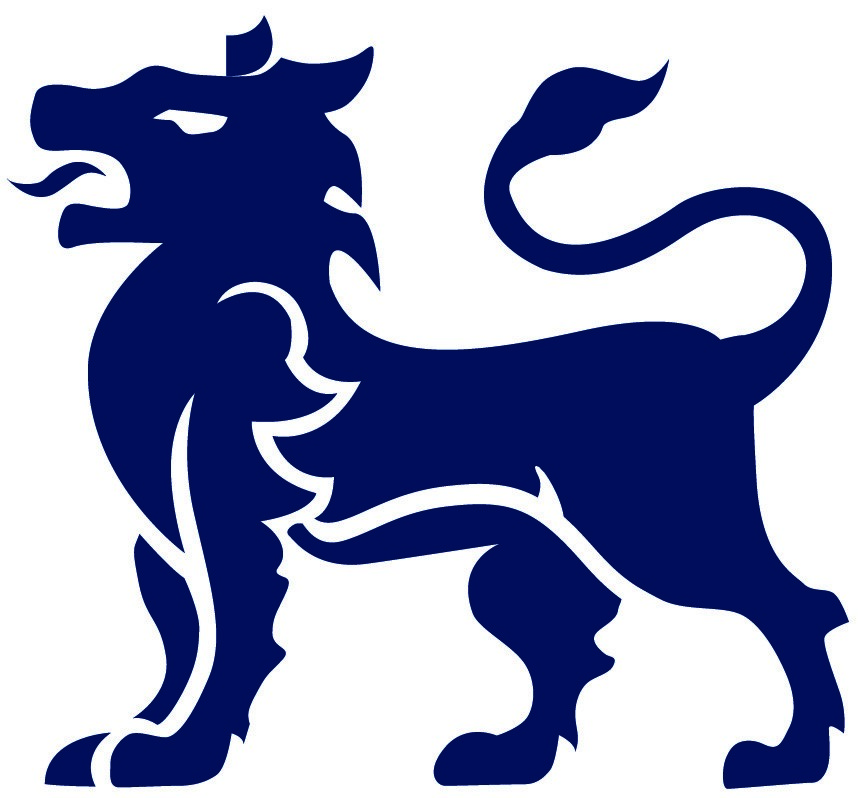
Student Mental Health Signposting App (My Mind)

Alexander Edward Davis



A report submitted as part of the requirements for the degree of BSc in Computer Science at the School of Computing and Digital Technology

Birmingham City University, Birmingham, UK

May 2017

Supervisor Dr. Yevgeniya Kovalchuk

Abstract

The Abstract of the report should be written here, it should provide a short summary of the work encompassing no more than one page.

# Acknowledgements

Dr. Yevgeniya Kovalchuk has been an incredibly supportive thesis supervisor. Her insightful criticisms and patient encouragement aided the progress within this project in innumerable ways.

# Dedication

This thesis is dedicated to my father, who sadly passed away in July 2017. It is also dedicated to Peter Bell, a friend who always questioned my processes and pushed me to do the very best within my field.

# Declaration

****I confirm that the work contained in this BSc project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.

Signed: Date: 19th May 2017

Alexander Edward Davis

Table of Contents

[Abstract ii](#_Toc481977395)

[Acknowledgements iii](#_Toc481977396)

[Dedication iii](#_Toc481977397)

[Declaration iv](#_Toc481977398)

[Table of Contents v](#_Toc481977399)

[List of Tables xi](#_Toc481977400)

[List of Figures xii](#_Toc481977401)

[Introduction 13](#_Toc481977402)

[1.1 Background 13](#_Toc481977403)

[1.1.1 Project Background 13](#_Toc481977404)

[1.1.2 Aim 13](#_Toc481977405)

[1.1.3 Objectives 13](#_Toc481977406)

[1.1.4 Product 15](#_Toc481977407)

[1.2 About this Thesis 17](#_Toc481977408)

[1.2.1 Reasons for Project 17](#_Toc481977409)

[1.2.2 Project Effects 17](#_Toc481977410)

[1.2.3 Comparison of other possible solutions 18](#_Toc481977411)

[1.3 Chapter List 18](#_Toc481977412)

[Literature Review 19](#_Toc481977413)

[2.1 Introduction 19](#_Toc481977414)

[2.2 The Deployment of Mental Health Services 1](#_Toc481977415)

[2.2.1 Mental Health Services in the United States of America 1](#_Toc481977416)

[2.2.2 Mental Health Services in the United Kingdom 1](#_Toc481977417)

[2.3 The Concept of Social Media 2](#_Toc481977418)

[2.3.1 Different Types of Social Media Platform 2](#_Toc481977419)

[2.3.2 The Uses of Social Media 2](#_Toc481977420)

[2.4 The Use of Social Media Within Mental Health Services 3](#_Toc481977421)

[2.4.1 Adoption within Education Institutions 6](#_Toc481977422)

[2.5 Conclusion 7](#_Toc481977423)

[Background Research 8](#_Toc481977424)

[3.1 Research Subjects 8](#_Toc481977425)

[3.2 Research Methods 8](#_Toc481977426)

[3.2.1 Interviews 8](#_Toc481977427)

[3.2.2 Analysis of Current System 9](#_Toc481977428)

[3.3 Conclusion 9](#_Toc481977429)

[Requirements 10](#_Toc481977430)

[4.1 App Design Requirements 10](#_Toc481977431)

[4.2 Platform Requirements 10](#_Toc481977432)

[4.3 Accessibility Requirements 10](#_Toc481977433)

[4.4 Time Restraints 10](#_Toc481977434)

[4.5 Conclusions 10](#_Toc481977435)

[Design 11](#_Toc481977436)

[5.1 App Workflow 11](#_Toc481977437)

[5.2 App Design Criteria 11](#_Toc481977438)

[5.3 Platform Decisions 11](#_Toc481977439)

[5.4 Conclusion and Final Design 11](#_Toc481977440)

[Implementation 12](#_Toc481977441)

[6.1 Implementation Options 12](#_Toc481977442)

[6.1.1 Whole System Implementation 12](#_Toc481977443)

[6.1.2 System Integration 12](#_Toc481977444)

[6.1.3 Phasing Implementation 12](#_Toc481977445)

[6.2 Conclusions and Final Implementation Decisions 12](#_Toc481977446)

[Testing 13](#_Toc481977447)

[7.1 Testing Techniques 13](#_Toc481977448)

[7.2 Test Plans 13](#_Toc481977449)

[7.3 Conclusions 13](#_Toc481977450)

[Evaluation of Product 14](#_Toc481977451)

[8.1 Evaluation Techniques 14](#_Toc481977452)

[8.1.1 Demonstrations 14](#_Toc481977453)

[8.1.2 Mental Health Awareness Week Stands 14](#_Toc481977454)

[8.1.3 App Questionnaires 14](#_Toc481977455)

[8.2 Comment Reviews 14](#_Toc481977456)

[8.3 Conclusions 14](#_Toc481977457)

[Evaluation of Process 15](#_Toc481977458)

[9.1 Personal Evaluation 15](#_Toc481977459)

[9.2 Possible Future Changes 15](#_Toc481977460)

[9.1 Conclusions 15](#_Toc481977461)

[Conclusion 16](#_Toc481977462)

[10.1 Conclusions 16](#_Toc481977463)

[10.2 Future Work 16](#_Toc481977464)

[References 17](#_Toc481977465)

[Bibliography 18](#_Toc481977466)

[Appendix A – Sample Questionnaire & Consent Form 19](#_Toc481977467)

[Appendix B – Presentation 22](#_Toc481977468)

[Appendix C – Source Code 23](#_Toc481977469)

[AppDelegate 23](#_Toc481977470)

[OnBoardingController 26](#_Toc481977471)

[Welcome Screens 28](#_Toc481977472)

[WelcomeScreenOne 28](#_Toc481977473)

[WelcomeScreenTwo 28](#_Toc481977474)

[WelcomeScreenThree 28](#_Toc481977475)

[Login Section and User Information 29](#_Toc481977476)

[UserViews 29](#_Toc481977477)

[Main Tab Views 32](#_Toc481977478)

[TabBarController 32](#_Toc481977479)

[MainViews 32](#_Toc481977480)

[FormViews 33](#_Toc481977481)

[ForumView 42](#_Toc481977482)

[Appendix D – Student Feedback Results 43](#_Toc481977483)

[Student Information and Groupings 43](#_Toc481977484)

[Participant Answers 43](#_Toc481977485)

[Strongly Agree 44](#_Toc481977486)

[Agree 44](#_Toc481977487)

[Some- what Agree 44](#_Toc481977488)

[No Opinion 44](#_Toc481977489)

[Some- what Disagree 44](#_Toc481977490)

[Disagree 44](#_Toc481977491)

[Strongly Disagree 44](#_Toc481977492)

[I found this app easy to use. 44](#_Toc481977493)

[5 44](#_Toc481977494)

[2 44](#_Toc481977495)

[I can use it without written instructions. 44](#_Toc481977496)

[5 44](#_Toc481977497)

[1 44](#_Toc481977498)

[1 44](#_Toc481977499)

[I can navigate through the app easily. 44](#_Toc481977500)

[5 44](#_Toc481977501)

[2 44](#_Toc481977502)

[The language used in the app is easily understood. 44](#_Toc481977503)

[5 44](#_Toc481977504)

[2 44](#_Toc481977505)

[The different sections/pages of the app are cohesive in look and feel. 44](#_Toc481977506)

[6 44](#_Toc481977507)

[1 44](#_Toc481977508)

[The app has adequate help documentation. 44](#_Toc481977509)

[1 44](#_Toc481977510)

[2 44](#_Toc481977511)

[1 44](#_Toc481977512)

[3 44](#_Toc481977513)

[The app has clearly marked way-finding buttons (exit, info, back, next page, etc.). 44](#_Toc481977514)

[1 44](#_Toc481977515)

[6 44](#_Toc481977516)

[The prompts for input are clear. 45](#_Toc481977517)

[3 45](#_Toc481977518)

[4 45](#_Toc481977519)

[The prompts for input are effective in making you want to contribute. 45](#_Toc481977520)

[5 45](#_Toc481977521)

[2 45](#_Toc481977522)

[The app does everything I would expect it to. 45](#_Toc481977523)

[4 45](#_Toc481977524)

[2 45](#_Toc481977525)

[1 45](#_Toc481977526)

[The app works the way I would want it to work. 45](#_Toc481977527)

[4 45](#_Toc481977528)

[2 45](#_Toc481977529)

[1 45](#_Toc481977530)

[The app is designed for all levels of users. 45](#_Toc481977531)

[4 45](#_Toc481977532)

[1 45](#_Toc481977533)

[2 45](#_Toc481977534)

[Using the app was an enjoyable experience. 45](#_Toc481977535)

[3 45](#_Toc481977536)

[2 45](#_Toc481977537)

[2 45](#_Toc481977538)

[I found the app unnecessarily complex. 45](#_Toc481977539)

[5 45](#_Toc481977540)

[2 45](#_Toc481977541)

[I found the various functions in the app were well integrated. 45](#_Toc481977542)

[3 45](#_Toc481977543)

[3 45](#_Toc481977544)

[1 45](#_Toc481977545)

[I think that I would use this app frequently. 46](#_Toc481977546)

[1 46](#_Toc481977547)

[1 46](#_Toc481977548)

[3 46](#_Toc481977549)

[1 46](#_Toc481977550)

[1 46](#_Toc481977551)

[I felt very confident using this app. 46](#_Toc481977552)

[4 46](#_Toc481977553)

[3 46](#_Toc481977554)

[Open Ended Question Feedback 46](#_Toc481977555)

[Did Anything in The App Confuse You 46](#_Toc481977556)

[Negative Aspects 46](#_Toc481977557)

[Positive Aspects 46](#_Toc481977558)

[Feature Requests 47](#_Toc481977559)

[Overall Feedback 47](#_Toc481977560)

[47](#_Toc481977561)

[47](#_Toc481977562)

[Appendix E - UCEEL Copyright Waiver 48](#_Toc481977563)

List of Tables

Table 1 - An Overview of the Academic Literature Surrounding the Effectiveness of Social Media within Mental Health Services 19

List of Figures

Figure 1 – Current BCU iCity Menu System 12

Figure 2 – Image showing sidebar not meant to be used by students 12

Figure 3 – Image showing question with unusable answers 12

Figure 4 – Diagram of the three sections of the app and how a user could access the other sections of the app. 14

Figure 5 – Question giving advice regarding mental health issues 4

Figure 6 – Question giving advice regarding mental health issues 5

Figure 7 – Question giving advice regarding mental health issues 5

Chapter 1

# Introduction

## 1.1 Background

### 1.1.1 Project Background

“In recent years, health problems, particularly mental health of students is so interested. College is an important setting in which to evaluate and address mental health.”

(NAMI ET AL., 2014)

Today’s society has recently begun speaking out on controversial and sensitive subjects become part of a regular conversation. This has started to include objects such as mental health and wellbeing including the diagnosis that many students who are at university having a mental health condition such as depression and anxiety disorders.

Many students starting at university are now facing the issue where they are unable to find support or unaware of any of the mental health support mechanisms available at their university.

### 1.1.2 Aim

The main aim of this project was to design, create and implement a mobile application that allows students at Birmingham City University can see the mental health support available from both the university and the students within the university.

The app also allows them to refer themselves to the Mental Health and Wellbeing student services team within the university and book and appointment with a mental health and wellbeing advisor.

### 1.1.3 Objectives

The mobile app follows the Apple iOS Human Interface Guidelines as I have primarily developed the app on the iOS mobile platform. Apple’s software design guidelines state:

“Three primary themes differentiate iOS from other platforms:

Clarity. Throughout the system, text is legible at every size, icons are precise and lucid, adornments are subtle and appropriate, and a sharpened focus on functionality motivates the design…

Deference. Fluid motion and a crisp, beautiful interface help people understand and interact with content while never competing with it…

Depth. Distinct visual layers and realistic motion convey hierarchy, impart vitality, and facilitate understanding…”

(APPLE INC, 2016)

I originally designed the app so it will work on iPhones and iPods however in future I will include iPads into my designs with the possibility to expand the deployment of this application to work with Android devices.

The app must be able to work within file and data entry systems including the Microsoft SharePoint platform. These are used by the Student Mental Health and Wellbeing services within the university to allow students to refer themselves to the department. The implementation of this is not very good as many students must be referred to the actual form by a member of staff and to find this form on iCity, the university’s information portal for staff and students, can be incredibly difficult which could potentially put off students from filling the referral form.

Figure one shows the current menu for the iCity information portal. As you can see, the ability to view the mental health and wellbeing page is not there however after several minutes of clicking you will be able to access the page by clicking several hyperlinks on many pages. Personally, as I have accessed this page previously, I am able to access it within a matter of minutes with only four clicks however some new students who are not used to the layout of iCity may not be able to do so in a similar fashion.

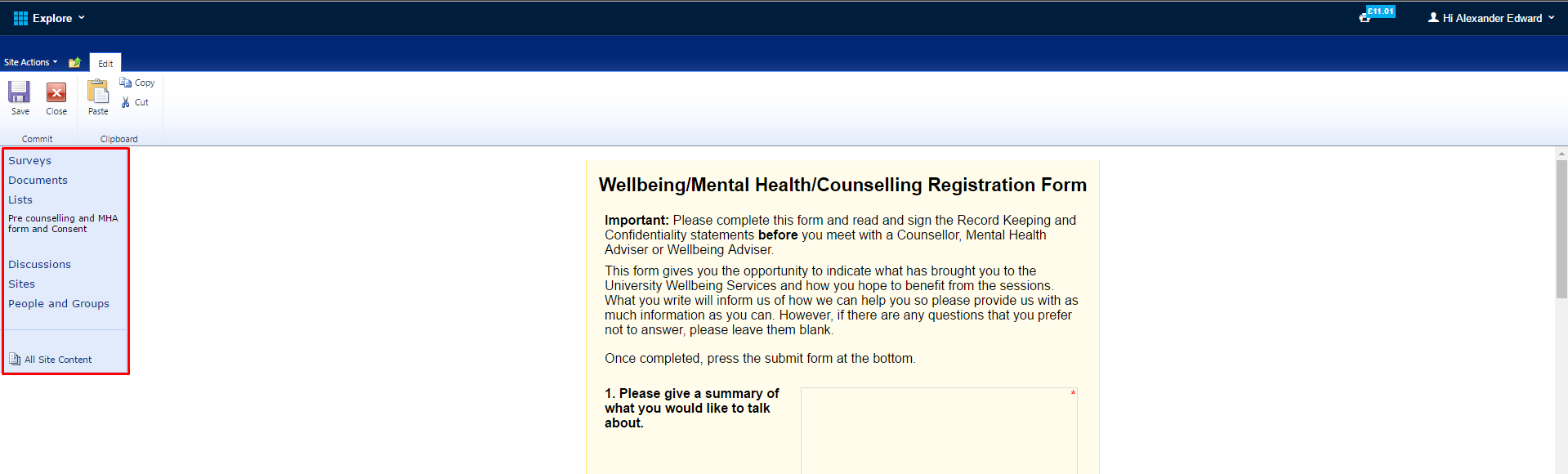


Source: (BIRMINGHAM CITY UNIVERSITY, 2015)

Figure – Current BCU iCity Menu System

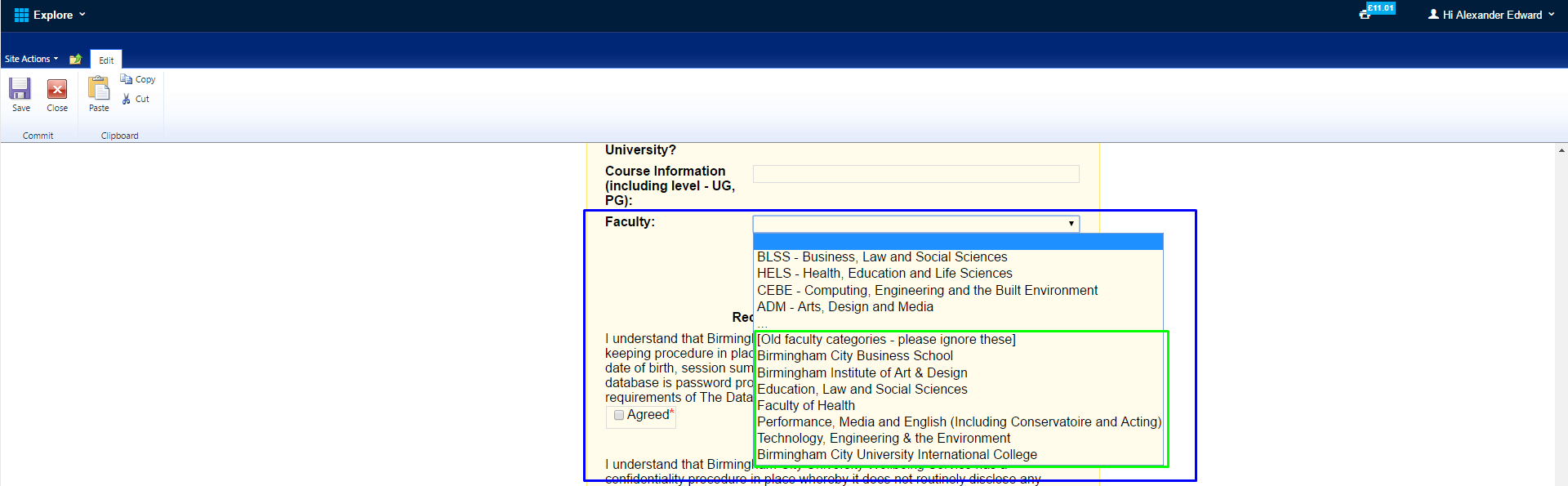
In the near future, the app will also allow members of staff to create, amend and cancel appointments for students. This will be done via an administrative website which will be linked to the same database as the student section. There is also the possibility to include the Doodle Online Appointment Management system into this as Doddle allows people to suggest times and view the availability of a specific person.

The app also includes a new, updated version of the referral form. The current design is not user friendly and some students may encounter difficulty when viewing and filling in the form. The form also includes unnecessary items, for instance, Figure 2 displays the sidebar (shown in the red box) at the top of the page that students do not need to access. Figure 3 also shows a question within the form that includes information that is now no longer valid. The question (shown in the blue box) in Figure 3 asks the user which faculty they belong to and includes faculties that now no longer exist (shown in a green box) and have merged to form other faculties. The design of this form has even includes a warning not to use these answers as they are not valid.



Source: (Birmingham City University, no date)

Figure – Image showing sidebar not meant to be used by students



Source: (BIRMINGHAM CITY UNIVERSITY, NO DATE)

Figure – Image showing question with unusable answers

### 1.1.4 Product

Figure 4 shows the three different sections of the app that users can interact with. The app will communicate with an online database to allow the input of data such as appointments and allow access to the forum, this has been shown with the database section holding three tables for the appropriate app areas.

The main referral section of the app will contain a user creation page where users will be required to create their own account for the service. This account implementation is for the forum and so that the student can create appointments with Mental Health and Wellbeing Advisors. Although there is a possibility for me to use the university’s single sign on service in future, I have decided that this is not a good idea for the current project as I would like to trial it on a select focus group and do not want to allow all students access to the app or the forum at this moment in time. There is the possibility in the future to implement the OpenAthens Single Sign-On service that most UK universities use. This would allow me to create a location specific, university specific experience for each user. Another reason for not requesting to the OpenAthens service is due to the time restriction within this project, there are several legal issues I would have to face and discuss with the service provider and this process would take longer than the allocated time I have for this project.



Figure – Diagram of the three sections of the app and how a user could access the other sections of the app.

## 1.2 About this Thesis

This is the report of Alexander Edward Davis, submitted as part of the requirements for the degree of BSc in Computer Science at the School of Computing and Digital Technology, Birmingham City University, Birmingham, UK.

### 1.2.1 Reasons for Project

Higher Education Institutions are seeing a common trend within first year students where around 6% of all first year entrants do not continue with their studies according to data from the Higher Education Statistics Agency. The Times Newspaper states

“Six per cent of first degree entrants aged under 21 who enrolled in 2013-14 did not continue their studies beyond their first year, according to data from the Higher Education Statistics Agency.”

(HAVERGAL ET AL., 2016)

Many institutions have theorised that most non-continuation cases are cause by mental health issues such as anxiety and depression. To combat this ongoing issue, many universities have set-up counselling services for students suffering with mental health conditions. Although the number of cases is not known, many counselling services are reporting an increase in referrals as stated by Ann Macaskill:

“There are increasing concerns globally about the mental health of students. In the UK, the actual incidence of mental disturbance is unknown, although university counselling services report increased referrals.”

(MACASKILL, 2013)

As these services are becoming available, I have noticed many issues with the referral system and knowledge of their existence within different university institutions. One common issue I have seen is that many students are aware how to get counselling support however the university’s website makes it rather difficult for the student to refer themselves to the support services available.

### 1.2.2 Project Effects

This project will allow the ease of use for support services to be accessed and will also create the availability where many students who do not want staff intervention, via counselling etc., to be able to gain support from other students experiencing similar issues. There is the potential for any products created within this project to be used in a live environment where staff and students would actively use the system.

The forum section of the app can also ease the strain that most support services are experiencing. As the general public are becoming more confident talking about mental health issues, many people are coming forward with various mental health issues.

### 1.2.3 Comparison of other possible solutions

A possible solution for this issue is a web based application that utilises HTML, PHP and SQL Databases. I have decided that this is not an appropriate solution as the system needs to be available to those who are not near a computer and using their mobile devices to access the internet. This can create an issue where data usage is limited therefore a solution that uses little to no mobile data is optimum for this project.

## 1.3 Chapter List

Provide a list of all the chapters within the report and a brief summary of the content. Ensure each summary avoids having a repetitive structure such as starting with “This chapter deals with”.

Chapter 2 Literature Review. This chapter reviews the Effectiveness of Social Media within Mental Health Services.

Chapter 3 Background Research. This chapter looks at the research methods used within this project.

Chapter 4 Requirements. This chapter defines the attainment criteria to evaluate this project and any products produced.

Chapter 5 Design. This chapter examines the design of the project based on the requirements.

Chapter 6 Implementation. This chapter shows how the designs will be implemented in a production environment.

Chapter 7 Testing. This chapter includes the testing sequences and plans created to evaluate the final product.

Chapter 8 Evaluation of Product. This chapter details the feedback received from people who have used the previous system and/or may use the new system once implemented.

Chapter 9 Evaluation of Process. This chapter analyses the processes within the project and evaluates the effectiveness of these processes.

Chapter 10 Conclusion. The conclusions of the report are presented. This chapter summarises the main positive outcomes and conclusions resulting from this body of work.

Chapter 2

# Literature Review

This chapter provides a comprehensive review of the Effectiveness of Social Media within Mental Health Services. Davis (2016) provides links to these and other resources.

## 2.1 Introduction

The growing use of Social Media within popular culture has shown that today’s society is more willing to share with others on a public forum their personal and social activities such as shopping, eating out and watching films.

Unfortunately, Mental Health Issues are rarely brought up online on social media platforms however many awareness groups, education institutions and healthcare providers around the world are starting to create and utilise the social media platforms to provide support and advice for those who require it. For example, the Birmingham City University Student’s Union (BCUSU) created and formed a mental health awareness society within the university. The Birmingham City University Mental Health Awareness (BCUMHA) Society’s mission is stated on Facebook.

“Raising awareness, tackling stigma, and enhancing the student experience for those who suffer from a mental health illness at Birmingham City University”

(Birmingham City University Mental Health Awareness et al., 2016)

Since the formation of BCUMHA and similar societies at other universities, students have been discussing and raising issues within their university and provided feedback on the support they have been offered.

Table 1 (shown below) contains all journal articles the author has selected surrounding the topic of the use of social media within mental health services.

|  |  |  |  |
| --- | --- | --- | --- |
| **Authors** | **Year** | **Paper Title** | **Paper Summary** |
| Moorhead et al | 2013 | A New Dimension Of Health Care: Systematic Review Of The Uses, Benefits, And Limitations Of Social Media For Health Communication | This paper researches into the uses and effects of using social media platforms to communicate between healthcare provider and patient. The paper identifies recommendations to enable general health communication to become more effective. |
| Hoge et al. | 2006 | Mental Health Problems, Use Of  Mental Health Services, And Attrition  From Military Service After Returning  From Deployment To Iraq Or Afghanistan | This paper looks into the mental health care provided to military personnel coming back from combat areas and how the use of mental health screening has allowed the care provided to be of a higher standard. |
| Maulik et al. | 2009 | The Role Of Social Network And Support In Mental Health Service  Use: Findings From The Baltimore ECA Study | This paper examine the possibility and the effects of using social media to treat mental health issues. |
| Orlowski et al. | 2016 | The Promise And The Reality: A Mental Health Workforce Perspective On Technology-Enhanced Youth Mental Health Service Delivery | Similar to the previous article, this paper examines the effects of using social media to treat mental health issues however this paper focuses on mental health issues faced by younger people and how social media can be used. |
| Rosenbaum et al. | 2012 | The Effect Of Instant Messaging Services On Society’s Mental Health | This paper shows how **instant messaging services** affect societal mental health within this 18-30 age group. It also looks into the issues of internet addition within this age group and its effect on their mental health. |
| Blanchard | 2011 | Navigating The Digital Disconnect | Blanchard looks into the strategies and their effectiveness of using new technologies to impact the mental health and wellbeing of young people aged 12 to 25. |
| Crown copyright | 1983 | The Mental Health Act 1983 | This legislation defines all circumstances for admission for mental health issues and guidelines and policies for the care of patients with mental health issues. |
| Great Britain. Department of Health | 1999 | Mental Health: Modern Standards And Service Models: Executive Summary | This book issues the standards that all NHS trusts within the United Kingdom should follow and adhere to. It includes guidelines on treatment and care. |

Table - An Overview of the Academic Literature Surrounding the Effectiveness of Social Media within Mental Health Services

## 2.2 The Deployment of Mental Health Services

Mental health services differ within each country; this critical literature review will focus mainly on the services provided within the United Kingdom and the United States of America.

### 2.2.1 Mental Health Services in the United States of America

Mental health services within the United States of America (USA) are provided by private companies who require payment from each recipient of their services.

The United States Military provides a service for personnel that assesses them for possible health issues created from working within the combat zones. This service begins with a form called the Post-Deployment Health Assessment form (PDHA). This allows the military to arrange and deliver specific healthcare treatments, including any mental health services to their staff. This service has a positive effect on the users of the service within the first year however statistics for after the first year of service are incomplete as stated in Hoge’s paper,

“This study provides new data showing the strong relationship between combat duty and a variety of mental health outcomes and most importantly high mental health care utilization in the first year after deployment. … Additional research is needed beyond a year after deployment to determine the long-term burden that this war will have on the mental health care system.”

(Hoge et al., 2006)

### 2.2.2 Mental Health Services in the United Kingdom

Mental health services within the United Kingdom is generally funded by the National Health Service (NHS). The NHS is currently aiming to cut down operational costs for these services. In 1999, the then Secretary of State for Health Frank Dobson MP, announced

“The Government is committed to do whatever is necessary to deliver a modern and dependable health service ... Mental health services and the professionals who provide them will get the attention and resources they deserve”

(Great Britain. Department of Health, 1999)

These commitments have not been fulfilled as the country has since experienced several recessions that have overshadowed the improvement to mental health services.

There are however stricter guidelines on providing efficient and quality mental health services. The Mental Health Act 1983 states all guidelines for the delivery and protection of mental health services.

“(1) It shall be an offence for any person who is an officer on the staff of or otherwise employed in…

to ill-treat or wilfully to neglect a patient for the time being receiving treatment for mental disorder as an in-patient …

or

(b) to ill-treat or wilfully to neglect, … a patient for the time being receiving such treatment there as an out-patient.

(2) It shall be an offence for any individual to ill-treat or wilfully to neglect a mentally disordered patient who is for the time being subject to his guardianship under this Act or otherwise in his custody or care.”

(Crown copyright, 1983)

## 2.3 The Concept of Social Media

Social media has become one of the human rights in some countries with many people who use the internet owning one or more social media accounts.

Social media platforms promote freedom of speech also as they are generally ungoverned and have in the past allowed those who are in oppressed countries to publicise the internal events going on.

### 2.3.1 Different Types of Social Media Platform

There are approximately 8 different social media platform categories; Social Networking, Video Sharing, Live-Streaming, Microblogging, Blogging, Social News, Photo Sharing and Content Curation. The most popular platforms for each category are Facebook, YouTube, Twitch, Twitter, WordPress, Reddit, Instagram and Pinterest respectively.

### 2.3.2 The Uses of Social Media

Many people well known within popular culture gain their popularity and ‘fame’ from using their social media accounts. The most famous case for this is Kim Kardashian who frequently publishes suggestive images of herself online to explicitly gain controversy.

Other users of social media platforms include tutorials, news articles, advertising, gameplay and general ideas for projects.

Many countries security services use these social networks to monitor possible security threats such as suspected terrorists. The United States of America’s National Security Agency (NSA) utilises social media to monitor security targets as Joseph Verble states in his paper.

“The NSA became the management system for the … researchers into new computer technology and communications infrastructure.”

(Verble, 2014)

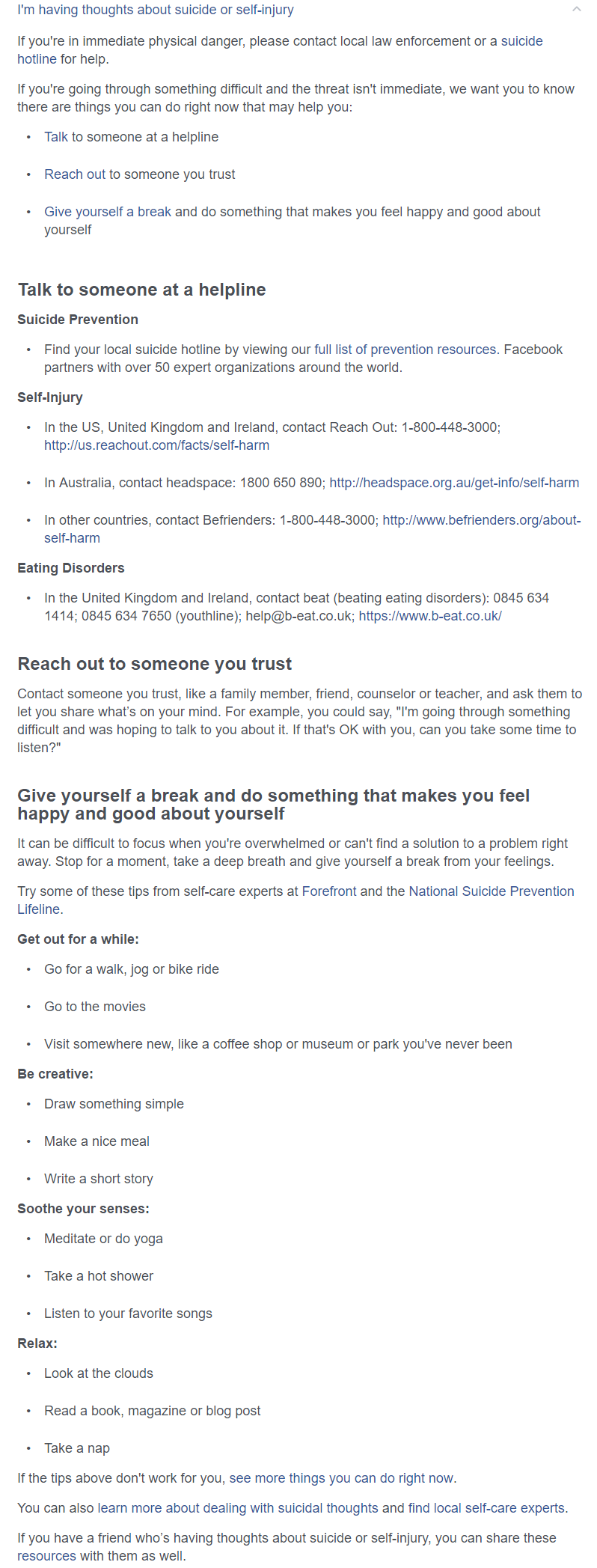
## 2.4 The Use of Social Media Within Mental Health Services

Recently, mental healthcare providers have been trialling the use of social media and online platforms for providing support and advice to their patients. Several journal articles suggest that online treatment would benefit however careful consideration on the design and use of the platforms should be observed. Orlowski states

“While the benefits of technology were seen in the ability to more closely track consumers’ progress and level of risk, it was also associated with concerns around an implied level of clinician responsivity and possible increases to workload. These concerns included a perceived lack of processing around information communicated via these modes of communication and unclear guidelines around when and how clinicians should respond to potentially risky information, particularly outside of work hours.”

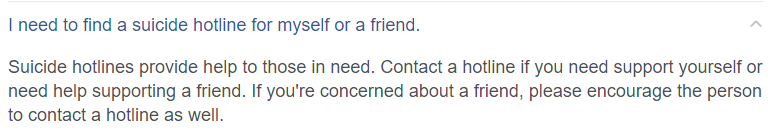
(Orlowski et al., 2016)

Social media can also be used by the general public to assess someone’s mental health although the system is heavily reliant on each person’s. This has already been implemented on Facebook where users can report those who are at risk of harming themselves or someone else. Figures 5, 6 and 7 show the advice and guidance Facebook provides its users who can either refer themselves or someone else to a mental health service.



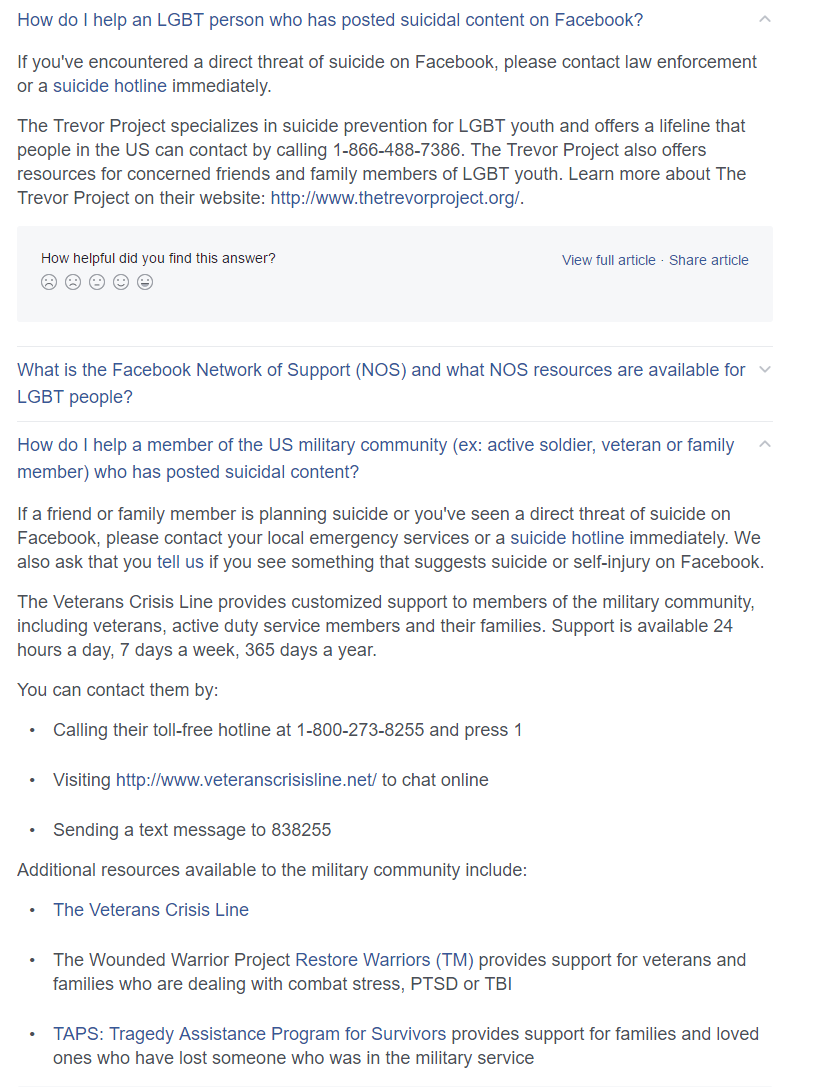
Source: (Facebook, 2017)

Figure – Question giving advice regarding mental health issues



Source: (Facebook, 2017)

Figure – Question giving advice regarding mental health issues



Source: (Facebook, 2017)

Figure – Question giving advice regarding mental health issues

Researchers have also found that the use of social media specifically the reduced contact between friends and family affected the odds of accessing mental health services. Maulik states

“The effect of social network and social support on general medical service use indicated that a reduced frequency of contact with either friends or relatives was associated with reduced odds of accessing services across different mental health conditions.”

(Maulik et al., 2009)

This demonstrates that mental healthcare providers should begin advertising and offering support on social media platforms as this would greatly increase the usage of their services and allow more flexibility with appointments. It would also ensure that people suffering with anxiety disorders are able to attend their appointments regardless of being in the provider’s facility or not.

The use of social media within healthcare can also be used within general healthcare also. The usage of services such as WebMD increase engagement and the awareness of other health conditions. Moorhead states

“Social media brings a new dimension to health care as it offers a medium to be used by the public, patients, and health professionals to communicate about health issues with the possibility of potentially improving health outcomes.”

(Moorhead et al., 2013)

### 2.4.1 Adoption within Education Institutions

Education institutions have been very hesitant to incorporate social media into their wellbeing programmes as there is evidence to suggest that students’ conditions may worsen if services are accessed using social media platforms. Rosenbaum demonstrates this saying

“research shows that internet addiction may exacerbate ADHD symptoms in adolescents”

(Rosenbaum et al., 2012)

Institutions may also be hesitant due to the cost of implementing such a service and developing procedures and training staff as demonstrated in Blanchard’s article.

“Information communication technologies have great potential in improving young people’s mental health ... However, if this potential is realized, guidelines for their safe and effective use need to be developed. Furthermore, investment needs to be made in securing appropriate technology infrastructure … and in training staff to better understand young people’s use of technology.”

(Blanchard, 2011)

## 2.5 Conclusion

The use of social media within mental healthcare allows more organisations to provide a more efficient service without affecting the quality at a lower cost compared to traditional methods such as face to face appointments.

The future of social media and mental health services is very unpredictable as many organisations are incredibly hesitant to start using social media platforms however as more digital devices including The Internet of Things devices become popular, the use of social media and mental health services may grow and become the main method for individuals to access these services.

Chapter 3

# Background Research

## 3.1 Research Subjects

To ensure that my results are unbiased results, all participants for my research and evaluation were students at a university level. This included people with and without mental health issues and many students who may not seek emergency assistance may use the product to assist someone who does require assistance.

## 3.2 Research Methods

Throughout the project, I have employed various techniques which enables me to gain the relevant knowledge and ability to create a specification that will suit all the needs for most if not all users. Due to the nature of my project and the app, I decided to employ the PICO assessment process for researching the relevant fields. The PICO process has the following elements as stated within the New York University library website:

“Evidence-based models use a process for framing a question, locating, assessing, evaluating, and repeating as needed. PICO (T) elements include: Problem/Patient/Population, Intervention/Indicator, Comparison, Outcome, and (optional) Time element or Type of Study.”

(Research Guides: Health (Nursing, Medicine, Allied Health): Search Strategies: Framing the question (PICO), 2017)

### 3.2.1 Interviews

As part of my research I interviewed many staff and students within Birmingham City University asking them of their experiences with the current system used within the university. I also asked each participant if they wanted to see the changes to the system if I were to only make modifications however there was a resounding request to rebuild the system from scratch due to the fact it was being integrated to a ‘semi-working’ system which did not suit the student’s needs.

#### 3.2.1.1 Time Trials

Part of my interviewing process. I timed selected students to access the original Mental Health and Wellbeing Counselling referral form and gave them a maximum of five minutes to complete the task. Out of the 8 students selected, 4 did not manage to complete the task within five minutes. The students who managed to complete the task within five minutes had a wide range of timings. For information about these timings please see Appendix D which contains the results of the student feedback sessions.

### 3.2.2 Analysis of Current System

## 3.3 Conclusion

Chapter 4

# Requirements

## 4.1 App Design Requirements

## 4.2 Platform Requirements

## 4.3 Accessibility Requirements

## 4.4 Time Restraints

## 4.5 Conclusions

The main conclusions for this chapter.

Chapter 5

# Design

## 5.1 App Workflow

## 5.2 App Design Criteria

## 5.3 Platform Decisions

## 5.4 Conclusion and Final Design

The main conclusions for this chapter.

Chapter 6

# Implementation

This chapter examines the implementation of the project.

## 6.1 Implementation Options

### 6.1.1 Whole System Implementation

### 6.1.2 System Integration

### 6.1.3 Phasing Implementation

## 6.2 Conclusions and Final Implementation Decisions

The main conclusions for this chapter.

Chapter 7

# Testing

This chapter evaluates the overall project and provides results of tests carried out.

## 7.1 Testing Techniques

## 7.2 Test Plans

## 7.3 Conclusions

The main conclusions for this chapter.

Chapter 8

# Evaluation of Product

This chapter evaluates the overall project and provides results of tests carried out.

## 8.1 Evaluation Techniques

### 8.1.1 Demonstrations

### 8.1.2 Mental Health Awareness Week Stands

### 8.1.3 App Questionnaires

## 8.2 Comment Reviews

## 8.3 Conclusions

The main conclusions for this chapter.

Chapter 9

# Evaluation of Process

## 9.1 Personal Evaluation

## 9.2 Possible Future Changes

## 9.1 Conclusions

The main conclusions for this chapter.

Chapter 10

# Conclusion

## 10.1 Conclusions

The main conclusions that may be drawn from the body of work.

## 10.2 Future Work

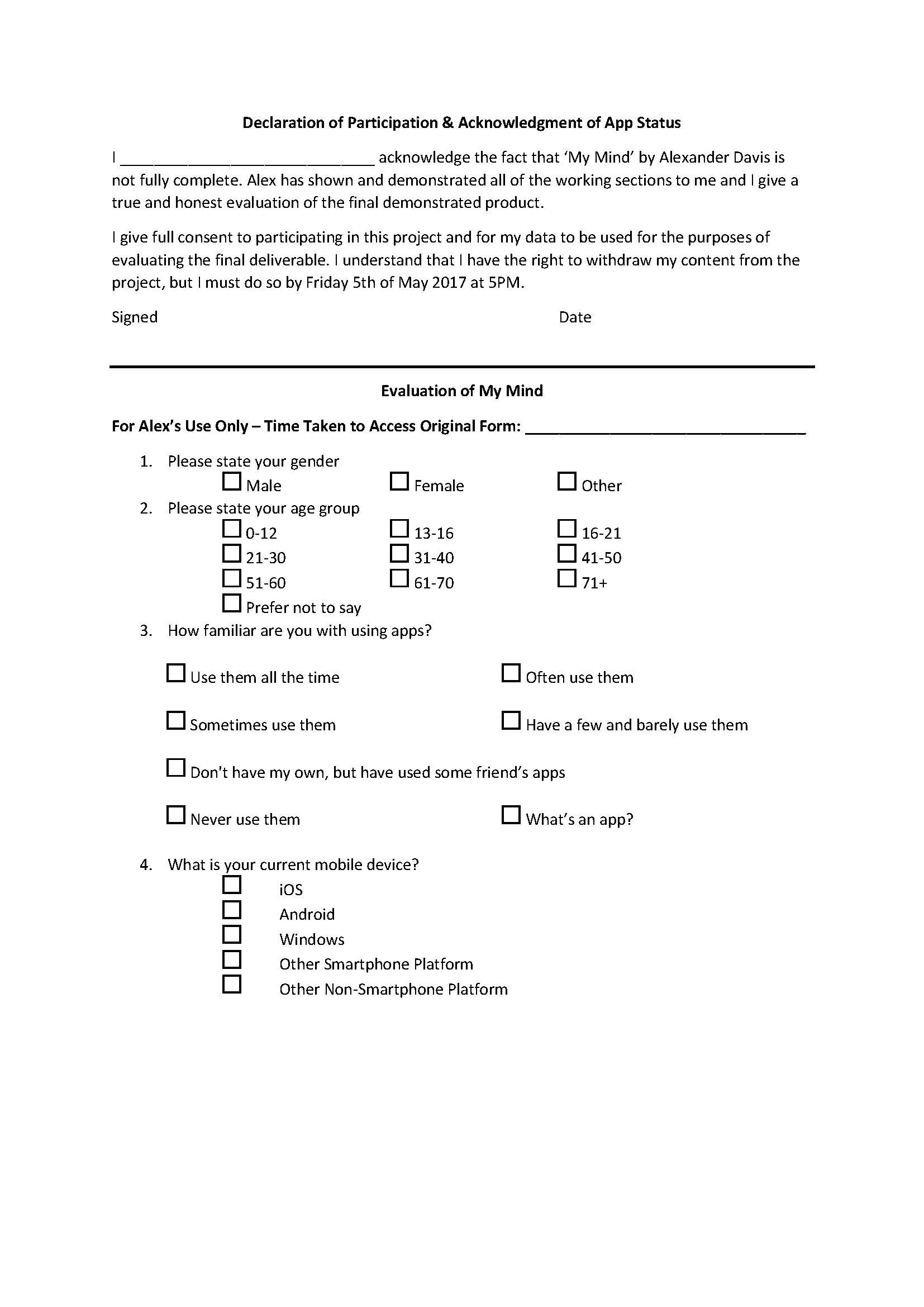
Further development that could be carried out in the future.

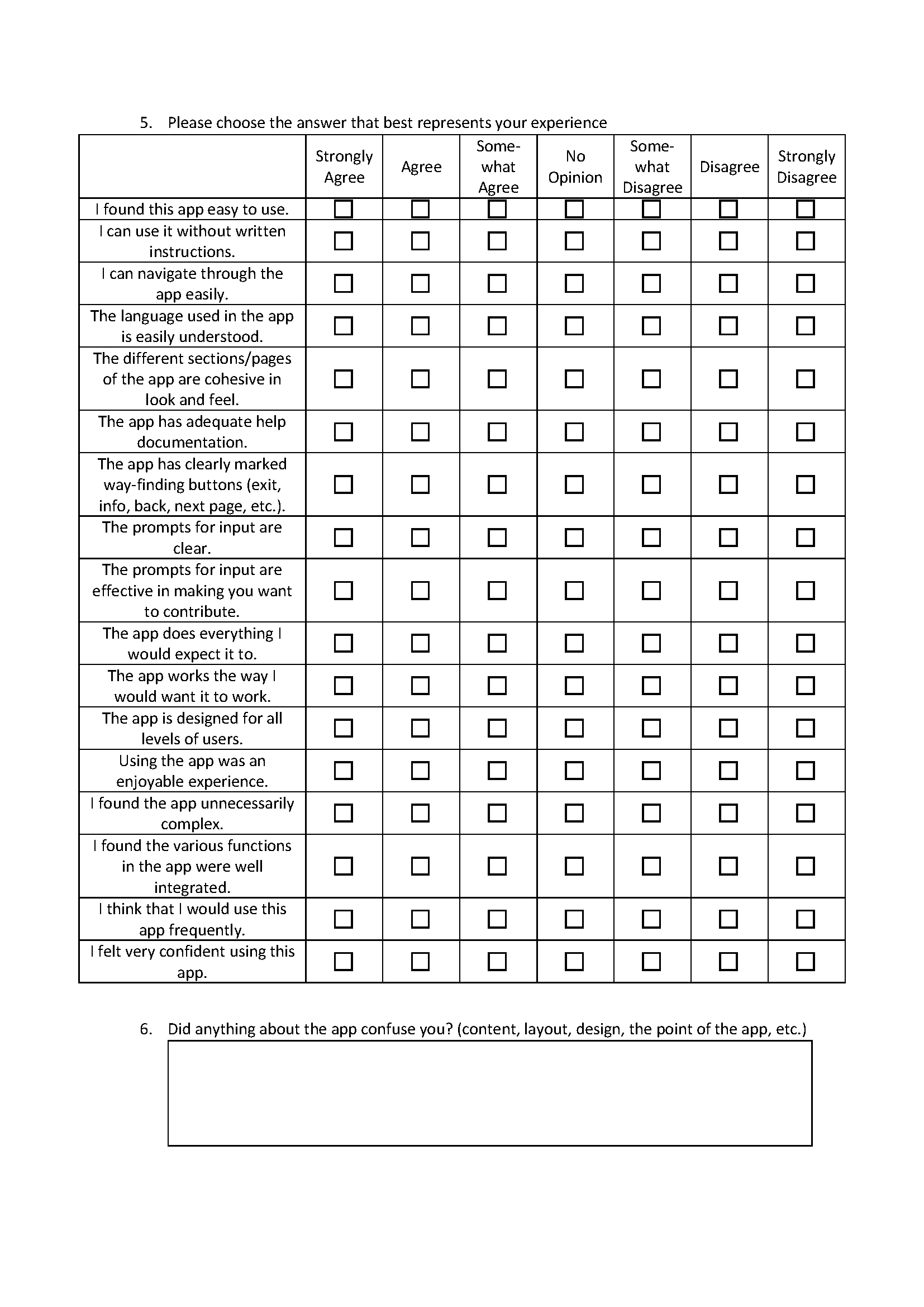
# References

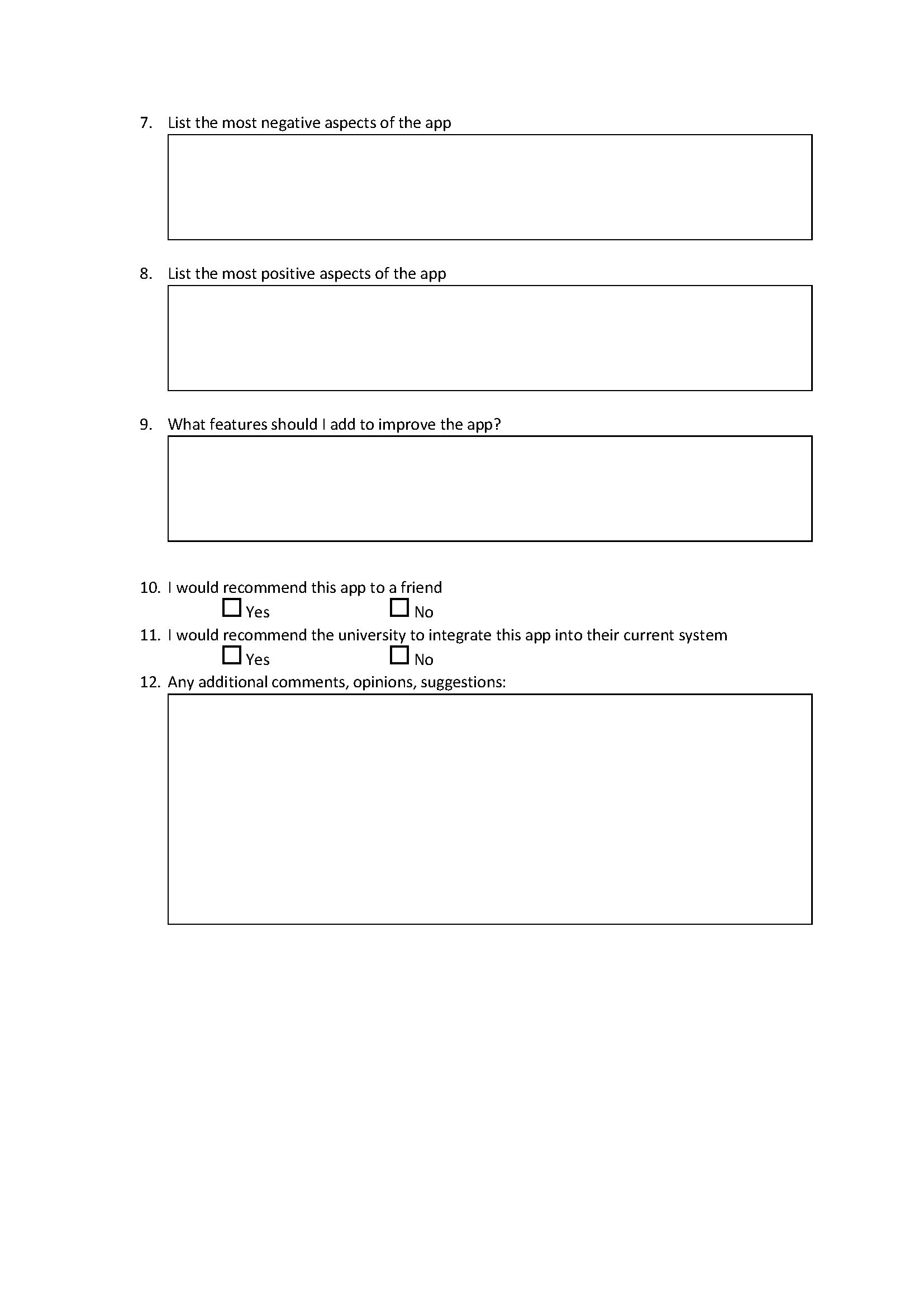
# Bibliography

# Appendix A – Sample Questionnaire & Consent Form

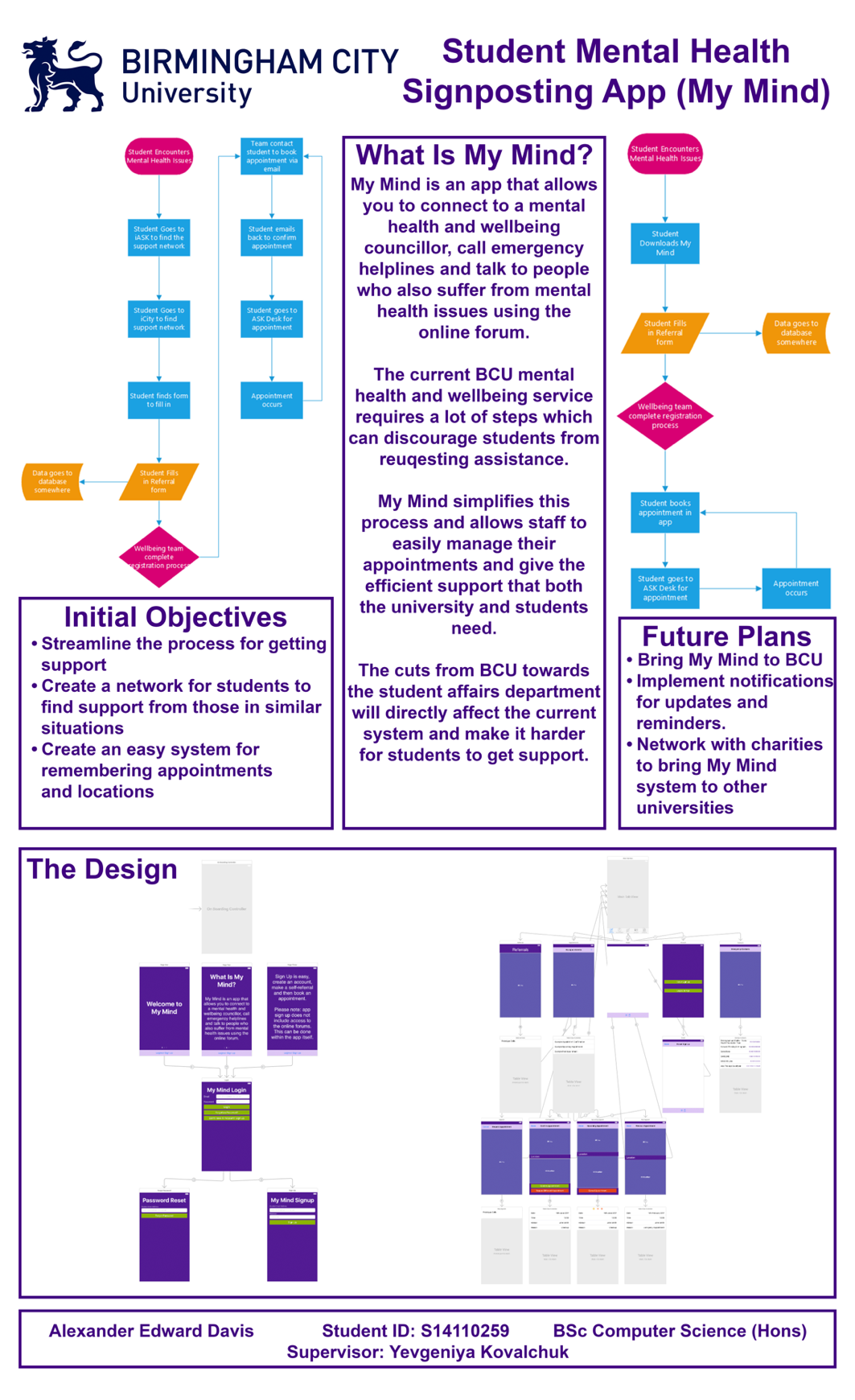
Below is the content of the questionnaire to evaluate my app and the consent form for participation within the project.







# Appendix B – Presentation



# Appendix C – Source Code

## AppDelegate

//

//  AppDelegate.swift

//  My Mind

//

//  Created by Alexander Davis on 09/04/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import UIKit

import Firebase

import UserNotifications

import NotificationCenter

import FirebaseMessaging

import FirebaseDatabase

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    let gcmMessageIDKey = "gcm.message\_id"

    func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {

        // Use Firebase library to configure APIs

        FIRApp.configure()

        FIRDatabase.database().persistenceEnabled = true

        //If User Is Signed In Direct to Main App Section

        let currentUser = FIRAuth.auth()?.currentUser

        let mainStoryboard = UIStoryboard(name: "Main", bundle: nil)

        if currentUser != nil

        {

            self.window?.rootViewController = mainStoryboard.instantiateViewController(withIdentifier: "MainController")

        }

        else

        {

            self.window?.rootViewController = mainStoryboard.instantiateViewController(withIdentifier: "OnBoard")

        }

        //Turn On Notifications for App

        if #available(iOS 10.0, \*) {

            // For iOS 10 display notification (sent via APNS)

            UNUserNotificationCenter.current().delegate = self

            let authOptions: UNAuthorizationOptions = [.alert, .badge, .sound]

            UNUserNotificationCenter.current().requestAuthorization(

                options: authOptions,

                completionHandler: {\_, \_ in })

            // For iOS 10 data message (sent via FCM)

            FIRMessaging.messaging().remoteMessageDelegate = self as? FIRMessagingDelegate

        } else {

            let settings: UIUserNotificationSettings =

                UIUserNotificationSettings(types: [.alert, .badge, .sound], categories: nil)

            application.registerUserNotificationSettings(settings)

        }

        application.registerForRemoteNotifications()

        return true

    }

    func applicationWillResignActive(\_ application: UIApplication) {

        // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.

        // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.

    }

    func applicationDidEnterBackground(\_ application: UIApplication) {

        // Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.

        // If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.

    }

    func applicationWillEnterForeground(\_ application: UIApplication) {

        // Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.

    }

    func applicationDidBecomeActive(\_ application: UIApplication) {

        // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.

    }

    func applicationWillTerminate(\_ application: UIApplication) {

        // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.

    }

    func application(\_ application: UIApplication, didReceiveRemoteNotification userInfo: [AnyHashable: Any]) {

        // If you are receiving a notification message while your app is in the background,

        // this callback will not be fired till the user taps on the notification launching the application.

        // TODO: Handle data of notification

        // Print message ID.

        if let messageID = userInfo[gcmMessageIDKey] {

            print("Message ID: \(messageID)")

        }

        // Print full message.

        print(userInfo)

    }

    func application(\_ application: UIApplication, didReceiveRemoteNotification userInfo: [AnyHashable: Any],

                     fetchCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {

        // If you are receiving a notification message while your app is in the background,

        // this callback will not be fired till the user taps on the notification launching the application.

        // TODO: Handle data of notification

        // Print message ID.

        if let messageID = userInfo[gcmMessageIDKey] {

            print("Message ID: \(messageID)")

        }

        // Print full message.

        print(userInfo)

        completionHandler(UIBackgroundFetchResult.newData)

    }

}

// [START ios\_10\_message\_handling]

@available(iOS 10, \*)

extension AppDelegate : UNUserNotificationCenterDelegate {

    // Receive displayed notifications for iOS 10 devices.

    func userNotificationCenter(\_ center: UNUserNotificationCenter,

                                willPresent notification: UNNotification,

                                withCompletionHandler completionHandler: @escaping (UNNotificationPresentationOptions) -> Void) {

        let userInfo = notification.request.content.userInfo

        // Print message ID.

        if let messageID = userInfo[gcmMessageIDKey] {

            print("Message ID: \(messageID)")

        }

        // Print full message.

        print(userInfo)

        // Change this to your preferred presentation option

        completionHandler([])

    }

    func userNotificationCenter(\_ center: UNUserNotificationCenter,

                                didReceive response: UNNotificationResponse,

                                withCompletionHandler completionHandler: @escaping () -> Void) {

        let userInfo = response.notification.request.content.userInfo

        // Print message ID.

        if let messageID = userInfo[gcmMessageIDKey] {

            print("Message ID: \(messageID)")

        }

        // Print full message.

        print(userInfo)

        completionHandler()

    }

}

// [END ios\_10\_message\_handling]

// [START ios\_10\_data\_message\_handling]

extension AppDelegate : FIRMessagingDelegate {

    // Receive data message on iOS 10 devices while app is in the foreground.

    func applicationReceivedRemoteMessage(\_ remoteMessage: FIRMessagingRemoteMessage) {

        print(remoteMessage.appData)

    }

}

// [END ios\_10\_data\_message\_handling]

## OnBoardingController

//

//  OnBoardingController.swift

//  My Mind

//

//  Created by Alexander Davis on 30/03/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

class OnBoardingController : UIPageViewController {

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

    override func viewDidLoad() {

        // Set the dataSource and delegate in code.

        // I can't figure out how to do this in the Storyboard!

        dataSource = self

        delegate = self

        // This is the starting point.  Start with step zero.

        setViewControllers([getPageOne()], direction: .forward, animated: false, completion: nil)

    }

    func getPageOne() -> PageOne {

        return storyboard!.instantiateViewController(withIdentifier: "WelcomeScreenOne") as! PageOne

    }

    func getPageTwo() -> PageTwo {

        return storyboard!.instantiateViewController(withIdentifier: "WelcomeScreenTwo") as! PageTwo

    }

    func getPageThree() -> PageThree {

        return storyboard!.instantiateViewController(withIdentifier: "WelcomeScreenThree") as! PageThree

    }

}

// MARK: - UIPageViewControllerDataSource methods

extension OnBoardingController : UIPageViewControllerDataSource {

    func pageViewController(\_ pageViewController: UIPageViewController, viewControllerBefore viewController: UIViewController) -> UIViewController? {

        if viewController.isKind(of: PageTwo.self) {

            // 2 -> 1

            return getPageTwo()

        } else if viewController.isKind(of: PageTwo.self) {

            // 1 -> 0

            return getPageOne()

        } else {

            // 0 -> end of the road

            return nil

        }

    }

    func pageViewController(\_ pageViewController: UIPageViewController, viewControllerAfter viewController: UIViewController) -> UIViewController? {

        if viewController.isKind(of: PageOne.self) {

            // 0 -> 1

            return getPageTwo()

        } else if viewController.isKind(of: PageTwo.self) {

            // 1 -> 2

            return getPageThree()

        } else {

            // 2 -> end of the road

            return nil

        }

    }

    // This only gets called once, when setViewControllers is called

    func presentationIndex(for pageViewController: UIPageViewController) -> Int {

        return 0

    }

}

// MARK: - UIPageViewControllerDelegate methods

extension OnBoardingController : UIPageViewControllerDelegate {

}

## Welcome Screens

### WelcomeScreenOne

//

//  WelcomeScreenOne.swift

//  My Mind

//

//  Created by Alexander Davis on 30/03/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

class PageOne : UIViewController {

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

### WelcomeScreenTwo

//

//  WelcomeScreenTwo.swift

//  My Mind

//

//  Created by Alexander Davis on 30/03/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

class PageTwo : UIViewController {

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

### WelcomeScreenThree

//

//  WelcomeScreenThree.swift

//  My Mind

//

//  Created by Alexander Davis on 30/03/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

class PageThree : UIViewController {

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

## Login Section and User Information

### UserViews

//

//  LoginView.swift

//  My Mind

//

//  Created by Alexander Davis on 31/03/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

import Firebase

import FirebaseAuth

class LoginView: UIViewController {

    @IBOutlet weak var Emailtbx: UITextField!

    @IBOutlet weak var Passwordtbx: UITextField!

    @IBAction func Loginbtn(\_ sender: UIButton) {

        if self.Emailtbx.text == "" || self.Passwordtbx.text == "" {

            //Alert to tell the user that there was an error because they didn't fill anything in the textfields because they didn't fill anything in

            let alertController = UIAlertController(title: "Error", message: "Please enter an email and password.", preferredStyle: .alert)

            let defaultAction = UIAlertAction(title: "OK", style: .cancel, handler: nil)

            alertController.addAction(defaultAction)

            self.present(alertController, animated: true, completion: nil)

        } else {

            FIRAuth.auth()?.signIn(withEmail: self.Emailtbx.text!, password: self.Passwordtbx.text!) { (user, error) in

                if error == nil {

                    //Print into the console if successfully logged in

                    print("You have successfully logged in")

                    //Go to the MainController if the login is sucessful

                    let alertController = UIAlertController(title: "Welcome", message: "You have successfully logged in", preferredStyle: .alert)

                    let vc = self.storyboard?.instantiateViewController(withIdentifier: "MainController")

                    let defaultAction = UIAlertAction(title: "OK", style: .default, handler: { action in self.present(vc!, animated: true, completion: nil);})

                    alertController.addAction(defaultAction)

                    self.present(alertController, animated: true, completion: nil)

                } else {

                    //Tells the user that there is an error and then gets firebase to tell them the error

                    let alertController = UIAlertController(title: "Error", message: error?.localizedDescription, preferredStyle: .alert)

                    let defaultAction = UIAlertAction(title: "OK", style: .cancel, handler: nil)

                    alertController.addAction(defaultAction)

                    self.present(alertController, animated: true, completion: nil)

                }

            }

        }

    }

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

 }

class SignUp : UIViewController {

    @IBOutlet weak var Emailtbx: UITextField!

    @IBOutlet weak var Passwordtbx: UITextField!

    @IBAction func createAccountbtn(\_ sender: UIButton) {

        if Emailtbx.text == "" {

            let alertController = UIAlertController(title: "Error", message: "Please enter your email and password", preferredStyle: .alert)

            let defaultAction = UIAlertAction(title: "OK", style: .cancel, handler: nil)

            alertController.addAction(defaultAction)

            present(alertController, animated: true, completion: nil)

        } else {

            FIRAuth.auth()?.createUser(withEmail: Emailtbx.text!, password: Passwordtbx.text!) { (user, error) in

                if error == nil {

                    let alertController = UIAlertController(title: "Sign Up Complete", message: "You have Successfully Signed Up", preferredStyle: .alert)

                    let vc = self.storyboard?.instantiateViewController(withIdentifier: "Login")

                    let defaultAction = UIAlertAction(title: "OK", style: .default, handler: { action in self.present(vc!, animated: true, completion: nil);})

                    alertController.addAction(defaultAction)

                    self.present(alertController, animated: true, completion: nil)

                } else {

                    let alertController = UIAlertController(title: "Error", message: error?.localizedDescription, preferredStyle: .alert)

                    let defaultAction = UIAlertAction(title: "OK", style: .cancel, handler: nil)

                    alertController.addAction(defaultAction)

                    self.present(alertController, animated: true, completion: nil)

                }

            }

        }

    }

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

class ForgotPass : UIViewController {

    @IBOutlet weak var Emailtbx: UITextField!

    @IBAction func Resetbtn(\_ sender: UIButton) {

        if self.Emailtbx.text == "" {

            let alertController = UIAlertController(title: "Oops!", message: "Please enter an email.", preferredStyle: .alert)

            let defaultAction = UIAlertAction(title: "OK", style: .cancel, handler: nil)

            alertController.addAction(defaultAction)

            present(alertController, animated: true, completion: nil)

        } else {

            FIRAuth.auth()?.sendPasswordReset(withEmail: self.Emailtbx.text!, completion: { (error) in

                var title = ""

                var message = ""

                if error != nil {

                    title = "Error!"

                    message = (error?.localizedDescription)!

                } else {

                    title = "Success!"

                    message = "Password reset email sent."

                    self.Emailtbx.text = ""

                }

                let alertController = UIAlertController(title: title, message: message, preferredStyle: .alert)

                let vc = self.storyboard?.instantiateViewController(withIdentifier: "Login")

                let defaultAction = UIAlertAction(title: "OK", style: .cancel, handler: { action in self.present(vc!, animated: true, completion: nil);})

                alertController.addAction(defaultAction)

                self.present(alertController, animated: true, completion: nil)

            })

        }

    }

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

class Logout : UIViewController {

    @IBAction func Logoutbtn(\_ sender: UIButton) {

        if FIRAuth.auth()?.currentUser != nil {

            do {

                try FIRAuth.auth()?.signOut()

                let vc = UIStoryboard(name: "Main", bundle: nil).instantiateViewController(withIdentifier: "OnBoard")

                present(vc, animated: true, completion: nil)

            } catch let error as NSError {

                print(error.localizedDescription)

            }

        }

    }

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

## Main Tab Views

### TabBarController

//

//  TabBarController.swift

//  My Mind

//

//  Created by Alexander Davis on 05/04/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

class MainTabView: UITabBarController{

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

### MainViews

//

//  MainViews.swift

//  My Mind

//

//  Created by Alexander Davis on 09/04/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

import Firebase

import FirebaseDatabase

class Referrals: UIViewController{

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

class ReferralConf: UIViewController{

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

class Appoint: UIViewController{

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

class Contact: UIViewController{

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

}

class EmergeContacts: UITableViewController{

    @IBAction func TrustCallBtn(\_ sender: UIButton) {

        if let url = URL(string: "telprompt://01213010000")

        {

            UIApplication.shared.open(url, options: [:], completionHandler: nil)

        }

    }

    @IBAction func FwdThinkingbtn(\_ sender: UIButton) {

        if let url = URL(string: "telprompt://03003000099")

        {

            UIApplication.shared.open(url, options: [:], completionHandler: nil)

        }

    }

    @IBAction func Samaritansbtn(\_ sender: UIButton) {

        if let url = URL(string: "telprompt://08457909090")

        {

            UIApplication.shared.open(url, options: [:], completionHandler: nil)

        }

    }

    @IBAction func Sanebtn(\_ sender: UIButton) {

        if let url = URL(string: "telprompt://08457678000")

        {

            UIApplication.shared.open(url, options: [:], completionHandler: nil)

        }

    }

    @IBAction func Mindbtn(\_ sender: UIButton) {

        if let url = URL(string: "telprompt://0300123393")

        {

            UIApplication.shared.open(url, options: [:], completionHandler: nil)

        }

    }

    @IBAction func Alexsbtn(\_ sender: UIButton) {

        if let url = URL(string: "telprompt://07871778000")

        {

            UIApplication.shared.open(url, options: [:], completionHandler: nil)

        }

    }

}

### FormViews

//

//  FormViews.swift

//  My Mind

//

//  Created by Alexander Davis on 11/04/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

import SwiftForms

import MessageUI

import MapKit

class ReferralForm: FormViewController, MFMailComposeViewControllerDelegate{

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

    struct Static {

        static let nameTag = "name"

        static let passwordTag = "password"

        static let lastNameTag = "lastName"

        static let addressTag = "address"

        static let IDTag = "idnumber"

        static let GenderTag = "gender"

        static let personalEmailTag = "personalemail"

        static let uniEmailTag = "universityemail"

        static let CourseTag = "course"

        static let phoneTag = "phone"

        static let Voicemail = "voicemail"

        static let DSATag = "disabledstudent"

        static let FacultyTag = "faculty"

        static let birthday = "birthday"

        static let PastSupportTag = "pastsupport"

        static let CurrentSupportTag = "currentsupport"

        static let subjectTag = "subject"

        static let timescaleTag = "timescale"

        static let button = "button"

    }

    required init(coder aDecoder: NSCoder) {

        super.init(coder: aDecoder)

        self.loadForm()

    }

    override func viewDidLoad() {

        super.viewDidLoad()

        //self.navigationItem.rightBarButtonItem = UIBarButtonItem(title: "Submit", style: .plain, target: self, action: #selector(ReferralForm.submit(\_:)))

    }

    // MARK: Actions

    func submit() {

        let data = self.form.formValues().description

        let alertController = UIAlertController(title: "Form Completed", message: "This form has sucessfully been submitted", preferredStyle: .alert)

        let vc = self.storyboard?.instantiateViewController(withIdentifier: "MainController")

        let defaultAction = UIAlertAction(title: "OK", style: .default, handler: { action in SendRef(input: data);self.tabBarController!.tabBar.items?[0].isEnabled = false;self.present(vc!, animated: true, completion: nil);})

        alertController.addAction(defaultAction)

        self.present(alertController, animated: true, completion: nil)

    }

    // MARK: Private interface

    fileprivate func loadForm() {

        let form = FormDescriptor(title: "Referral Form")

        let section1 = FormSectionDescriptor(headerTitle: nil, footerTitle: nil)

        var row = FormRowDescriptor(tag: Static.button, type: .button, title: "Send Form")

        row.configuration.button.didSelectClosure = { \_ in

            self.view.endEditing(true)

            self.submit()

        }

        section1.rows.append(row)

        let section2 = FormSectionDescriptor(headerTitle: "About Yourself", footerTitle: nil)

        row = FormRowDescriptor(tag: Static.nameTag, type: .name, title: "First Name")

        row.configuration.cell.appearance = ["textField.placeholder" : "First Name" as AnyObject, "textField.textAlignment" : NSTextAlignment.right.rawValue as AnyObject]

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.lastNameTag, type: .name, title: "Last Name")

        row.configuration.cell.appearance = ["textField.placeholder" : "Last name" as AnyObject, "textField.textAlignment" : NSTextAlignment.right.rawValue as AnyObject]

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.IDTag, type: .number, title: "Student ID")

        row.configuration.cell.appearance = ["textField.placeholder" : "12345678" as AnyObject, "textField.textAlignment" : NSTextAlignment.right.rawValue as AnyObject]

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.birthday, type: .date, title: "Date of Birth")

        row.configuration.cell.showsInputToolbar = true

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.uniEmailTag, type: .email, title: "University Email")

        row.configuration.cell.appearance = ["textField.placeholder" : "john.smith@mail.bcu.ac.uk" as AnyObject, "textField.textAlignment" : NSTextAlignment.right.rawValue as AnyObject]

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.personalEmailTag, type: .email, title: "Alternative Email")

        row.configuration.cell.appearance = ["textField.placeholder" : "john.smith@outlook.com" as AnyObject, "textField.textAlignment" : NSTextAlignment.right.rawValue as AnyObject]

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.phoneTag, type: .phone, title: "Phone")

        row.configuration.cell.appearance = ["textField.placeholder" : "Mobile Number" as AnyObject, "textField.textAlignment" : NSTextAlignment.right.rawValue as AnyObject]

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.Voicemail, type: .booleanSwitch, title: "Can we leave a voicemail message?")

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.GenderTag, type: .picker, title: "Gender")

        row.configuration.cell.showsInputToolbar = true

        row.configuration.selection.options = (["F", "M", "O", "U"] as [String]) as [AnyObject]

        row.configuration.selection.optionTitleClosure = { value in

            guard let option = value as? String else { return "" }

            switch option {

            case "F":

                return "Female"

            case "M":

                return "Male"

            case "O":

                return "Other"

            case "U":

                return "I'd rather not to say"

            default:

                return ""

            }

        }

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.addressTag, type: .multilineText, title: "Address")

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.DSATag, type: .booleanSwitch, title: "Are you a disabled student (eligible for DSA)?")

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.CourseTag, type: .text, title: "Course")

        row.configuration.cell.appearance = ["textField.placeholder" : "BSc Computer Science (Hons)" as AnyObject, "textField.textAlignment" : NSTextAlignment.right.rawValue as AnyObject]

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.FacultyTag, type: .picker, title: "Faculty")

        row.configuration.cell.showsInputToolbar = true

        row.configuration.selection.options = (["CEBE", "HELS", "ADM", "BLSS"] as [String]) as [AnyObject]

        row.configuration.selection.optionTitleClosure = { value in

            guard let option = value as? String else { return "" }

            switch option {

            case "CEBE":

                return "Computing, Engineering and The Built Environment"

            case "HELS":

                return "Health, Education and Life Sciences"

            case "ADM":

                return "Arts, Design and Media"

            case "BLSS":

                return "Business, Law and Social Sciences"

            default:

                return ""

            }

        }

        section2.rows.append(row)

        let section3 = FormSectionDescriptor(headerTitle: "What you would like to talk about.", footerTitle: nil)

        row = FormRowDescriptor(tag: Static.subjectTag, type: .multilineText, title: "")

        section3.rows.append(row)

        row = FormRowDescriptor(tag: Static.timescaleTag, type: .picker, title: "How long have you been concerned?")

        row.configuration.cell.showsInputToolbar = true

        row.configuration.selection.options = (["NL", "ST", "QL", "VL"] as [String]) as [AnyObject]

        row.configuration.selection.optionTitleClosure = { value in

            guard let option = value as? String else { return "" }

            switch option {

            case "NL":

                return "Not Long (within 1 month)"

            case "ST":

                return "Some Time (1-3 months)"

            case "QL":

                return "Quite Some Time (3-6 months)"

            case "VL":

                return "A Long Time (6+ months)"

            default:

                return ""

            }

        }

        section3.rows.append(row)

        row = FormRowDescriptor(tag: Static.CurrentSupportTag, type: .multipleSelector, title: "Are you currently receiving support?")

        row.configuration.cell.showsInputToolbar = true

        row.configuration.selection.options = (["No", "CBT", "CPN", "CP", "GP", "MHWA", "Psyi", "Psyo", "SW", "O"] as [String]) as [AnyObject]

        row.configuration.selection.optionTitleClosure = { value in

            guard let option = value as? String else { return "" }

            switch option {

            case "No":

                return "No"

            case "CBT":

                return "CBT Practitioner (e.g. Healthy Minds)"

            case "CPN":

                return "Community Psychiatric Nurse (CPN)"

            case "CP":

                return "Counsellor/Psychotherapist"

            case "GP":

                return "GP"

            case "MHWA":

                return "Mental Health/Wellbeing Adviser"

            case "Psyi":

                return "Psychiatrist"

            case "Psyo":

                return "Psychologist"

            case "SW":

                return "Social Worker"

            case "O":

                return "Other"

            default:

                return ""

            }

        }

        section3.rows.append(row)

        row = FormRowDescriptor(tag: Static.PastSupportTag, type: .multipleSelector, title: "Have you received support in the past?")

        row.configuration.cell.showsInputToolbar = true

        row.configuration.selection.options = (["No", "CBT", "CPN", "CP", "GP", "MHWA", "Psyi", "Psyo", "SW", "O"] as [String]) as [AnyObject]

        row.configuration.selection.optionTitleClosure = { value in

            guard let option = value as? String else { return "" }

            switch option {

            case "No":

                return "No"

            case "CBT":

                return "CBT Practitioner (e.g. Healthy Minds)"

            case "CPN":

                return "Community Psychiatric Nurse (CPN)"

            case "CP":

                return "Counsellor/Psychotherapist"

            case "GP":

                return "GP"

            case "MHWA":

                return "Mental Health/Wellbeing Adviser"

            case "Psyi":

                return "Psychiatrist"

            case "Psyo":

                return "Psychologist"

            case "SW":

                return "Social Worker"

            case "O":

                return "Other"

            default:

                return ""

            }

        }

        section3.rows.append(row)

        form.sections = [section1, section2, section3]

        self.form = form

    }

}

class ReqAppoint: FormViewController, MFMailComposeViewControllerDelegate{

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

    struct Static {

        static let date = "date"

        static let time = "time"

        static let location = "location"

        static let phoneTag = "phone"

        static let adivsor = "advisor"

        static let reason = "reason"

        static let button = "button"

    }

    required init(coder aDecoder: NSCoder) {

        super.init(coder: aDecoder)

        self.loadForm()

    }

    override func viewDidLoad() {

        super.viewDidLoad()

        //self.navigationItem.rightBarButtonItem = UIBarButtonItem(title: "Submit", style: .plain, target: self, action: #selector(ReferralForm.submit(\_:)))

    }

    // MARK: Actions

    func submit() {

        let data = self.form.formValues().description

        let alertController = UIAlertController(title: "Request Sent", message: "Your Request Has Been Sent", preferredStyle: .alert)

        let vc = self.storyboard?.instantiateViewController(withIdentifier: "MainController")

        let defaultAction = UIAlertAction(title: "OK", style: .default, handler: { action in SendRef(input: data);self.tabBarController!.tabBar.items?[0].isEnabled = false;self.present(vc!, animated: true, completion: nil);})

        alertController.addAction(defaultAction)

        self.present(alertController, animated: true, completion: nil)

    }

    // MARK: Private interface

    fileprivate func loadForm() {

        let form = FormDescriptor(title: "Referral Form")

        let section1 = FormSectionDescriptor(headerTitle: nil, footerTitle: nil)

        var row = FormRowDescriptor(tag: Static.button, type: .button, title: "Send Form")

        row.configuration.button.didSelectClosure = { \_ in

            self.view.endEditing(true)

            self.submit()

        }

        section1.rows.append(row)

        let section2 = FormSectionDescriptor(headerTitle: nil, footerTitle: nil)

        row = FormRowDescriptor(tag: Static.date, type: .date, title: "Date Requested")

        row.configuration.cell.showsInputToolbar = true

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.time, type: .time, title: "Time Requested")

        row.configuration.cell.showsInputToolbar = true

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.location, type: .picker, title: "Location")

        row.configuration.cell.showsInputToolbar = true

        row.configuration.selection.options = (["Centre", "North", "South", "Cons", "Margaret", "Jew","Bour"] as [String]) as [AnyObject]

        row.configuration.selection.optionTitleClosure = { value in

            guard let option = value as? String else { return "" }

            switch option {

            case "Centre":

                return "City Centre (Curzon Building)"

            case "North":

                return "City North (Baker Building)"

            case "South":

                return "City South (Seacole Building)"

            case "Cons":

                return "The Birmingham Conservatoire"

            case "Margaret":

                return "Margaret Street"

            case "Jew":

                return "Victoria Street"

            case "Bour":

                return "Bournville Campus"

            default:

                return ""

            }

        }

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.phoneTag, type: .phone, title: "Phone")

        row.configuration.cell.appearance = ["textField.placeholder" : "Mobile Number" as AnyObject, "textField.textAlignment" : NSTextAlignment.right.rawValue as AnyObject]

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.adivsor, type: .picker, title: "Preferred Advisor")

        row.configuration.cell.showsInputToolbar = true

        row.configuration.selection.options = (["JS", "AD", "TM", "U"] as [String]) as [AnyObject]

        row.configuration.selection.optionTitleClosure = { value in

            guard let option = value as? String else { return "" }

            switch option {

            case "JS":

                return "John Smith"

            case "AD":

                return "Alexander Davis"

            case "TM":

                return "Tim Minchin"

            case "U":

                return "I Don't Mind"

            default:

                return ""

            }

        }

        section2.rows.append(row)

        row = FormRowDescriptor(tag: Static.reason, type: .multilineText, title: "Reason for Appointment")

        section2.rows.append(row)

        form.sections = [section1, section2]

        self.form = form

    }

}

class ConfAppoint: UIViewController{

    @IBOutlet var ConfirmMap: MKMapView!

    override func viewDidLoad() {

        let initalLocation = CLLocation(latitude: 52.483358, longitude: -1.883024)

        centerMapOnLocation(location: initalLocation)

    }

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

    let regionRadius: CLLocationDistance = 1000

    func centerMapOnLocation(location: CLLocation) {

        let coordinateRegion = MKCoordinateRegionMakeWithDistance(location.coordinate,

                                                                  regionRadius \* 2.0, regionRadius \* 2.0)

        ConfirmMap.setRegion(coordinateRegion, animated: true)

    }

    @IBAction func Confirm(\_ sender: UIButton) {

        let alertController = UIAlertController(title: "Appointment Confirmed", message: "Your appointment has been confirmed.", preferredStyle: .alert)

        let vc = self.storyboard?.instantiateViewController(withIdentifier: "MainController")

        let defaultAction = UIAlertAction(title: "OK", style: .default, handler: { action in self.present(vc!, animated: true, completion: nil);})

        alertController.addAction(defaultAction)

        self.present(alertController, animated: true, completion: nil)

    }

    @IBAction func RequestAptbtn(\_ sender: UIButton) {

        let alertController = UIAlertController(title: "New Request Made", message: "A new appointment request has been made for you.", preferredStyle: .alert)

        let vc = self.storyboard?.instantiateViewController(withIdentifier: "MainController")

        let defaultAction = UIAlertAction(title: "OK", style: .default, handler: { action in self.present(vc!, animated: true, completion: nil);})

        alertController.addAction(defaultAction)

        self.present(alertController, animated: true, completion: nil)

    }

}

class UpcomingAppoint: UIViewController{

    @IBOutlet var UpcomingMap: MKMapView!

    override func viewDidLoad() {

        let initalLocation = CLLocation(latitude: 52.483358, longitude: -1.883024)

        centerMapOnLocation(location: initalLocation)

    }

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

    let regionRadius: CLLocationDistance = 1000

    func centerMapOnLocation(location: CLLocation) {

        let coordinateRegion = MKCoordinateRegionMakeWithDistance(location.coordinate,

                                                                  regionRadius \* 2.0, regionRadius \* 2.0)

        UpcomingMap.setRegion(coordinateRegion, animated: true)

    }

    @IBAction func Cancelbtn(\_ sender: UIButton) {

        let alertController = UIAlertController(title: "Appointment Cancelled", message: "Your appointment has been cancelled.", preferredStyle: .alert)

        let vc = self.storyboard?.instantiateViewController(withIdentifier: "MainController")

        let defaultAction = UIAlertAction(title: "OK", style: .default, handler: { action in self.present(vc!, animated: true, completion: nil);})

        alertController.addAction(defaultAction)

        self.present(alertController, animated: true, completion: nil)

    }

}

class PastAppoint: UIViewController{

    override func viewDidLoad() {

        let initalLocation = CLLocation(latitude: 52.483358, longitude: -1.883024)

        centerMapOnLocation(location: initalLocation)

    }

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

    @IBOutlet var PastMap: MKMapView!

    let regionRadius: CLLocationDistance = 1000

    func centerMapOnLocation(location: CLLocation) {

        let coordinateRegion = MKCoordinateRegionMakeWithDistance(location.coordinate,

                                                                  regionRadius \* 2.0, regionRadius \* 2.0)

        PastMap.setRegion(coordinateRegion, animated: true)

    }

}

### ForumView

//

//  ForumView.swift

//  My Mind

//

//  Created by Alexander Davis on 31/03/2017.

//  Copyright © 2017 Alexander Davis. All rights reserved.

//

import Foundation

import UIKit

class ForumViewController: UIViewController {

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

    @IBOutlet var webView: UIWebView!

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view, typically from a nib.

        let url = URL (string: "https://www.mymindforum.com");

        let requestObj = URLRequest(url: url!);

        webView.loadRequest(requestObj);

    }

    override func didReceiveMemoryWarning() {

        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

}

class ForumSignUpViewController: UIViewController {

    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent

    }

    @IBOutlet weak var ForumSignUp: UIWebView!

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view, typically from a nib.

        let url = URL (string: "https://www.mymindforum.com/ucp.php?mode=register");

        let requestObj = URLRequest(url: url!);

        ForumSignUp.loadRequest(requestObj);

    }

    override func didReceiveMemoryWarning() {

        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

}

# Appendix D – Student Feedback Results

## Student Information and Groupings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Age Group | Gender | Time Trial Exercise | Familiarity with Apps | Mobile Device Type |
| Participant A | 16-21 | Female | 4:12 | Use them all the time | iOS |
| Participant B | 22-30 | Male | 0:25 | Use them all the time | iOS |
| Participant C | 16-21 | Male | 03:29 | Use them all the time | Android |
| Participant D | 22-30 | Male | DNF | Use them all the time | iOS |
| Participant E | 16-21 | Male | DNF | Use them all the time | Other Non-Smartphone Device |
| Participant F | 16-21 | Female | DNF | Use them all the time | Android |
| Participant G | 21-30 | Female | DNF | Use them all the time | iOS |

## Participant Answers

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Strongly Agree | Agree | Some- what Agree | No Opinion | Some- what Disagree | Disagree | Strongly Disagree |
| I found this app easy to use. | 5 | 2 |  |  |  |  |  |
| I can use it without written instructions. | 5 | 1 | 1 |  |  |  |  |
| I can navigate through the app easily. | 5 | 2 |  |  |  |  |  |
| The language used in the app is easily understood. | 5 | 2 |  |  |  |  |  |
| The different sections/pages of the app are cohesive in look and feel. | 6 | 1 |  |  |  |  |  |
| The app has adequate help documentation. | 1 | 2 | 1 | 3 |  |  |  |
| The app has clearly marked way-finding buttons (exit, info, back, next page, etc.). | 1 | 6 |  |  |  |  |  |
| The prompts for input are clear. | 3 | 4 |  |  |  |  |  |
| The prompts for input are effective in making you want to contribute. | 5 |  | 2 |  |  |  |  |
| The app does everything I would expect it to. | 4 | 2 | 1 |  |  |  |  |
| The app works the way I would want it to work. | 4 | 2 |  | 1 |  |  |  |
| The app is designed for all levels of users. | 4 | 1 | 2 |  |  |  |  |
| Using the app was an enjoyable experience. | 3 | 2 |  | 2 |  |  |  |
| I found the app unnecessarily complex. |  |  |  |  |  | 5 | 2 |
| I found the various functions in the app were well integrated. | 3 | 3 | 1 |  |  |  |  |
| I think that I would use this app frequently. | 1 | 1 | 3 | 1 |  | 1 |  |
| I felt very confident using this app. | 4 | 3 |  |  |  |  |  |

## Open Ended Question Feedback

### Did Anything in The App Confuse You

* Struggled to find the “What do you want to talk about” box – could be more obvious.
* How to add/request an appointment.
* Who sees what data – data protection information.

### Negative Aspects

* Too much text
* App seems to be a bit too busy
* Spelling Mistakes
* Constantly jumps back to initial page when selecting back button
* No help documentation

### Positive Aspects

* Colour scheme
* Easy navigation
* Useful features overall
* Intuitive
* Emergency contacts
  + Feature highlighted twice by different participants
* Appointment creation/request tool
* Overall system easier to use in comparison to current system
* Name of app does not create a stigma

### Feature Requests

* Instant messaging feature
* Instead of building location, have a room location map
* Touch ID Protection
* Reminders/notifications of appointments the day/hour before perhaps an option to set a reminder

## Overall Feedback

## 

## 

# Temp new logoAppendix E - UCEEL Copyright Waiver

**Student Name: \_Alexander Edward Davis\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project/Thesis Title \_Student Mental Health Signposting App (My Mind) \_\_\_\_\_**

**Course: \_BSc Computer Science with Honours\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Student Agreement**

1. I confirm that Birmingham City University can electronically archive and make accessible the project / thesis described above via the UCEEL Electronic Library system. I retain all other ownership rights to the copyright of the document / project work described above.

2. I confirm the above project / thesis is a true and unaltered representation of the project / thesis as submitted to Birmingham City University course tutors and examiners.

3. I confirm that the above project / thesis material copied from a source (e.g. a book) where ownership of the copyright does not belong to myself.

If the **project / thesis includes** such materialplease supply the following details:

Please see references section within report.

**I have not** obtained and attached a written permission statement from the owner(s) of each third party copyrighted matter included in my project / thesis

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Print Name: \_ Alexander Edward Davis\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: \_19th May 2017­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**N.B. If you are at any time in consultation with a publisher regarding this work you will need to declare the copy held on UCEEL. Some publishers may regard the UCEEL copy as constituting prior publication. The copy can be removed from UCEEL if it becomes an obstacle to future commercial publication.**

**Official Use only**

ORION unique number: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Date added to the system: **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**