came while searching for an application to help elderly people stay organised. There was a gap in the market for a good quality, multi-platform application which could be used by elderly people. The number of elderly people using technology and applications is always rising but to bridge the gap in technical capability amongst the elderly it was thought that a facility to allow family members and care assistants to assist the elderly person with the application via their own account would be beneficial.

3.2 Project Overview

To ensure that users could use the application on multiple platforms and even gain an app like experience it was decided at an early stage that the application would be a Progressive Web App. Progressive Web Apps give a similar feel no matter what device the user accesses them from and even allow the user to install on any device. By installing the Progressive Web App the user will have a similar experience to that of a native mobile or desktop application. Another benefit to developing a Progressive Web App is that the user can access certain features without an internet connection and receive push notifications.

It was felt that this approach would reach as many users as possible no matter their preferred device type with the benefit of only having to develop once. Also with the interconnectivity of care assistant users in mind, the care assistants would not have to use the same type of device as the elderly person. Once the type of application was decided upon the technologies for development had to be decided upon before development could begin. The final technologies used are outlined below.

3.3 Technologies

3.3.1 Frameworks



Angular 7 - Bootstrap 4

3.3.2 Languages



Html - CSS - Javascript - Typescript - JSON

3.3.3 Tools



Node JS - Firebase - Webstorm IDE - Github

3.4 Application Screenshots

3.4.1 Complete Details Form

After the user has signed up an account they are greeted with this form. The form must be completed before the user can proceed to use the application. If the user selects assistant under account type an additional field will appear where the user must enter the email address of the elderly person they wish to aid. Once they successfully connect with the primary account of the elderly user they will be able to access information and functionality relating to the elderly person. If the user opts for a primary account they can then get started as an elderly user.

The screenshot shows the 'COMPLETE DETAILS TO CONTINUE' form within the ELDER-CARE application. The form is centered on a light blue background. At the top, the title 'COMPLETE DETAILS TO CONTINUE' is displayed in a light blue font. Below the title, there are four input fields and a submit button. The first field is a dropdown menu labeled 'Account Type' with a 'Select...' option. The second and third fields are text boxes labeled 'First Name' and 'Last Name', both with placeholder text 'Enter first name' and 'Enter last name' respectively. The fourth field is a text box labeled 'Phone Number' with placeholder text 'Enter phone number'. A green 'Submit' button is located at the bottom center of the form. The application's navigation bar is visible at the top, showing 'ELDER-CARE' on the left and 'DASHBOARD', 'APPOINTMENTS & EVENTS', 'DIARY', 'MEDICATION', and 'MORE' on the right. The footer at the bottom shows the date 'Wednesday, April 17, 2019', the copyright 'ELDER-CARE © 2019', and a link for 'Emergency Contacts'.

Fig.1: Complete Details Form

3.4.2 Elderly Person’s Dashboard

When the elderly person has successfully completed the details form in Fig.1 above they are redirected to their dashboard. When they first reach their dashboard there will be no information in their Appointments & Events window or their To-Do List window. The following screenshot was taken after some test information was entered for a test elderly user. This is the main area of the application and the aim here is that the elderly person can get to as many places as possible as quick as possible.

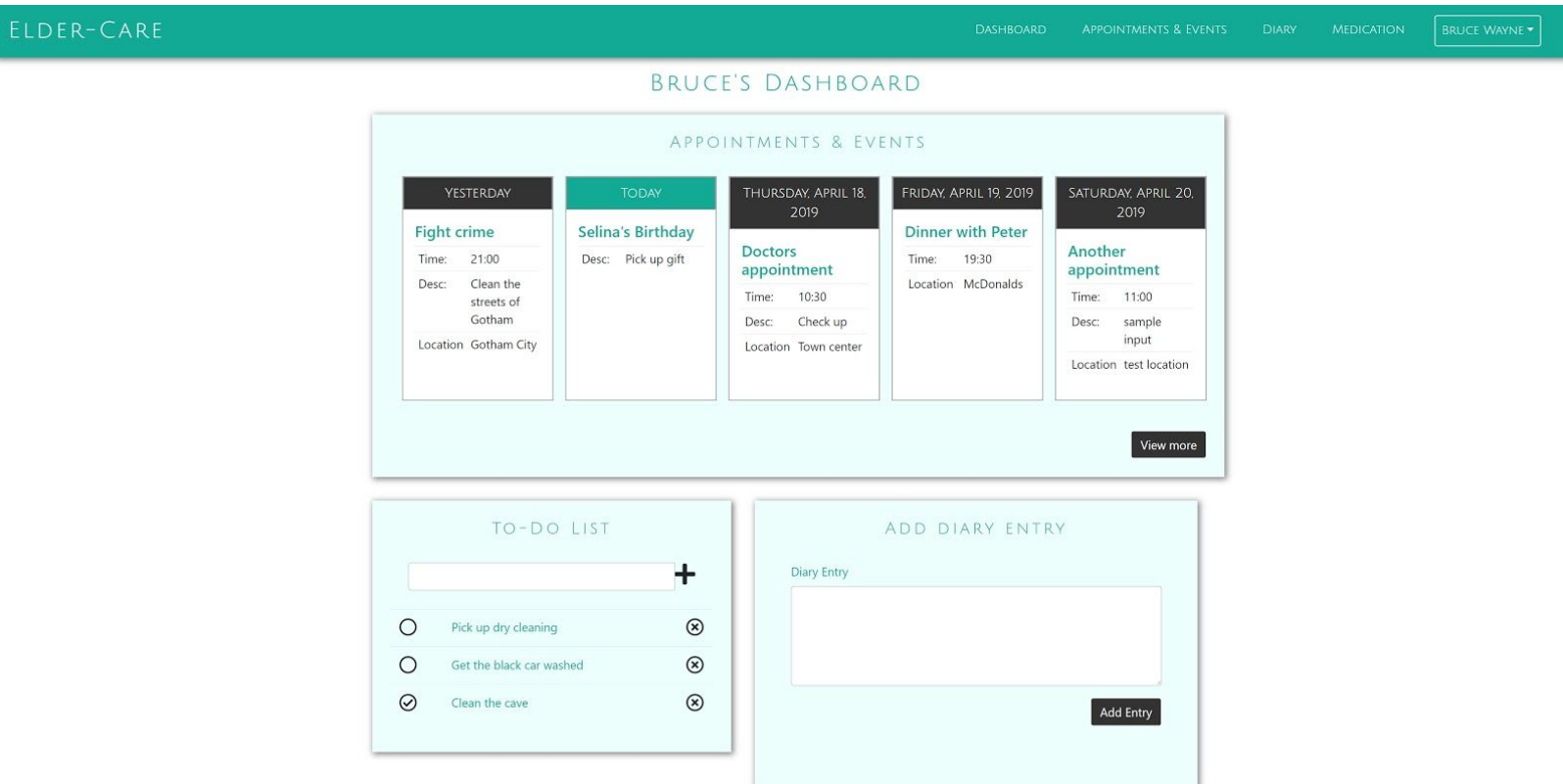


Fig.2: Elderly Dashboard

3.4.3 Care Assistant Dashboard

Similar to the elderly person's dashboard above the care assistant will reach their dashboard after successfully completing the details form. When the care assistant user is on their dashboard they can see the upcoming appointments and events for the elderly person they are associated with. They can also log an entry in the group diary from this screen.

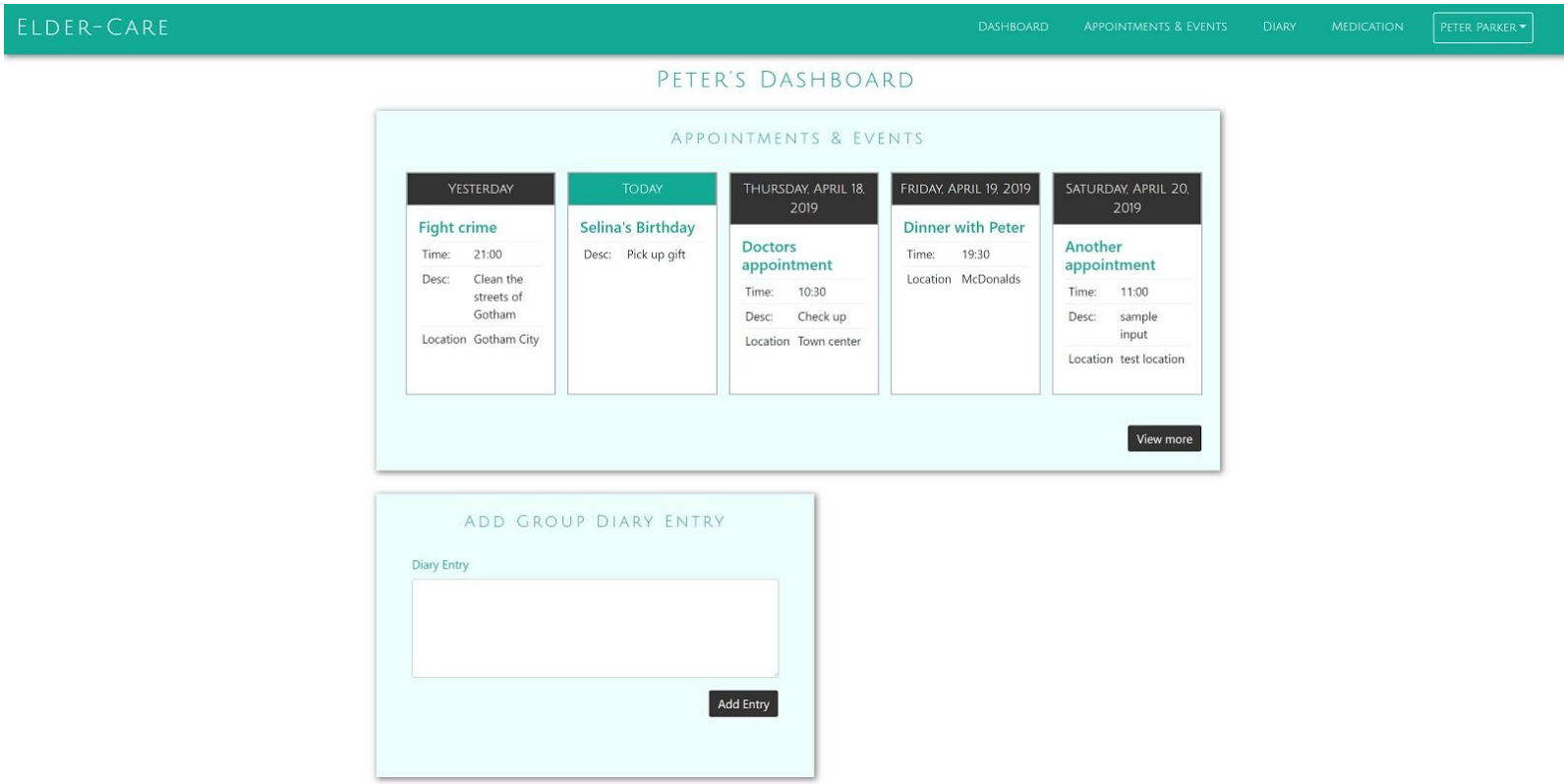


Fig.3: Care Assistant Dashboard

3.4.4 Dashboard on Mobile

The following screenshot is an example of an elderly users dashboard on mobile. As the project is a Progressive Web App the application must resize nicely for any screen size. Please note that this screenshot is of the top of the screen, the user would have to scroll to see the rest of the application. For the purposes of this section some of the content has been omitted.

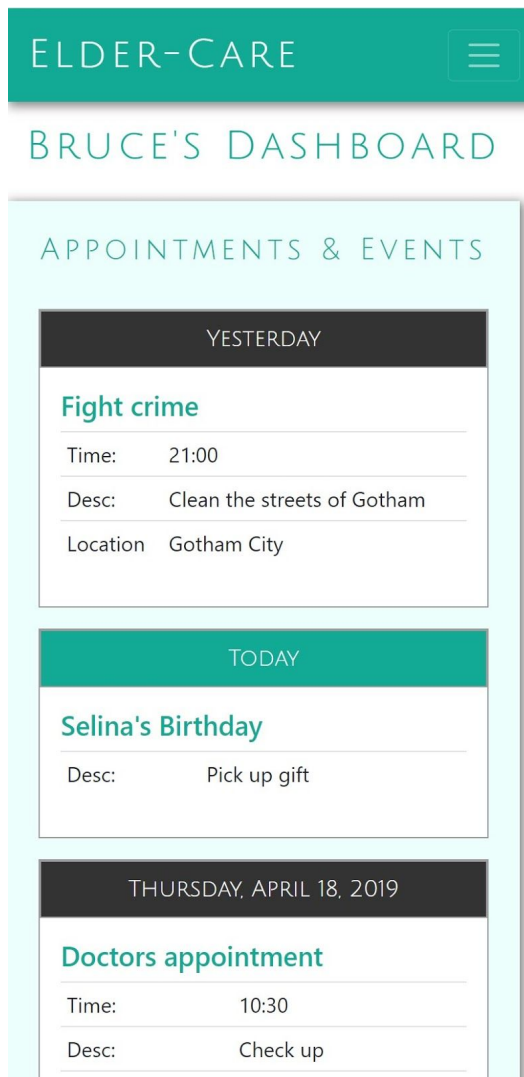


Fig.4: Mobile Version of Dashboard

3.4.5 Emergency Contacts

This screen is accessible to both kinds of user. Both elderly and care assistant accounts can add to emergency contact list. When an assistant account is created, the users details is automatically added to this list. Likewise when a user updates their details this list is updated. As this screen is deemed important it can be found in multiple locations. There is a link in the footer on every screen and also a link in the menu to the right of the navigation bar.

First	Last	Phone Number
Jim	Gordon	123456
Peter	Parker	654321
Clarke	Kent	98765
Craig	Lawlor	1234567
Selina	Kyle	0998765
Bruce	Wayne	123456

Fig.5: Emergency Contacts

3.4.6 Emergency Contacts on Mobile

The following screen is the mobile version of the emergency contacts screen.

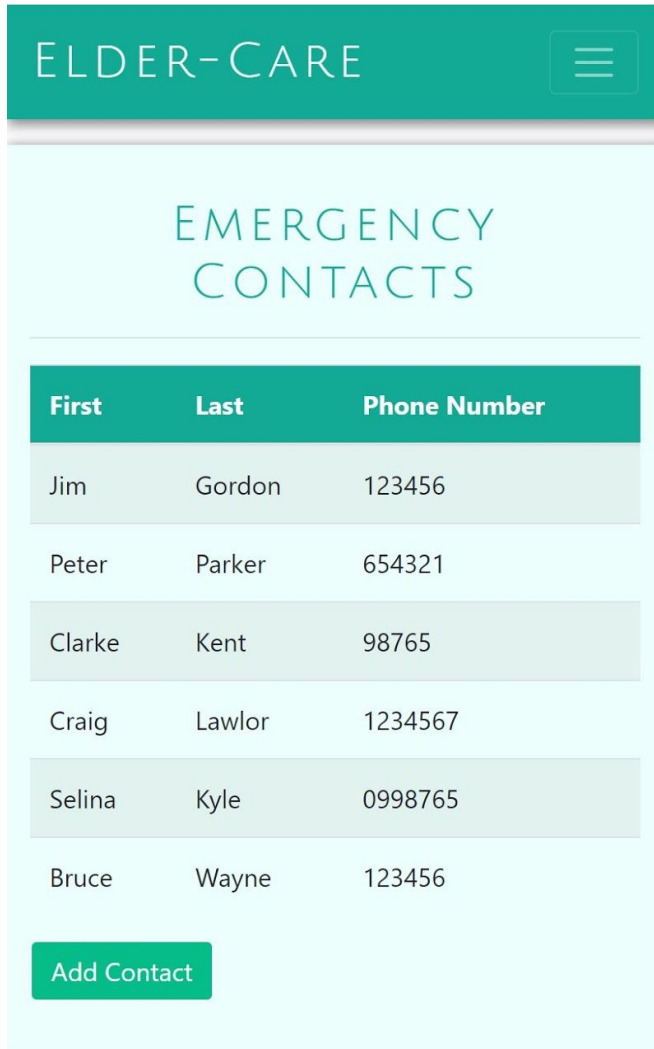


Fig.6: Mobile Version of Emergency Contacts

3.4.7 Add Appointment/Event

This screen is displayed when the user wishes to add an appointment or event. The Title and Date fields are required and must be entered to submit the form. The three additional fields are optional. When the form is successfully completed the user is redirected to the appointment and events screen.

The screenshot shows the 'APPOINTMENTS & EVENTS' form in the ELDER-CARE system. The form is titled 'APPOINTMENTS & EVENTS' and includes a header with navigation links: DASHBOARD, APPOINTMENTS & EVENTS, DIARY, and MEDICATION. The user's name, BRUCE WAYNE, is displayed in the top right corner. The form contains the following fields:

- Title***: A text input field with the placeholder 'Enter title'.
- Date***: A date input field with the placeholder 'dd/mm/yyyy'.
- Time (24hr)**: A time input field with the placeholder '--:--'.
- Description**: A text input field with the placeholder 'Enter description'.
- Location**: A text input field with the placeholder 'Enter location'.

A green 'Submit' button is located below the form fields. A note above the form states: 'Required fields are indicated with a *'. The footer of the page displays the date 'Wednesday, April 17, 2019', the copyright 'ELDER-CARE © 2019', and a link for 'Emergency Contacts'.

Fig.7: Add Appointment/Event

3.4.8 Appointments & Events

This screen is the main page for appointments and events. When an appointment is entered it is sorted and inserted into the appropriate place by date. The following appointments move down one position to the right. The user can filter the appointments by month or by all. All appointments are shown by default. As this is a single page application to improve the speed of querying by month when the page is loaded each month is pre populated into arrays which can be selected and loaded very quickly. Additionally when an appointment is added or removed these arrays are updated appropriately.

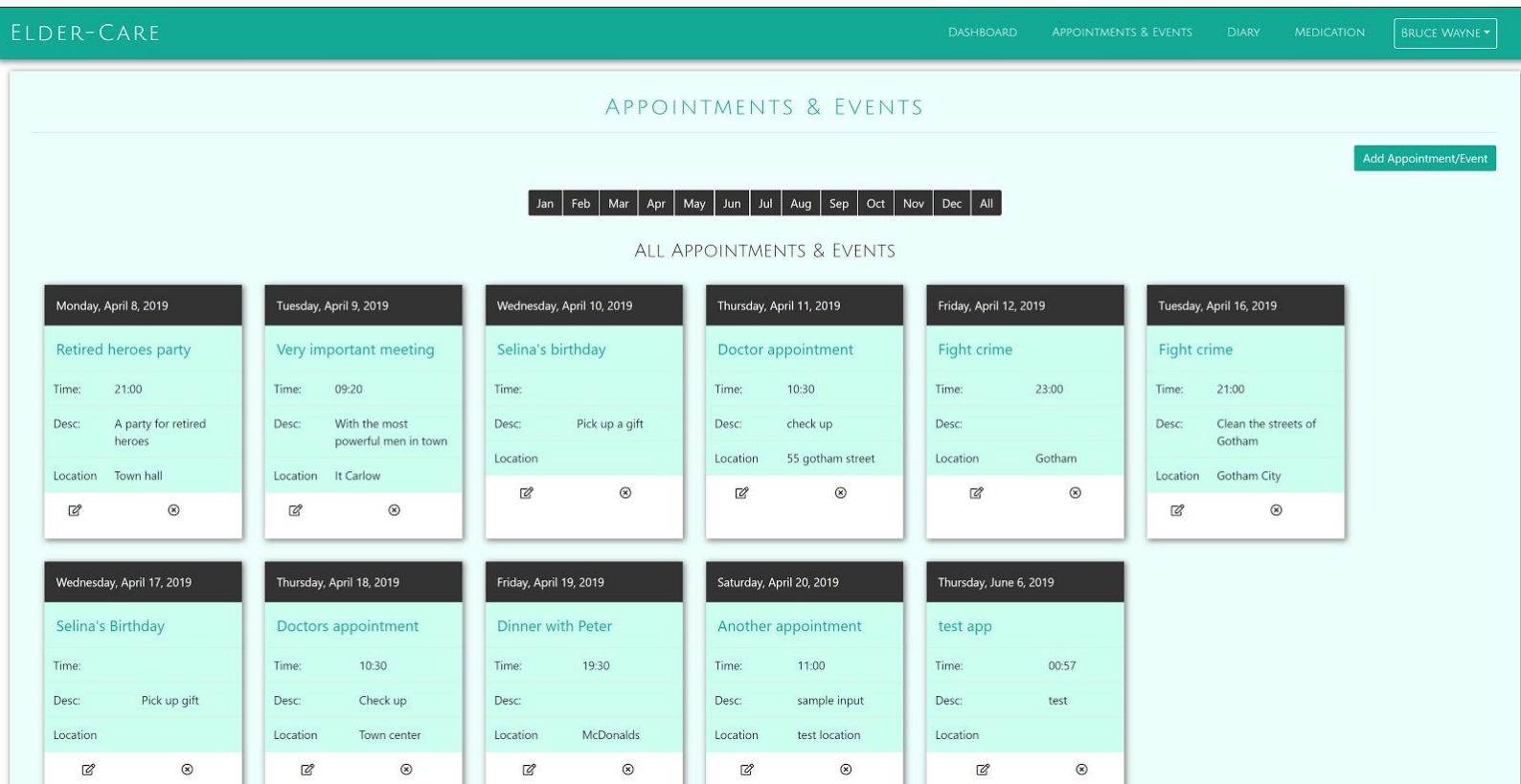


Fig.8: Appointments & Events

3.4.9 Add Appointment/Event

This is the mobile screen for appointments and events. Similarly to the mobile dashboard above in Fig.4 above not all of the information can be seen in this screenshot. The user would have to scroll to see all of the information.

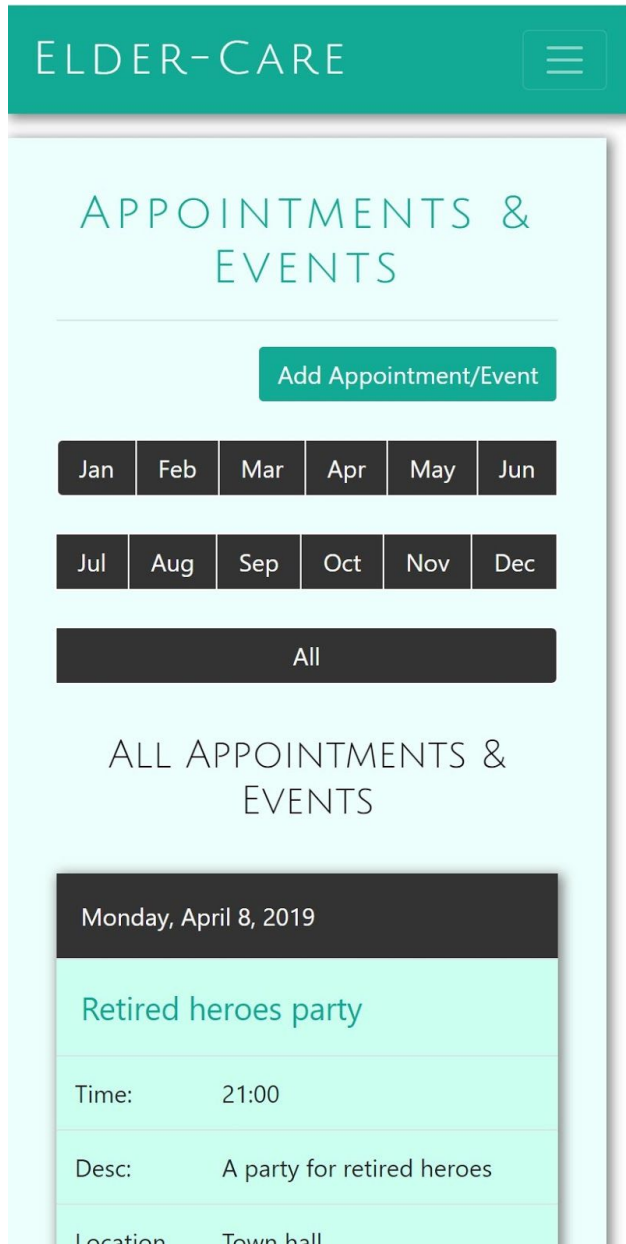


Fig.9: Mobile Version of Appointments & Events

3.4.10 Elderly Person’s Diary

The following screen is unique to elderly account holders. This diary section is only visible to the elderly account holder. A diary entry can be created by clicking the ‘Create Diary Entry’ button at the top right. Additionally a diary entry can be created from the dashboard. The user can choose to edit or remove a diary entry from this screen. The date at the top of the entry is the original date of creation. The date at the bottom of the entry is the date the entry was last edited.

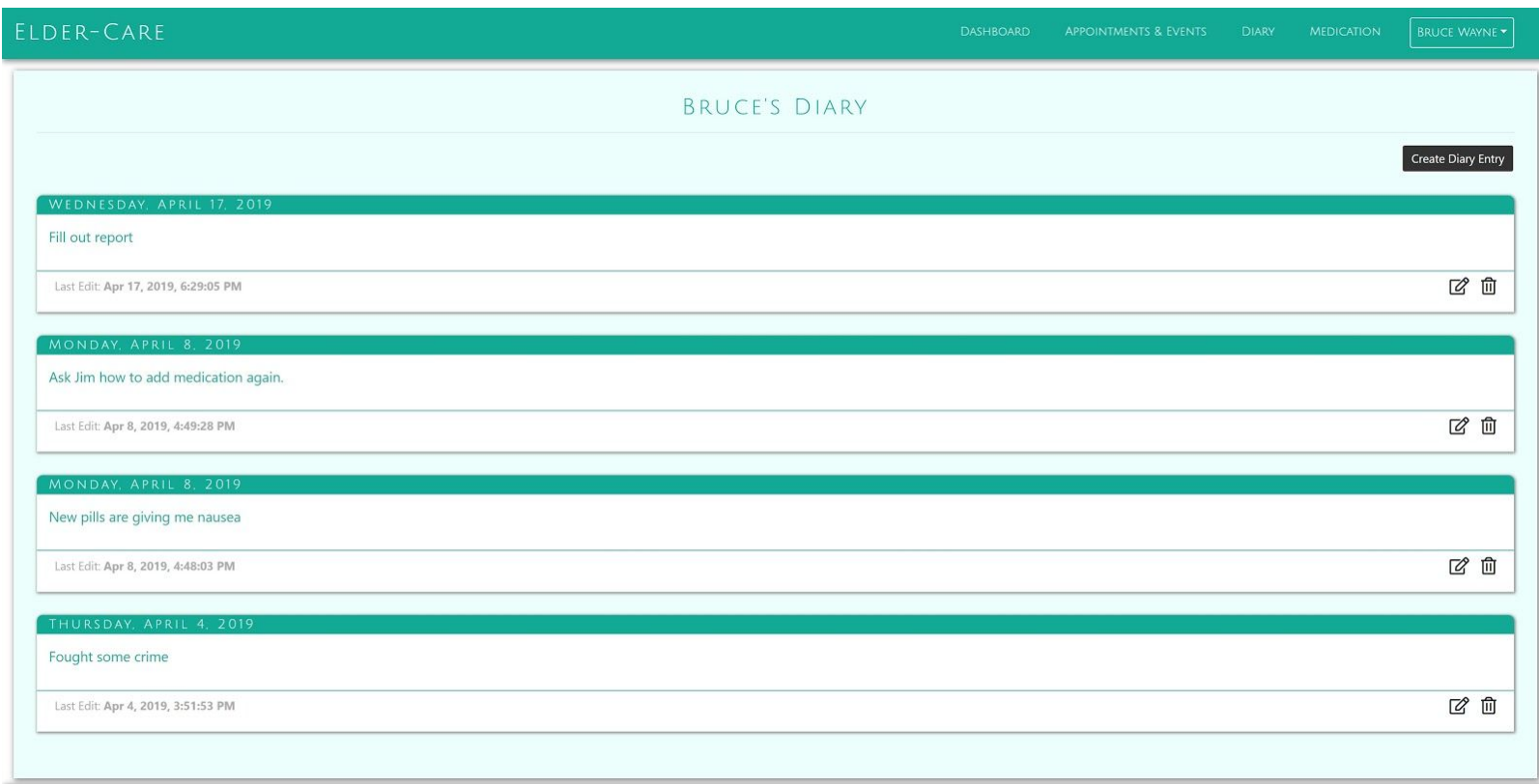


Fig.10: Elderly Person’s Diary

3.4.11 Elderly Person’s Diary on Mobile

This screen is the mobile version of the elderly persons diary.

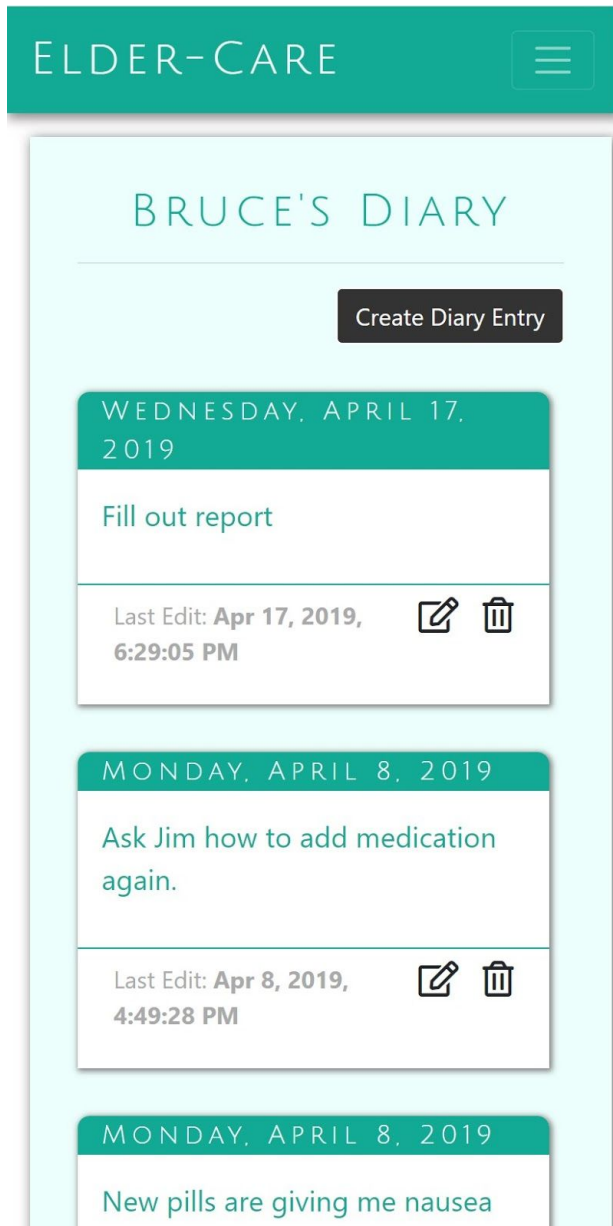


Fig.11: Mobile Version of Elderly Person’s Diary

3.4.12 Group Diary

Unlike the elderly persons diary this group diary is visible to all care assistant account holders, though it cannot be seen by the elderly account holder. Another difference is that the name of the person who created the diary entry can be seen on the top right of the entry. In the screenshot below the test user 'Peter Parker' can only edit or delete the second entry as they created it. A diary entry made by another user cannot be edited or removed. Please note that this section could also be viewed as a blog. The name group diary was chosen.

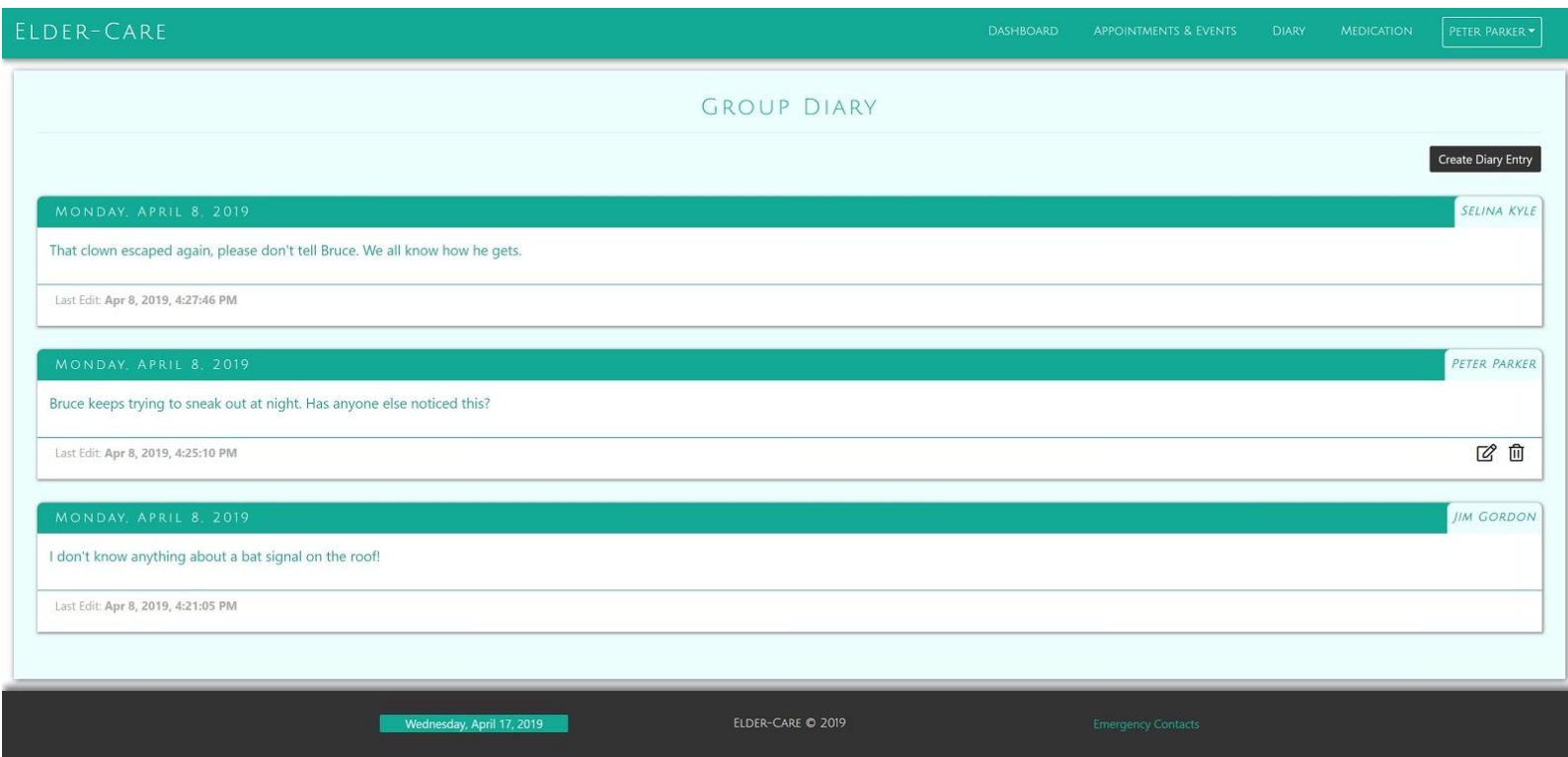


Fig.12: Group Diary

3.4.13 Group Diary on Mobile

The following screen is the mobile version of the group diary.

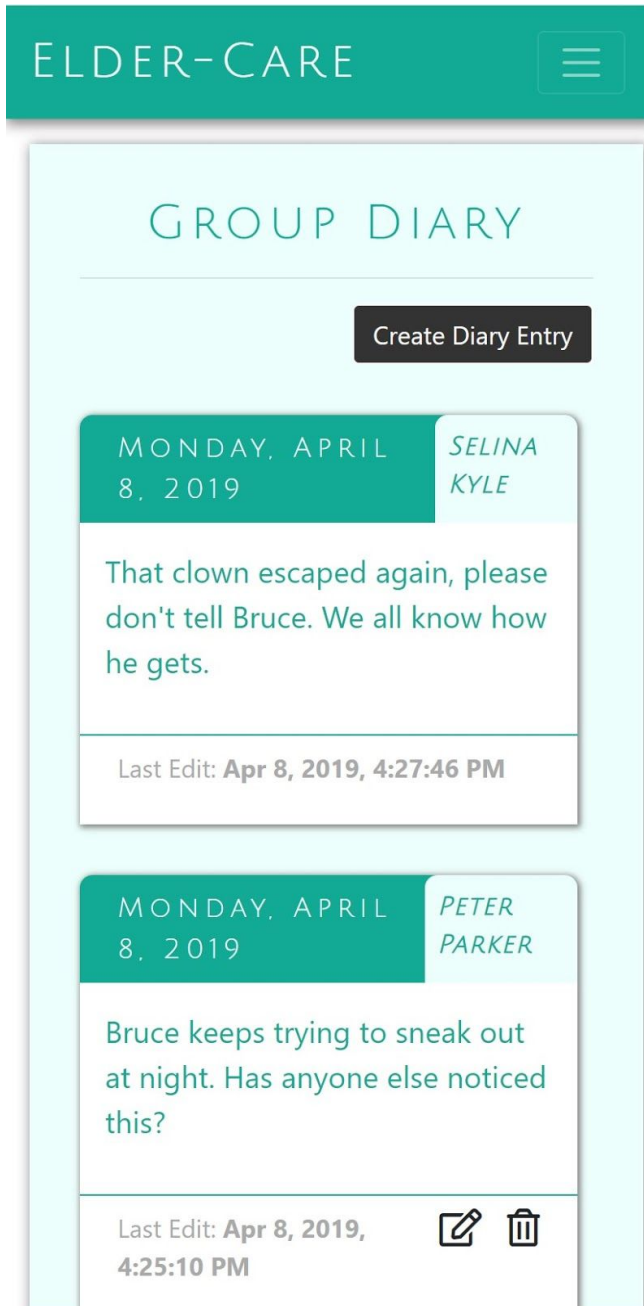


Fig.13: Mobile Version of Group Diary

3.4.14 Medication

The following screen is the medication chart for the elderly person. This screen can be seen by all accounts associated with the elderly account. The medication can be added to, edited by or removed by anyone. The aim of this section is that anyone who may be helping the elderly person with their medication can quickly check that they are giving the right medication and in the right amounts. The special instructions section can include any information which may be helpful.

The screenshot shows the 'MEDICATION' section of the ELDER-CARE interface. It features a table with the following data:

Medication Name	Dosage	Frequency (per day)	Special Instructions
Laughing gas antidote	1000mg	1	Take after a tangle with the Joker
paracetamol	500mg	3	
Bat meds	500mg	3	Take after meals

Below the table is an 'Add Medication' button. The footer of the page includes the date 'Wednesday, April 17, 2019', the copyright 'ELDER-CARE © 2019', and a link for 'Emergency Contacts'.

Fig.14: Medication

3.4.15 Medication on Mobile

The following screen is the mobile version of the medication screen.



Fig.15: Mobile Version of Medication

3.4.16 Update Details

This screen can be accessed by any user at any time to change their details. If a user changes these details the changes will be updated in the emergency contacts section.

The screenshot shows the 'UPDATE DETAILS' form within the ELDER-CARE application. The form is centered on a light blue background and contains the following elements:

- Header:** 'UPDATE DETAILS' in a light blue font.
- Instruction:** '* Only enter the fields you wish to change' in a small red font.
- Form Fields:**
 - First Name:** A text input field containing the value 'Bruce'.
 - Last Name:** A text input field containing the value 'Wayne'.
 - Phone Number:** A text input field containing the value '123456'.
- Submit Button:** A green button labeled 'Submit' located below the form fields.

The application's navigation bar is visible at the top, showing 'ELDER-CARE' on the left and 'DASHBOARD', 'APPOINTMENTS & EVENTS', 'DIARY', 'MEDICATION', and a user profile 'BRUCE WAYNE' on the right. The footer at the bottom displays the date 'Wednesday, April 17, 2019', the copyright 'ELDER-CARE © 2019', and a link for 'Emergency Contacts'.

Fig.16: Update Details

4. Conformance to Specification & Design

4.1 Early Prototype

The original plan for this project was to use a relational database such as MySQL and to use React and Angular together. The reason React was chosen was that when a React app is created it is already set up to be a Progressive Web App. This would have helped to speed things up in the beginning. Early challenges presented themselves with how React would work with the back end and how binding with Angular would work. At this point using a relational database was proving difficult as both React and Angular seemed to work better with noSQL databases.

From the research carried out at that point of the project it was discovered that single page applications such as applications created using Angular worked well with real time databases. Another question posed at this point of the project was how authentication would work with the chosen technologies. As authentication was an important part of the project along with the creation of the different types of user accounts, it was important to get this right at the beginning of the prototype.

4.2 Changes made to Original Specification & Design

Having carried out much research posed by the challenges above a decision was made to use Firebase for the back end of the project. Firebase is a noSQL real time database owned by Google. Though the database for the project was already designed for a relational database, much of the information could be reused in the design of the noSQL database. Coding had already begun using React but it became apparent early on that Firebase and Angular worked well together through the AngularFire library.

After some further research a decision was made to drop React and use Angular. This would mean that the Progressive Web App capabilities would have to be created as Angular did not include this by default like React. The first steps taken from that point were to create the shell of the application using Angular and Bootstrap and provide the files and code to make the application progressive. Once the basic progressive functionality was implemented it was time to begin the authentication phase with Firebase.

5. Description of Learning

5.1 Personal

Personally what was learned most from this project is how important it is to manage time. The overall course was very time demanding and the main project often got pushed to the side. If I had known we would have an assignment for every other module in the final month of the project I would have done many things differently. The pressure created from the demands of the other modules impacted the final project produced greatly. The only solace I can take from the time problems is that everyone was in the same boat. As time went on I found I was capable of getting a lot of work done at certain points but this was to the detriment of family life. I believe in the final two weeks the amount of work completed well surpassed what I had expected. That being said if I had the time back again I would have ensured that I spent at least a certain amount of time per week on the project.

With regards to communication I learned the hard way that when you have a problem you should seek help earlier rather than later. I gave up on supervisor meetings by January as I felt I was not gaining any useful guidance. I would leave meetings with more pressure rather than the direction I required. I decided at this point to go things alone. By the time I sought help/advice regarding the issues I faced too much time had passed. From the experience I have learned not to let problems fester too long and to communicate better with people. With all of this in mind I am extremely proud of what I achieved alone.

5.2 Technical

With regard to technical learning I had never created a single page application or a Progressive Web App. This meant I had to complete many tutorials and courses on Angular as well as some tutorials on Progressive Web Apps. This took a lot of time in the beginning and continued intermittently throughout the project. Prior to this project I had some experience with Javascript but I would not say I was very proficient with it. As Angular uses a language called Typescript which is based on Javascript there was a lot to learn personally.

Prior to this project I had quite a lot of experience with Bootstrap and CSS styling, though I had never integrated Bootstrap with Angular. Once some early teething problems were ironed out the styling of the application went very smoothly. As a

Progressive Web App must be fully responsive Bootstrap was very helpful. From the research carried out on Angular the main styling framework used is Angular Material. I made the decision to use Bootstrap instead based on my prior experience. In the end I was glad with this decision.

As mentioned in a previous section the chosen back end was Google's Firebase. There were two types available, the Real Time Database and the newer Cloud Firestore. I chose to use Cloud Firestore as it will eventually replace the real time database. I had some prior experience with noSQL databases such as MongoDB but I was in no way proficient. That being said after an extremely difficult learning curve I found Firebase very nice to use. Also mentioned in a previous section, I used a library called AngularFire which made connecting these languages much more manageable.

The package manager I used for this project was Node JS. I had used this package manager very briefly while on work placement and so was not very proficient with it. That being said I found it invaluable at all stages of development for the project. Everything from creating an angular project to the creation of individual Angular components to running localhost servers Node JS was absolutely fundamental. I would view Node JS as the foundational building blocks on which the project is built.

The IDE used for development was JetBrains' Webstorm. All aspects of development were carried out within this IDE. The built in terminal made working with Node JS very easy, and once a localhost server was running changes made to the code were instantly updated in the web browser. Webstorm also made version control quite easy with it built in functionality to connect to a Github repository. Changes could be committed and pushed very easily. Though I had always used the command line for git in the past the functionality provided by webstorm was a welcome change for me.

6. Project Review

6.1 What went wrong

In the beginning of the coding phase the technologies chosen did not work out as intended. This meant certain things like the back end had to be redesigned. This took more time than expected even though the general layout was the same. As the back end technology was different the approach to connecting the technologies used for the front end and logic had to be reassessed. This meant another change in technology moving from using React and Angular together to using Angular on its own. I felt at the time Angular and Firebase would work better together. Unlike React, Angular did not create the required setup for a Progressive Web App. This meant I had to set this up myself, which in turn meant I had to take a number of courses online to learn how to create a Progressive Web App. A further problem I faced with Angular was the version I was using. I chose Angular 7 as it was the latest version at the time. Though much of the documentation and examples found online were for older versions and many of the functions were deprecated.

Though I found Firebase very nice to use overall, in the beginning I found the learning curve very difficult. Firebase provides some nice features for setting up authentication within an application. There are many ways this authentication can be done such as via Google login, Facebook login and many more. I chose to use the email and password authentication. Once setup the users email and password was stored in Firebase and a unique ID was created. The problem I faced was setting up a user collection with this UID (unique ID). It took a lot longer than I thought it would. Part of the problem to using the newer Cloud Firestore was that is fairly new and a lot of the documentation and examples found online were for the Real Time Database. This was a lot more problematic than I thought it would be. I decided to stick with the Cloud Firestore version and try to get things working. Eventually things started to work.

Continuing on from the problems above the library used to join Angular and Firebase, AngularFire, was also moving to a newer version. Much of the examples found online were for the older version and the imports were wrong in many places. Again this took more time to get right than expected. A particular area of the application where this was problematic was in the authGuard section. To stop certain areas being used by unauthenticated users code had to put in place. The problem for me at this point was getting the information needed from Firestore for the user that was authenticated. This reflects what I mentioned earlier about connecting the authentication to the user collection. As time went on I found better ways of getting the information I needed from

Firestore. The approach for a lot of the retrieval of information was quite different to that of a relational database.

In general I found working with dates in various areas of the application quite challenging. Firebase uses a timestamp format which I had to split into a date and time for use with the appointments & events section. I had to work with how Angular handled dates and how Firebase handled dates. It sounds simple thinking back on it but it was a lot more difficult than I expected.

6.2 What went right

As mentioned earlier in a previous section the styling aspects of the application went very smoothly. Once passed the steep learning curves of the above mentioned problems development in general went very well. I found using things I had learned in other areas of the code could be applied to new components I was working on. One of the most difficult parts of the project was getting the different types of user accounts working together whilst only showing the appropriate information to the authenticated user. Though it was difficult I felt it went quite well especially when I became more comfortable with the technologies.

After some time working with Firebase I found it worked very well and the real time updates helped identify if things were working correctly. It felt much more flexible than any relational database I have used before. When changes needed to be made it was quite painless as I could just change the key-value pairs within the code and as long as the code was pointing in the right direction the information would be stored as I intended in the database.

6.3 What I would do differently

If I were to start this project again I would ensure to communicate problems earlier and not let things fester. In terms of the project code I would spend more time in the beginning working with the interconnectivity of the Angular components. I feel I did not learn how to properly share variables between components and may have saved myself time in the long run if I had have learned how this was done properly in the beginning. To add to this point I believe I could have cut the code count down substantially by reusing existing code in certain areas.

Additionally I would have used more service Typescript files to reduce code count. I may have used Firestore's real time database if I were to do things again, purely to save time. Though looking back I am glad I stuck with Cloud Firestore as it is the future of Firebase. I would have also liked to put the project live on a server to have been able to show off more of the progressive aspects of the project. I felt in the demo this side of the project fell a bit flat, especially after the amount of time I spent making the project as a Progressive Web App.

With regard to time management I would have spent more time on the project rather than allowing the demands of the other modules to take over.

7. Conclusion

Though slow to get started this project ran for almost all of the final year. It was extremely demanding given the deadlines, either official or created to cause pressure. There were many aspects I found very challenging, some of which were time pressures and lack of support. With that being said I am very proud with how the project turned out. I learned a lot about technologies I had never used before and even more about technologies I had used. Once things got moving I found the development of the project very enjoyable and rewarding.

8. Acknowledgements

I would like to thank Dr Joseph Kehoe for his continued support throughout the year. I would also like to thank my mother without who I could not attend college, my wife for her continued support and my children for their understanding on why 'Daddy' has to sit for so many hours in the small room on his computer.

To my grandfather Michael who was the inspiration and the intended user for this project. He sadly passed away the day I gave my project demo. Rest in peace.