G52GRP Interim Group Report

Bliss Charity Mobile Application

9th November 2015

Group Id: Blanchfield 1

Hamzah Abdulla: 4232805

Samuel Allison: 4231121

Daniel Betts: 4229939

Oliver Bodinar: 4224835

David Fricker: 4203801

Chen Jiahe: 4255607

Thomas Vine: 4234145

Supervisor: Peter Blanchfield

What is the problem that needs to be solved?

Around the beginning of October the group was presented with a brief; the main aims being to design and develop a multi-platform application and a website for the neonatal charity Bliss. Bliss provides fundamental information, support and advice for parents with pre-term babies. These infants are born earlier than expected making them more vulnerable to the risks of ill health, stunted growth and other developmental problems. Consequently, these fragile babies spend more time in the hospital than other babies born closer to their estimated delivery date (EDD), being looked after by professional clinicians and nursing staff until they are deemed well enough to be discharged. It is during this important time that the need for advice and support is paramount for parents and Bliss has been established to provide such a resource during the difficult transition from hospital to homecare. Due to the length of time spent in hospital, the parents finally leave with little knowledge of how to care for their fragile baby and every little irregularity causing them to panic. With the application, we are hoping to give them easy and fast access to specialised and vital information without the need for further hospital involvement.

With Bliss' already existent website proving very impractical and hard to navigate through we immediately identified the source of the problem being due to the fact that important information, which is vital to the care of the baby, was hard to get to and the delay was generally caused by the time taken to open the site – not to mention having to further navigate the extensively broad and deep navigation menu to get to the relevant information. Our main goal of designing this application was to streamline the menu design and allow fast access to helpful information required by parents at a particularly stressful time. To achieve this it was necessary to consult with a professional contact at the Queen's Medical Centre.

After discussions between ourselves and the clinician, it was found that we would need to design the application on the two most popular mobile platforms, iOS and Android. Furthermore, the idea of the website was scrapped as Bliss decided they would be redoing theirs and therefore had little need for ours. However, for consistency, the layout and design would be kept the same for both platforms by working off a similar set of prototypes designed in advance. The multi platform application decision was to ensure that as many people as possible had access to this application. All of the resources i.e. text, videos and pictures present on the website are to be supplied to us via the Neonatal team at the Queen's Medical Centre as well as from Bliss itself.

It was also agreed upon to include a forum to enable parents to contact each for mutual support. The forum provided a safe place for parents to share advice as well as post comments and advice. Here the major issue we faced was moderation, mainly, who would be watching the forums for any negative activity that would inevitably present itself.

As a result of our efforts, Bliss have shown great interest in adopting this application and allowed us to to use their branding so parents who are familiar with their website will trust the application.

Research and background information

At the beginning of our design stages, we did some research into other applications that performed similarly to what we hoped to include in our final application. From what we could find, there were not many that were providing a similar experience or range of function to what we were aiming to do, however, numerous applications implemented one or two features key features that were similar to what we wanted, as a result, we looked to each of those for inspiration towards our design decisions.

Knowing our application would be quite unique in its usage as a medical information application for a very specific use case, we chose to focus on the demographic of people most likely to use our finished application. We researched the applications that were popular with that demographic in order to help us tailor an application for our niche.

After speaking with our client Shalini, who works in the field of child health, we learned that it is was most common for mothers with preterm babies to either be quite young or very old, but few in between the two age ranges. Of these two core demographics, the group most likely to use an application for gathering of information would be the younger one. In a *Pew Research Center American Trends Panel* survey conducted in late 2014, it was found that over 85% of young adults (between the ages 18-29) owned a smartphone, and of these, over 62% used their smartphone to look up information about a health condition.

We also estimated that while our application is aimed at parents in general, our user base would mainly consist of mothers. As a result, we decided to implement features to make the application more appealing to both mothers and fathers alike. We aimed to design an application where the features were just as helpful to mothers as they were to fathers so that we were not ruling out anyone as a potential user. To do this we implemented a 'couple login' feature, where both parents could have separate accounts with separate favourites and forum identities but were both based around the same essential core aspect of the baby. In doing so, we ensured we were not cutting anyone off from essential information or limiting our user reach and that each parent could have a more custom experience when using the application. The visual design was the aspect influenced the most by choosing to focus on both mothers and fathers, we decided to keep the interface quite neutral, intuitive and welcoming so as not to risk alienating either user.

Knowing our core demographic allowed us to research other applications most commonly used by that group, to find features to implement into our own application, or to find the most successful ways of implementing features we had already decided on. Our menu was going to be something of a challenge as we had a long list of articles and features to be able to swap between, some inspiration was gained from looking at popular applications such as Facebook and Snapchat to see how menus were handled in other applications used by our target audience.

To decide what systems to develop for we again looked to how many people from our demographic owned what kind of system. While 85% of the demographic owned a smartphone, far less of them owned a tablet, just 48% of young adults owning a tablet with 52% of 30-49 year olds owning a tablet. It seemed the smartphone would be the preferred system of choice for the people most likely to use our application, so we decided to develop for phones. We also looked into which types of smartphones were the most popular, and it was clear that the iPhone and Android markets were by far the most popular, especially so in our target demographic.

As we were developing for two systems, we had to choose what tools we would use as there are many available for developing for Apple and Android systems. Android was a simple choice as one member of our team had a lot of experience using Android Studio, and so that was a good fit. No one had as much Apple specific experience, leaving us to decide which language we'd be using. An initial suggestion was to use ionic, a language that can quickly create an application and be exported to both Apple and Android, but we eventually decided that it didn't give us enough control over the final product and instead chose to limit ionic for creating prototypes. We finally decided on using Xcode, Apple's native IDE, for the creation of the application as it was the most well documented, allowing for quick learning and problem solving.

After doing research into any algorithms that would allow us to compare baby weights to averages on a graph, nothing specific enough was found. We needed a custom algorithm that would compare the weight of a baby, week to week, and compare it to certain growth trends. We learned, from our client, that staying within two centiles on the graph was the perfect place to be for a baby and needed an algorithm that could do this comparison with ease. In the end, we decided to write one ourselves that will compare the weights and trend lines as we can alter this to be as specific to our requirements as required.

Requirements specification

1. The Mobile Application

#	Feature	Definition
1.1	Search	A search box to match phrases against all text information contained in-app. Localised to search current area of app, e.g. the articles, or book titles.
1.2	Core content available offline	Articles (Text and images only) should be available without an internet connection from the first launch of the app.
1.3	Store baby weight	Provide simple method to store baby weight over time, provide warning message if the weight is entered more than once a week.
1.4	Graph baby weight	Display baby weight super imposed on a graph of baby weight data from the NHS.
1.5	Measurement alerts	If baby weight/height is too far off target for too long, a non-threatening tip should appear suggesting to the parent to discuss this with the doctor on the next visit.
1.6	Message board for users	A place for parents to interact with each other sharing tips and answering questions.
1.7	Shared accounts for partners	Parents share login credentials, where the joint account can have one or more babies associated with it.
1.8	Simple account switching	Allow app users to switch between shared accounts simply and quickly without having to re-type login credentials repeatedly.
1.9	Provide access to eBooks	Provide users with a selection of eBooks, available for viewing in-app via PDF format.
1.10	Provide favourite functionality	Allow users to favourite /bookmark articles for quick access later.

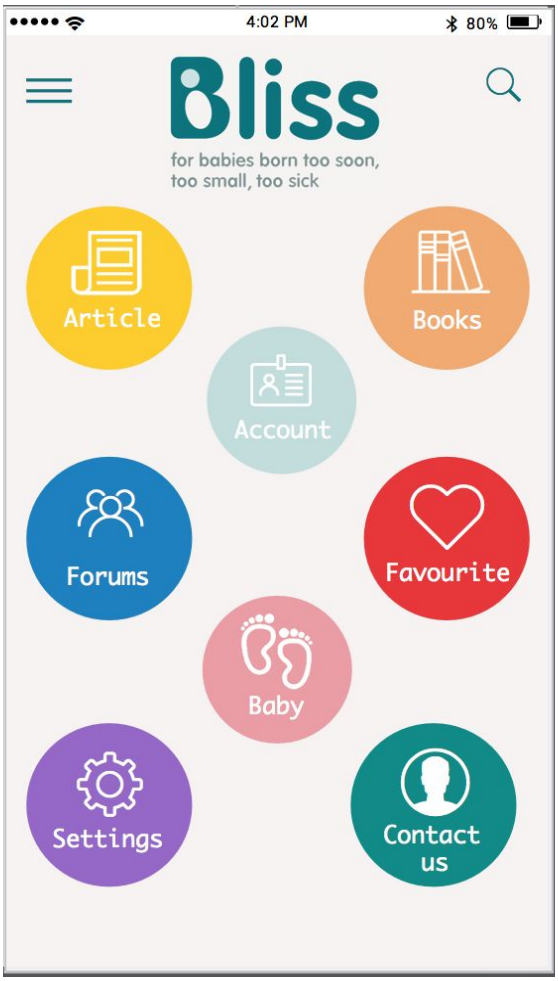
1.11	Change password	Logged-in users should be able to update their current password to a new one.
1.12	Reset password	Users who have forgotten their password should be able to reset it buy using a security email provided at signup.
1.13	Provide access to articles	The app should contain articles, which consist of pictures, text, and links organised by menus which can be nested inside of each other.

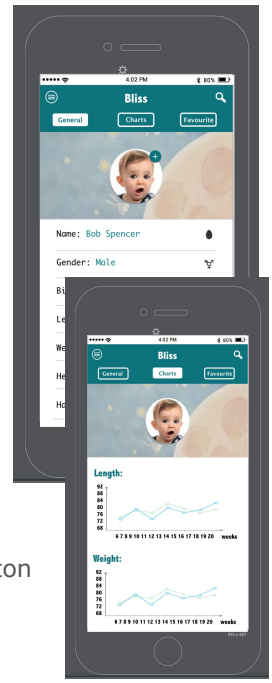
2. CMS requirements

#	Feature	Definition
2.1	Moderation for the message board	Allow moderation of the message board by nurses and other authorised users.
2.2	Varying access levels	Moderators should only have access to moderation tools and not article editing tools; the same goes for editor accounts. Administrators should have the ability to add and remove these users.
2.3	Rearranging menu items	Menu items should be able to be moved up or down in the order of the menu.
2.4	Adding new menu items	The addition of new menu categories should be possible.
2.5	Delete menu items	Removing current menu categories should be possible.
2.6	Article edits	The ability to edit the contents of an article using a visual editor (by use of a WYSIWYG editor).
2.7	Article addition	The ability to write new articles using a visual editor (by use of a WYSIWYG editor).
2.8	Article deletion	The ability to delete existing articles.

Initial design of the system and the interface

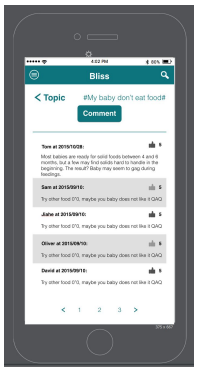
The categories we have are based on the current Bliss website. The main page will be displayed like figure 1 (see left). The left menu icon opens up a burger menu that displays a detailed list of all the categories, including, but not limited to: In Hospital, Going Home, At Home, Growing Up, The Next Pregnancy, Coping with Loss and Support' etc. When the user clicks on the menu icon, the categories will slide out from the left, furthermore, the user can also access the burger menu by swiping in from the left of the screen.



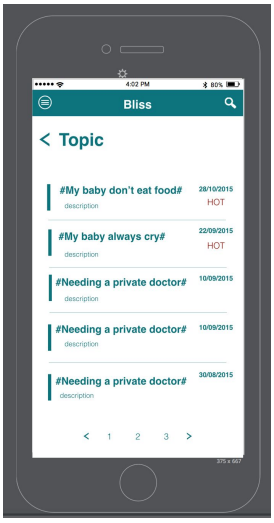


If a user clicks on the baby button on the main page, we will have the following three sub pages: Baby profile - which will display the basic information for the baby , Baby Chart - which will display the baby's weight and finally, Baby Graph - which shows the average baby trends so parents can compare the growth of their baby to data corresponding with typical growth patterns. The favourites section, which can be accessed from the bottom tab bar on iOS and the top navigation pane on Android, will contain a collection of saved events, articles and tips and these will be specific for the device.

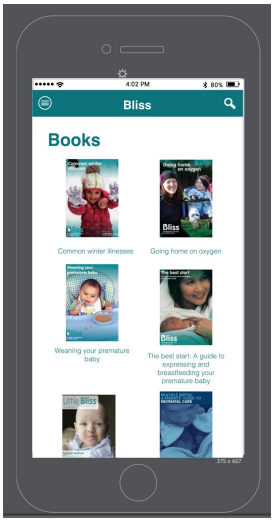
When clicking on the forum button on the main page, it will primarily display some hot topics and displayed relevant questions the time



questions. Should the user want to see something other than the first articles, they can click on the topics or questions section to see all of the material posted there and have a browse through. In the topic and section the user will see: the topic name, the description of the topic and of posting.

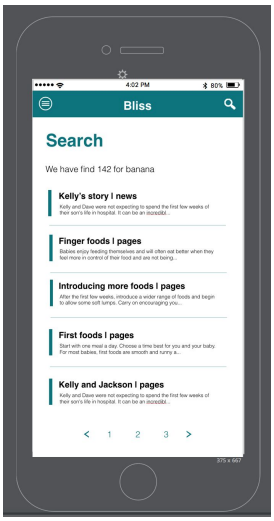


When clicking on a particular topic, the user can see all the detailed information for that question or topic. Users can 'like' any given comment, and all the comments will be ordered by the number of likes that they receive. If the user clicks on the comment button above, it will enable the user to add their own comments.



Under the Book section, located on the main page in the navigation bar, we display all of the books available from the website. They will be displayed as a list containing the cover page of the books and the title of the book. Each book is stored as a PDF, and it can be opened as a PDF as well, either within the app for quick reading or an external PDF reader for convenience.

The search results are displayed quite comparably to the favourites section,



in that the layout is similar, however, the order of the articles would be sorted by relevance to the search item. On a further note, the articles page is also displayed in a similar fashion, so as to stay consistent in our design choices and keep the general user experience intuitive throughout the application.

The UI for pages with more difficult to display content, such as disease, is slightly different. Contained within a box at the top of the page, a brief introduction and some important tips will be displayed, immediately followed by other detailed

information related to the topic, this other information may branch off into its own page if necessary.

For all pages within the app, the navigation bar will be present at the very top. This allows easy access to other pages in the app, and also helps us to maintain a consistent look and feel as pages have common elements to them.

Implementation choices

As our specification requires us to create a mobile application we decided that we would target the Android and iOS platforms as between them they make up 91.4% of the mobile operating system market share in the UK. The next most popular mobile platform is Windows Phone with 8.2% market share, we decided not to create a dedicated application for Windows Phone as we couldn't justify the time that would be required to be spent on development compared to how little usage the application would get. However, in the future we do have the potential to port the application to Windows Phone using software such as Microsoft's Project Astoria, which allows Android applications to run on the Windows Phone operating system with little to no changes when it is released.

For Android development, we are using the Java programming language, with Android Studio as our development environment. Java is the standard language of use for Android programming and luckily has a lot of well supported libraries which are useful for our project, such as Retrofit for downloading and parsing JSON files, and Realm for storing data locally on the device. Android Studio is a relatively recent addition to the Android development kit and is based on the IntelliJ IDEA Java IDE and has some very useful tools built in, features such as: an easy to use drag and drop layout builder and tools such as the SDK manager etc. As we have been using Java a lot in our course modules this was a good choice for us as it meant that we didn't have to learn a new language to start on the development, just the Android framework and APIs.

For iOS development we decided on using Apple's new Swift programming language, and Xcode as our development environment. We decided on using Swift over Objective-C because it seems that Apple are currently putting a lot of effort into the language, for example, they recently open sourced the entire project, and therefore it seemed like it would be better for us to use the most modern development platform as we were creating an entirely new project. Use of the OS X operating system is required to run Xcode and develop iOS applications, we did not have a problem with this as members of the group and those on the iOS development team had Macbook laptops which they could use to work on this.

As a group we decided that we needed to have a server-side part to our project as we needed to easily update information such as the menu categories and articles which were going to be displayed on our application. We also felt this would make adding new articles or content to the application much simpler in the future. To facilitate this, we are currently working on the development of a JSON API that the applications can connect to and parse. Doing so means that the applications now requires a network connection, we also decided that we would cache the information on the devices so it can be used offline, which will be useful for our target audience, as the application may potentially be used in hospitals a lot which are infamous for having a lack of phone signal.

Offline usage was also a considerable advantage to those users wishing to access their favourite articles at their own convenience without the need for a connection. We will also include a base JSON file with the application so that if the user does not have an internet connection on the first run then they will still be provided with information, and then when they do run the application again in the future with an internet connection, the latest information will be downloaded. Having a JSON API also tied into a later design decision we made to expand the application further and have it store data, such as weight, about the babies, tied to an account on the server.

We will also be developing a content management system (CMS) to allow users to update information in an easy to use manner, as those potentially updating the information in the future may not be as technically able and should not be expected to use a complicated system designed for developers to have to update the information. We decided to use PHP to develop the CMS because it will potentially not be running on servers we control in the future, and therefore it will easily fit into a server set up with the very popular LAMP software stack. The LAMP software stack is made up of Linux, Apache web server, MySQL database, and PHP; these are all free, open source software packages.

As a group we have a range of Android and iOS hardware devices which we are able to use to test our applications. It is important for us to test our applications on many different types of devices, especially on the Android side where users will be running a huge range of screen sizes, and versions of the Android operating system, and it has been known that certain vendors have implemented certain API methods differently than they should, and can cause bugs on certain devices and not others. We are also able to use software emulators for both Android and iOS to test the applications and make sure that they look fine on both phones and tablets, as we need to make sure that they look fine on the different screen sizes.

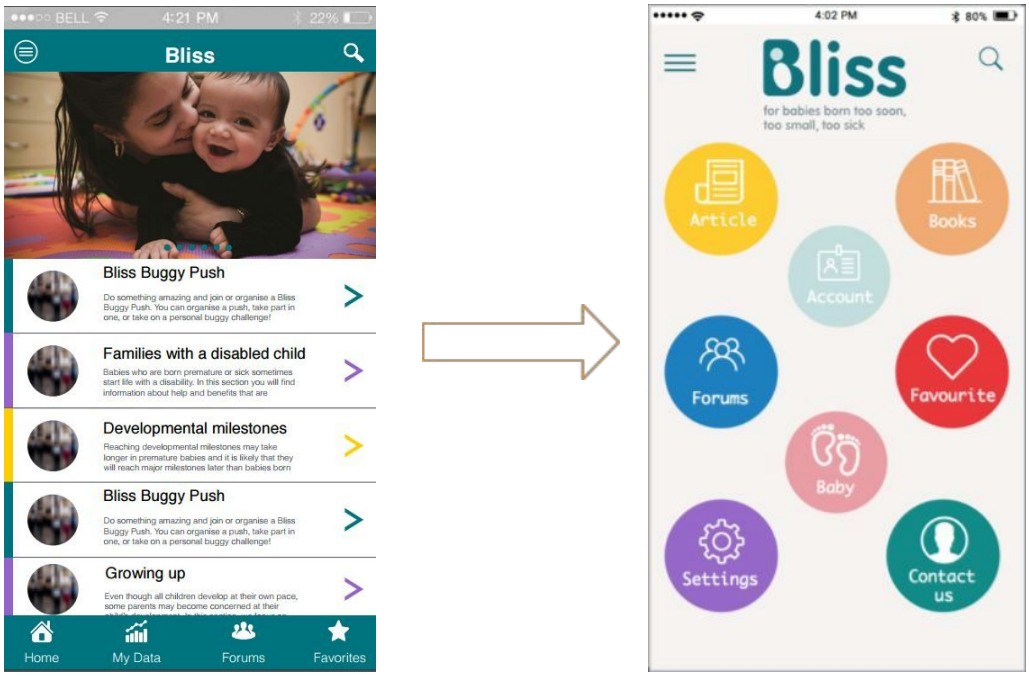
Results of prototyping

After our primary stages of prototyping and implementation decisions we discovered, in subsequent meetings with the client, that the system we were designing was to be larger and have greater

functionality than she had led us to believe in the previous instances. The client was pleased with most of what we had decided to do with the design of the system and the direction we were going with both implementations of the application, both on Android and iOS alike. As time went by, problems arose at an alarming rate, we found that the menu structure of the website that we were basing the application upon was incredibly poor, not only because of the reduced screen real estate that we would have on the phone application compared to the website, but because the breadth and depth of the general menu hierarchy of the application was much too wide and much too deep, respectively, for any useful or convenient navigation.

Finding an article you wanted meant you had to navigate through an ocean of menu categories, some of which were 5 layers deep causing the search time for users to be quite high.

As a result of this, we had to halt our development process as we held a plethora of general meetings with only our group members as we tried our best to arrive at an appropriate menu structure that allowed the user the optimum balance between functionality and ease of use. So that the discoverability of the content was kept as high as possible whilst taking into consideration the amount of data available to display. In the article: *"A comparison of broad versus deep auditory menu structures we discovered the research that illustrates that users cope better and have a general preference towards menus that tend towards breadth over depth, this plays a part on the short term memory capacity of general users as deep menu structures are harder to remember."* We eventually came to the conclusion that we were depending on the website too much for the menu structure. In wanting to keep the application similar enough to the website so it was easier and more familiar for users that had previously used the website we were sacrificing the individuality that comes with an application. We changed the entire structure of our menu. In accordance to this decision, we included that along with this we would make the best use of our menu access and main page. We turned the main page into the main navigation pane, replacing the welcome screen containing article shortcuts with a simple and attractive navigation pane.

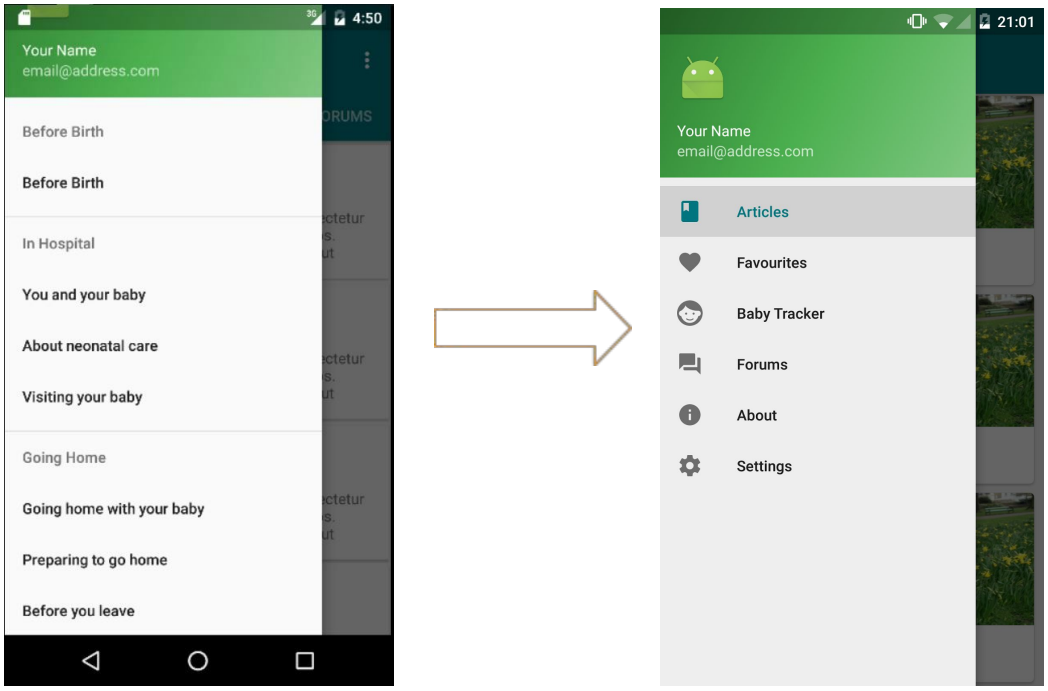


For our menu category layout, we decided to give the articles, that were the source of our menu troubles, a category all by themselves so that we would have an entirely separate page and layout to handle them and give them the attention that they deserved.

This in turn made the burger menu redundant and so we decided to remove it entirely. And instead, selecting something from the main navigation pane, for example the “Articles” page, would slide the pane over to the left, exposing a further menu layout dedicated to handling the specific category with a further menu category.

This would avoid the steep learning curve as implementing the slide in feature gives the impression that the user isn’t somewhere completely new, but instead, accessing a part of the main interface.

This implementation allows us to present the user with a more intuitive way to navigate through the numerous articles available through a navigation menu dedicated to that option. Taking into account that the “Articles” option itself is a major part of the design, we wanted to provide the user with the best and easiest navigation experience that we could.



The depth of the "Articles" category brought with it another problem; with a user deep into the category, having navigated through all the options available, if they wanted to get back to the main page they would have to navigate all the way back, this was where the burger menu re entered the scene and allowed us to give the user the option to either return to the main screen, select a different menu category or just quickly go back to the highest hierarchy of the current menu they were in.

With the presence of a back button in Android, and the swipe to the right gesture to go back in iOS, we decided to leave the burger menu at the top left of the screen, on the left in the toolbar. And in turn, decided to also have a back button in the burger menu for those users who weren't as keen on the swipe feature in iOS.

Problems encountered so far

While working on a joint project within a large group there will inevitably be certain issues and problems that arise. These issues can range from small changes in the overall time plan to major issues, for which we will need to reassess what we believe we can achieve in the given timeframe. The first set of issues we had to deal with was working with people we did not previously know, this can lead to problems as the workload begins before the group has a chance to get to know one another, in particular their strengths and weakness and how they can be best applied within the project. This is a common occurrence within a group project, in particular

among those with limited experience within a professional team, which included ourselves, however this was overcome quite quickly as we began to meet early on outside of the group work to get to know each other and bridge this gap.

We were hastily informed, during our first meeting with our supervisor, that a member of our group was going to be involved in the study abroad program within the university. This meant that they would be spending the second half of the academic year at another university and would create many issues as they would still need to be involved in our group project while being in a different country. Upon further investigation it was discovered that the time difference was not as severe as first feared, being a full 12 hour difference would allow meetings via Skype at 9am GMT which would just catch them before they went to bed. By conducting our meetings in person as a Skype call this would allow them to be directly involved and feel like they are still a member of the team instead of only sending and receiving emails regarding their individual workload.

Whilst assessing each individual's strengths and weaknesses it was clear there was many different levels of experience on different pieces of the project. These different skill sets gave us a problem as it is difficult to assign multiple individuals to one piece of work where one of those individuals has lots of previous experience and the others do not, an issue being finding the balance between everybody contributing equally to the work and one person racing ahead and completing everything. In order to limit this, we made sure when assigning work to split the work into portions that, although were in varying difficulty, would take approximately the same time to complete, as a result of the range of abilities.

A technical issue that came to light early on with the use of GitLab. GitLab allows individual members to upload their progress onto a remote central hub for other members to access. This is a noticeable change of mindset for us as we are use to only having to save work locally and not remember to keep uploading changes as we go. This would become a larger issue if one member's work was dependant on progress another member had made, and if they have not updated the central files it could slow down progress of the project. In order to overcome this, during our meetings, we would look into the existence of these dependencies that week and those members could plan on which days the tasks would be completed.

One element that differentiated our project from others being undertaken is that we have both an academic supervisor and a project client, with the intention of using our product after we have completed our project. This meant we had extra issues to contend with that others did not, these included the addition and/or deduction of certain features as the project progressed depending on the resources available and the clients' needs. At the start of the project, the application was primarily to display information and over time additional features such as a Forum and a Weight Tracker have been added.

These different elements have different levels of importance to the project and different workloads, and as such we have had to take an Agile approach to the project as we are increasingly aware that during the time it takes to complete the project, the application may have become completely obsolete. During the project we would have biweekly meetings with the client in order to showcase our progress whilst gaining feedback early on which could help to keep us on track of what they needed. An example being the Forum, an idea that was not originally included but added early on and then pushed back once the development and ongoing costs became apparent. Due to our agile approach we were able to effectively adapt to these changes without a drastic loss of progress.

Initially our time plan for the project was to have a working testable prototype in January after we returned from the christmas break, this was pushed backwards by our client who wished to trial the application with volunteers as soon as possible. We believed at the time this was an achievable goal to set due the amount of work to be done, the free time we had then and the workload from our other modules. In approximately late October/ early November the workload from other modules became heavier than we had anticipated, this limited how much time we could dedicate to the project and as such, stunted any progress we could make.

Upon consulting amongst ourselves, we decided that we would not be able to deliver a complete product of the desired quality we wanted by this deadline. Faced with this problem we adapted and, with the consultation of our client, pushed this deadline back to mid February. This would give us the time needed to complete the prototype to the standard we felt we can achieve whilst also allowing us time to focus on our other pieces of coursework and of course, our January exams.

REFERENCES:

- Pew Research Center American Trends Panel
  - [http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/]
  - [http://www.pewinternet.org/files/old-media//Files/Reports/2014/PIP_E-reading_011614.pdf]
- A comparison of broad versus deep auditory menu structures
  - http://www.ncbi.nlm.nih.gov/pubmed/18354973

- Smartphone OS market share
  - http://www.kantarworldpanel.com/global/smartphone-os-market-share/

- Project Astoria
  - https://dev.windows.com/en-us/bridges/android

- Swift open source
  - https://swift.org/about/#swiftorg-and-open-source