Context

# Introduction

The following chapter is focused around the development of the idea proposed in the Chapter Three: New Ideas. Within; a summary of the steps taken whilst following the Lean UX project methodology, the tools used for development are discussed with a focus on the reasoning behind this choice, a summary of key elements of the system that were developed with the aim of meeting the projects goals and finally a summary of the user testing that took place during the project.

# Lean UX project methodology

This section is a summary of the steps taken by the development team to accurately follow the process of managing a project using the Lean UX project methodology (Gothelf and Seiden, 2013). For a full writeup of the process and results of the individual steps please refer to **Appendix A: Lean UX**.

The first stage of Lean UX is to identify a problem with the current status quo, this is to say some situation where the goals of a target audience are not being fully met. In this case, the problem identified in Chapter 3: New Ideas, was that there are no solutions that offer a more passive approach to user fitness and entertainment, and that the uses of such a solution have yet to be fully explored.

Once a problem had been identified the project team conducted research into the user demographic of the proposed application, this step is conducted as it aids in empathizing with the target demographic, and helps to identify any bias or false assumptions the project team may have. These steps are all in aid of creating a better targeted solution for the users. The result of this stage is a list of assumptions the project team had about their target audience.

Once the previous stage was complete any assumptions the development team have decided upon were ranked in order of priority, the justification for this is to generate a testing schedule that focuses upon the highest priority items first. Priority is determined by a combination of two factors; knowledge and risk. Those assumptions where the team had little knowledge of the validity of the assumption but presented a high risk to the projects failing were ranked as the highest priority and so required testing first.

With a prioritised list of assumptions, the next task was to develop a series of hypotheses, these are summaries of the assumptions discussed in the previous sections with the key difference that these hypotheses also contained a marker for their success. This allows the team to be able to quantifiably tell if the hypothesis in question is in fact true and valid. Below are the two hypotheses that were established as a result of this investigation\*;

“I believe my target audience would make use of an application that aims to produce an enjoyable experience but that offers them the opportunity to gain an in-game rewards if they complete a fitness based challenge. I shall be able to test this by producing a prototype application which combines the above elements, which I can then use to test with members of the target demographic. ”

&

“I believe one of the biggest risks to the success of this project is a lack of knowledge of whether my target audience want an application that aims to achieve the gaols laid out in this report, I can produce a paper prototype that will allow for a rapid testing session with my user base to conform or invalidate this hypothesis. “

*\*The format perspective of the hypotheses is in first person, this is in line with the guidelines of Lean UX (Gothelf and Seiden, 2013)*

The last step in this process is the creation of minimum viable products that are created to test each individual hypothesis for validity, in this case a paper prototype and an electronic prototype of the application were constructed as is in line with the requirements of hypothesis one and two. The results from the creation of these minimum viable products are discussed in Chapter 5: Results.

# Development

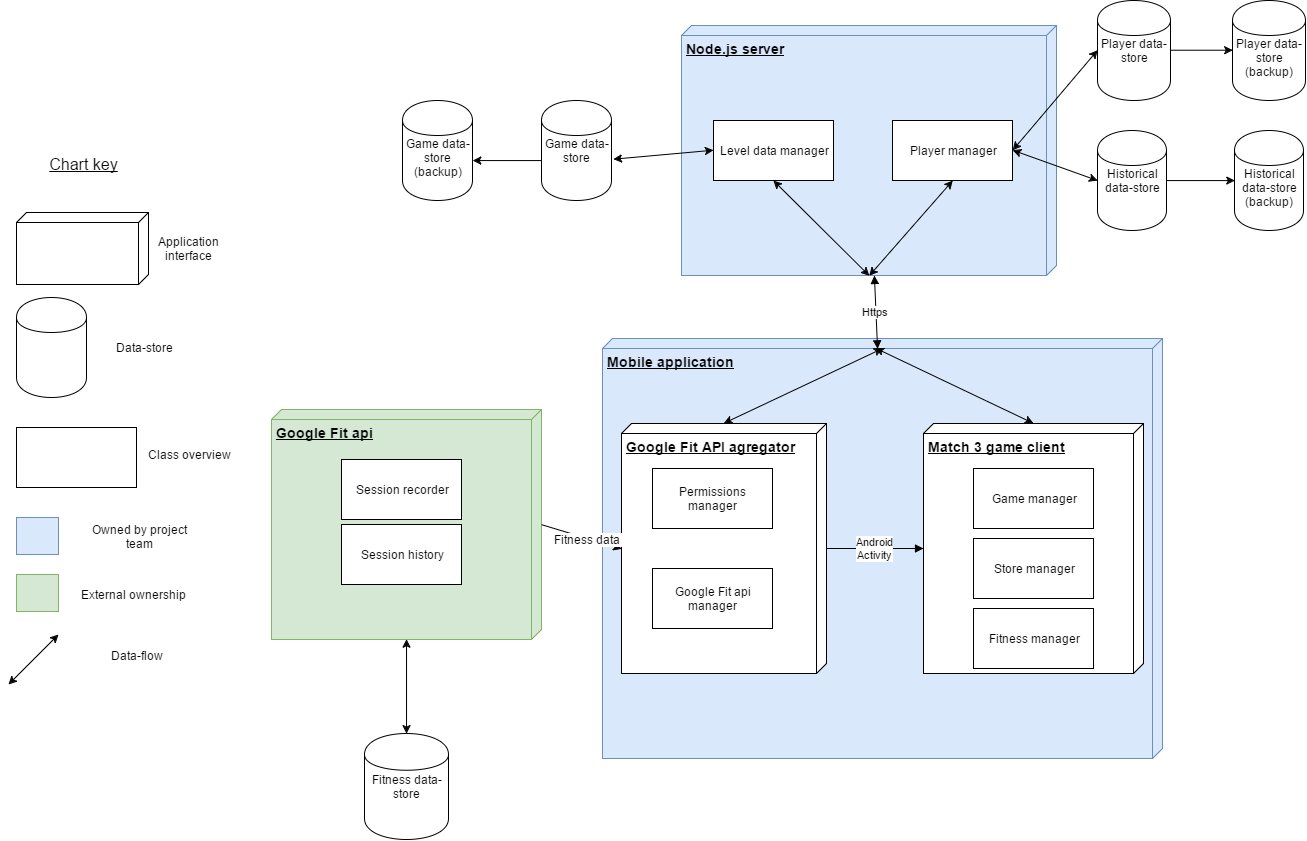
Within this section, the development of the application to fulfil the projects aims is detailed. First a development overview is explained to discuss the general architecture of the project application, which proceeds into a discussion around the tools used to achieve the required functionality, finally this section is rounded off by a more in depth look at some of the key functionalities within the solution.

## Overview

The application developed to achieve the aims of this project can be split into three key components, a Node.js webserver, a mobile game and a wrapper application to link with the Google Fit API. The general functionality of these components as well as how they link together is discussed below, in addition to this a high-level system overview diagram is provided in **FIGuRE !** to aid in representing how the system operates;

* Google Fit API wrapper – This section of the application is designed to request permissions from the user to gain access to their historical fitness data, once permission has been received, the application will then proceed to collect the fitness data relevant to the running of the overall solution. Once the relevant data has been collected it is then pushed to the Node.js webserver to be stored in the users datafile. Which shall be used later in the application when calculating fitness goals.
* Node.js webserver – The Node.js webserver is designed to maintain security for the overall solution. Data relating to the user is stored in a secure server as opposed to the device and only the user profile in question is able to subsequently retrieve this data. Player and level information is stored within the server as to allow the application to be a lighter in weight. Additionally, storing this information enables the application to meet its goal of having a focus on social interactivity, as highscores between all players can be stored using the webserver to promote some competition between the application userbase. The webserver also enables functionality outside the scope of this project such as the ability to create cross platform applications that would run on several operating systems, as key data related to the running of the application is not stored locally. Finally, historical data can be stored on the usage of the application with the webserver, this will aid in future work outside the scope of this project but discussed briefly towards the end of Chapter 3: New Ideas and in more depth within Chapter 6: Results.
* Mobile game – As decided upon in Chapter 3: New Ideas the theme and style of the game used in this application is based upon the Candy Crush Saga, this being a match three genre game in which the player must reach a certain score be matching up three or more items on a screen. The mobile game is the only aspect of this project that is visible to the end user, as such it must meet all of the thematic styles of Candy Crush Saga the game decided as a base template for this application in Chapter 3: New Ideas. The game element of this solution pulls data relating to the player upon start, tracks the player’s fitness progress and provides motivation for further action through the use of rewards that can be spent in game on improving the players in game profile.

### System overview diagram



## Development tools

The following section outlines the main tools used during the development of this project as well as justifications for these choices. Additional tools such as Git, Photoshop and Atom were used throughout the project, however the contributions of these tools to the project are too minor to receive full discussion in this section.

### Android and Android studio

One of the objectives of this project is that the application should be designed to operate in on a mobile operating system, this is to allow for ease of access to the applications, as so long as the user has their mobile device with them they will be able to make use of the proposed application. Several operating systems are available that could be used to host the proposed application. Android has been chosen as the most suitable for the following reasons (Martin, 2016);

* Android holds the largest market share of any mobile operating system, in Q2 2016 it saw a market share of 87.6% with Apple’s iOS holding 11.7%. Table 1 below shows the dominance of the android platform over other platforms between 2015 and 2016 (Martin, 2016)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Period** | **Android** | **iOS** | **Windows Phone** | **Others** |
| **2015Q3** | 84.3% | 13.4% | 1.8% | 0.5% |
| **2015Q4** | 79.6% | 18.6% | 1.2% | 0.5% |
| **2016Q1** | 83.4% | 15.4% | 0.8% | 0.4% |
| **2016Q2** | 87.6% | 11.7% | 0.4% | 0.3 |

* Google hold a userbase of 1.4 billion active Android users (Martin, 2016), such a high user base would enable the application to gain access to a greater number of potential users improving upon the validity of any potential scientific testing conducted within the realms of this project

#### Justification for a single platform approach

With android set as the primary choice of operating system, the question did present itself as to what method of development would be used to produce the Android code, indeed there were several options available each offering different feature sets, some of the options present such as Cordova and HTML5 presented the opportunity to create an application that would work cross platform with Android, IOS and within a Web browser (Vensi, Inc., 2017; Rajput, 2017). However, these options were passed up in favour of developing through Android Studio with Java for the following reasons;

Google Fit is an open API that comes installed in Android phones operating version 2.6 and onwards (Google Developers Android API, 2017), this represents a significant portion of the android userbase (Developer.android.com, 2017), as the application proposed in the report intended to use fitness data to improve the users experience with the app the use of the Google Fit API was a crucial piece of functionality. The choice of Android Studio was in a large part influenced by this key functionality as access to the Google Fit API was not available through the presented other options as the trade-off of having multiplatform availability meant the lack of access to functionality native specifically to the Android platform

* Android Studio also provides the use of the Android Virtual Device manager (AVD), the AVD allows for the testing of an application on devices with multiple different configurations such as operating system version and screen size, as the proposed application is designed to appeal and work for a large audience, testing with these variations is another key reason Android Studio was chosen as the platform of development for this project.

### Node.js, Express and CouchDb

The tools in this section all make up a web server set up to host and manage the application.

Node.js is a community driven JavaScript based programming language that allows for non-blocking I/O in an event-driven architecture (dzone.com, 2017). Node.js has been found to be 10 times faster in I/O operations than competitors such as Java (Tilkov and Vinoski, 2010). This is achieved using an asynchronous event loop that does not block up other requests, this allows Node.js to offer a performance that does not decrease with a higher number of web calls. The choice of Node.js for this project is due to its ability to scale well with simultaneous web calls, this will allow the application to scale easily with an increasing user base without the need for any substantial changes to the server architecture.

An additional justification behind the choice of Node.js is the community behind it, with the use of the Node Package Manager (NPM) users have easy access to packages created by the community to fulfil many different requirements, one such package available through the NPM is Express. Express is a web application framework that allows for the relatively simple creation of a RESTful API that in the scope of this project will allow the mobile application to send data to and receive data from the application server. Express was chosen as the most suitable method of managing the http server as it is designed to be light weight and easy to use as well as the documentation available for Express is widely available and well maintained.

The final component within the http server architecture is a database with which to store data to allow the application to be persistent even if a user were to swap their primary mobile device. CouchDB a NoSQL JSON document based database was selected for use in this project due to its ease of use, especially within a JavaScript based environment as the JSON based format used by CouchDB allows for simple translation across multiple languages. The API used to access CouchDB is written in REST, this matches up with the use of Express meaning there will be no additional overhead required to pull documents and send them from the server.

The point could be argued that an SQL database would have performed as efficiently in this project as CouchDB, the justification for using a NoSQL database over one based on SQL is also related to the choices of all the tools that make up the web server, developer preference. Whilst in industry on placement the author gathered experience using the tools discussed in this section as these were a part of the tools used daily by several teams within the organisation.

### Unity and C#

The final major tool used throughout the development of the project, Unity and by extension C# (as this is the primary development language within Unity) is a freely available, cross platform game development tool kit that can be used to develop 2d or 3d games. As Unity is able to produce applications that work natively on Android it satisfies the major project requirement of creating an application that targets mobile devices. Unity was selected over other available tools such as Unreal Engine and LibGDX for a number of reasons;

Unity is extremely popular amongst mobile game developers with 34% of the top 1000 free mobile games being created using unity (Unity, 2017), as such the community around the tool is also vast meaning access to documentation for Unity is readily available and simple to find.

The drag and drop nature of the Unity interface is clean and easy to understand, this also enables the user to focus more on the functionality of the game as opposed to the rendering of items within a scene, as this is done by Unity automatically (Craighead, Burke and Murphy, 2007).

#### Closing remarks

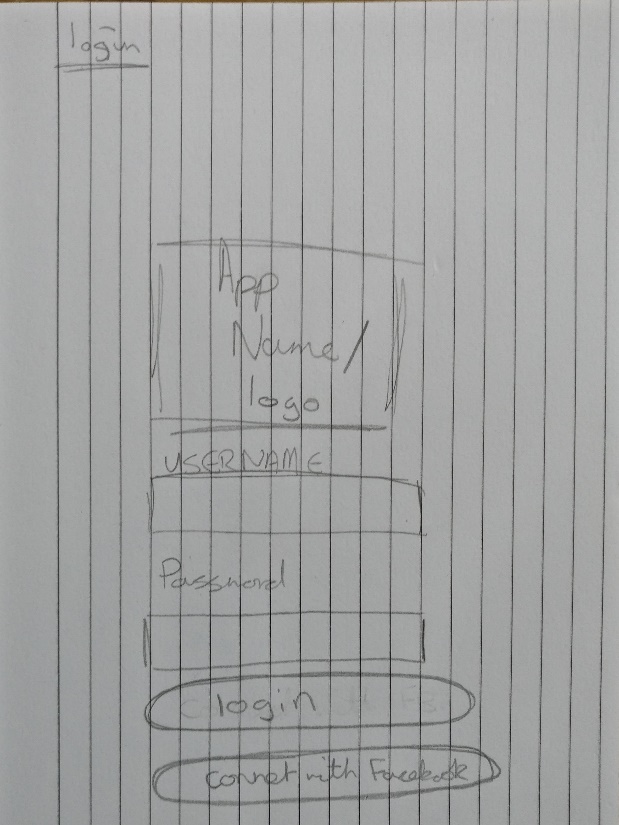
It should be noted however that during the course of this project, issues did arise with the use of Unity that could have been overcome through the use of a different library such as LibGDX, these issues will be discussed in greater detail in the sections below.

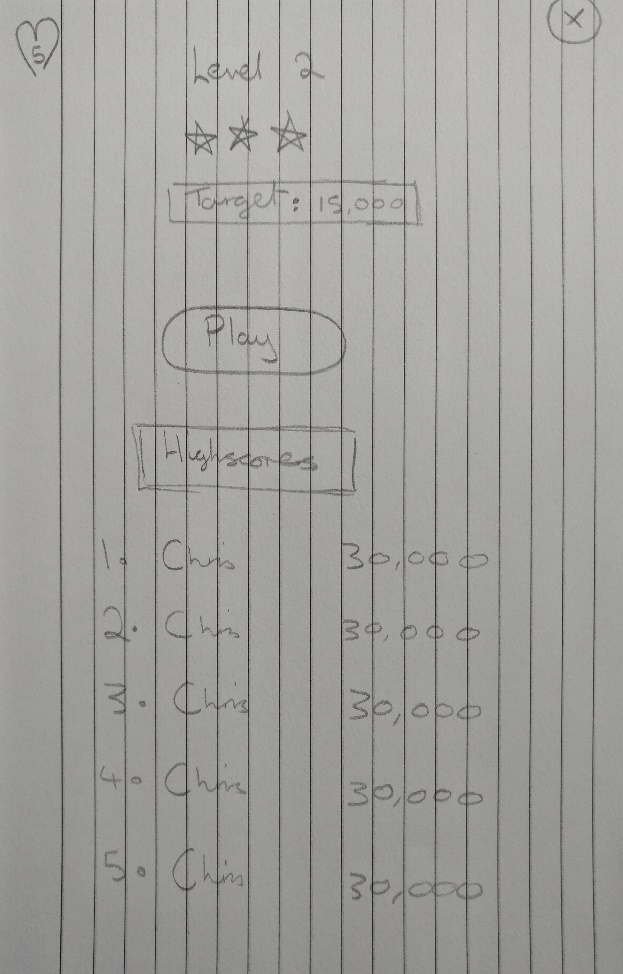
## Key features

Within this section, the key features of the application will be discussed with a focus around the functionality they provide, and the justification for the implementation of these features in the manner chosen.

### Paper prototype

In accordance with the Lean UX project methodology it was determined that a paper prototype would constitute the minimum viable amount of work required to test whether there would be user interest for this application as defined in Hypothesis 2, below are a couple of screen shots of the paper prototype that was developed for this purpose, **For the full paper prototype please see Appendix A:**





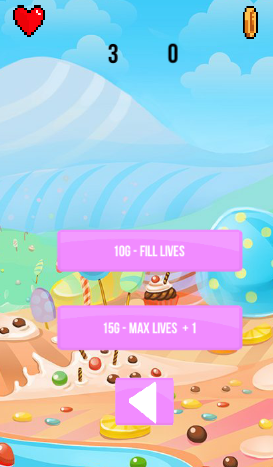
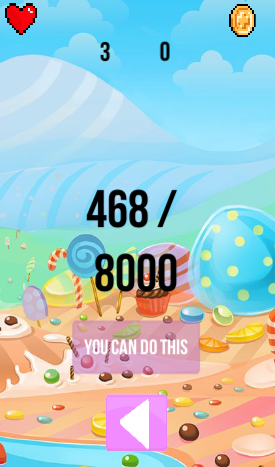
### Candy Crush Style Match 3 Game

At the close of Chapter 3: New Ideas it was determined the most suitable game for development that would meet the goals of the project was a game in the style and genre of Candy Crush Saga, this being a bright and colourful theme wrapping around a game of the match three genre. Following the ideals of Lean UX to create a minimum viable product as quick as possible to test the defined hypotheses, an open source guide on the creation of a match three game was followed as well as assets used within this guide were also incorporated into this project (Dgkanatsios, 2015). This guide detailed the core gameplay mechanics of a match three game such as the functionality of swapping objects on screen to generate score and the subsequent spawning of new randomised objects to take the place of those previously destroyed through swapping. However, the guide only went this far and gameplay elements such as level management encompassing; remaining moves, score objectives, score keeping, highscore recording, and a lives total were not a part of this guide, so required development for this project. **Figure A below shows the game running during one of its levels**



In addition to the developed game play, the game element of this project required close integration with the game server, this was to enable data to pass through the Unity application but not exist persistently on the user’s mobile device, the benefits of this have already been discussed. To enable this functionality an older version of the .NET framework was used to create a REST API interface that could communicate with the Node.js server, an older version of .NET was utilised for this purpose as the version of MonoDevelop; the tool unity uses to build C# code is not currently compatible with more modern versions of .Net (Paczkowski, 2017). Issues were faced in making use of POST requests through the developed RESTful interface, this was due to the security settings present in the webserver and the content tags added when building a URL request in C#. The problem was overcome by altering the way in which requests were sent to the server to rely more on GET calls as opposed to the previously mentioned POST calls.

The fitness motivation functionality was integrated into an activity named “Quests” that would provide context to the user of their current steps taken since accessing the application, if the user is able to reach a certain number of steps they are rewarded with in game currency which could be spent in a Store activity. The Store activity contains a couple of items the user can spend their in-game currency on that are designed to improve the game experience for the player, the items currently available are focused on either refreshing the players lost lives, or increasing the total number of lives the player has available to them. **Figure A below shows the game running during one of its levels**



One final area within the Unity portion of the project worth note is the Highscore activity, the activity is designed to retrieve the current highscores for individual levels so players are able to compare their scores to that of other players, the aim of this is to attempt to kindle some social competition between the user base, this matches up with the project objective of making an application that has a focus on social interaction. **Figure A below shows the game running during one of its levels**



### Node.js Webserver

A Node.js server was created to implement the following functionalities;

#### Social user interaction

With the project objective in mind of creating a game that allows for some form of social interaction between users, the Node.js server would allow for data concerning individual player’s performance in game to be stored in a database separate from the user’s physical device, this allows the game controller present on a user’s device to request information such as other players scores resulting in the functionality for a globally available high scoring system. In its current state the webserver has functionality enabled to record and retrieve a global high score system as well as the ability to retrieve the high scores of users within a certain range, this allows a user to see the scores of other users who are close to their score for a particular level with the goal of fostering some competition and interaction between these users.

#### Anti-cheat safeguard

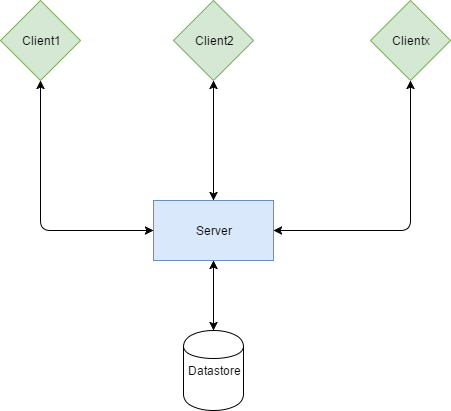
Despite the low probability of this occurring, an additional reason behind the choice of a Node.js webserver was to remove the opportunity for cheating within the user base. Game development guidelines suggest that using a webserver to handle updates to player information where possible is a sensible move as it guards against any possibility for foul play (Bramer, 2014). This functionality is achieved by making the player controller within the game send requests to the server when a major game event occurs, such as when a player fails a level and loses a life. The player controller sends a message to the server updating it of the event, the server will then go through the steps of updating the player’s information and initiates a function to refresh the players missing lives.

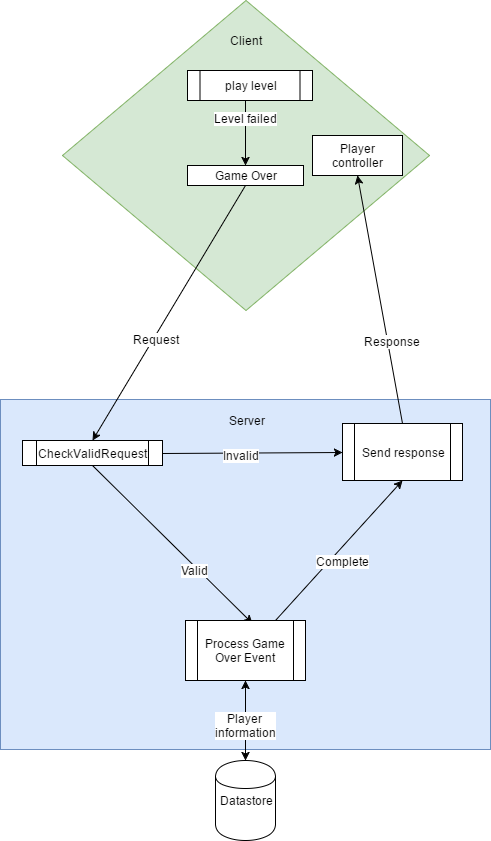
#### Historical data

The final major element of the Node.js webserver is functionality designed around the ability to store historical data about the applications usage over time, recording historical data opens up the ability to monitor the success of the application by checking how often users are logging in and making use of the various features of the application. Historical recording also opens up the possibility for further research such as the potential research questions discussed briefly towards the end of Chapter 3: New Ideas and in Chapter 6: New Ideas.

#### System diagrams

Below are two system overview diagrams, the first **FIGURE 1** shows the interaction between the node server and any clients at the highest level, the second shown in **FIGURE 2** shows a slightly more complex representation of the system process of when a user fails a level in the Unity application and actions the server takes on this event.





### Google Fit API

#### Introduction

Despite Unity hosting much of the mobile application code in this project, the functionality around collecting fitness data, essential to the aim of this project was inaccessible to the platform due to its nature as a cross platform entity having to balance many plates at the same time. To gain access to the user’s fitness data a wrapper application was created using Android Studio that would gather the required information and send it to the Node.js webserver, wherein it could be assessed by the game portion of the project.

#### Selecting a fitness metric

The wrapper application had functionality to determine the current user of a mobile device, establish a connection to the Google Fit api, then pull a variety of different historical fitness data types from the users fitness store. The Google Fit API has access to several types of data that were considered for use in this project such as;

* “com.google.calories.expended“ - This data type would return the total calories expended by the user within a given timeframe. An implementation of this data type could have required the user burn a certain number of calories to receive an in-game reward. However, this method of reward was passed over due to its lack of interactivity and in relative terms quantifiability, as calorific burning would be difficult to for a user to convert in to what actual exercise is required to meet said goal, this is especially so when such a goal could simply be set by the application as opposed to one based on calorie burning
* “com.google.activity.summary” – The activity summary datatype is able to return full fitness activities the user partook in within a given timeframe, this opened up the possibility of setting up more challenging goals for users, for example to partake in an activity that involved using a bike. However, this complexity was ultimately the reason this data type was not used, as it did not really fit in with the applications goal of creating an experience that doesn’t require a high amount of engagement in its fitness aspect
* “com.google.step\_count.delta” – The delta step count data type can be used to return the number of steps a user has taken within a given time, this can therefore be used to determine the number of steps a user took during a single day. This datatype was chosen as the basis value with which to build the fitness portion of the application around.

#### Integration into the wider application

With a metric selected to base the fitness element of this application from the next step necessitated enabling this functionality in a way the user could interact with it. With this goal in mind a system was set up within the Unity project that allowed users to claim a daily reward of in game currency for attaining a certain number of steps each day. At present the current daily target for all users has been set to 8000 steps, this value has been chosen based on the findings presented in Tudor-Locke and Bassett (2004), in which up to 7500 steps is considered within the norm for general activity throughout a day, with the goal being set only slightly above the realms of normal activity a user may become motivated to achieve their daily goal if they are able to see how close they are to achieving it.

#### Reward benefit

Rewards generated from taking part in the fitness aspect of the application are required by the project objectives to provide some in game benefit, to this end when a user completes their daily goal they are granted in game currency, which can be used to purchase a number of player bonuses, such as the ability to fully refresh the character’s life count, and a purchase that increases the players maximum available lives.

### Integration of all components

One key area to this project was in fact making sure all elements were able to interact well with each other, in some projects where for example only one tool is used, integration of an applications classes can be a simple process so long as the development is well thought out and planned ahead of time. However, in this project many unforeseen issues were caused when attempting to combine applications elements together, for example when exporting the game element of the project from Unity to Android, Unity would often overwrite sections of the wrapper application, whilst this issue was mitigated through the regular use of source control, being able to identify what components required manual integration was no small task. Another issue faced during development was through gaining access to the account name of the user once the application had loaded into the game, functionality was in place to fetch the Android account name from the Android wrapper which would then pass it to the Unity application, however due to the way in which the Unity application handled Android Activities this approach would not work. Eventually an entirely new approach required use to enable the Unity project to gain access to the Android account name.

# User Testing

To verify the project was following its aim of creating a solution targeted to a specific demographic, and as a method of maintaining code quality and functionality User testing was conducted on a number of occasions throughout the course project, this section summarises this process.

**Add reference to the appendix with the questionaires and pictures**

## User testing overview

Based on the findings of research conducted in (Martin, 2017), Empirical Usability testing was selected as the most appropriate method of usability testing, this was due to the low requirement of participants within the study, as finding suitable candidates with which to test was an issue throughout the testing phase of this project. Again, following the work in (Martin, 2017) as a final metric of testing on the system, participants will complete a System Usability Scale test (Brooke, 1996), this choice was down to the system usability test being a well-established tool for getting a standardised comparable set of results, which would be unavailable in a project if the team decided upon their own metrics for analysis.

## User testing results

In this section, the final user testing session has been summarised in table form with an analysis of the testing results presented after the table. As with the formal process that goes along with Empirical Usability testing a documented testing guide is available for review in **Appendix A,** which among other things gives a more in depth look at the tasks performed during testing (Dumas and Redish, 1999).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task name & description | Completion rate | Critical errors | Noncritical errors | Average time on task | Subjective comments |
| Task 1 – Achieve a high score within the game and then view it on the game leader board | 100% - Task was completed by all participants | 0%  No critical errors found | 0%  No non-critical errors found | 39 seconds | Quote from participant in trial – “I like it the game is fun, I’m pretty sure my score was higher than that.” |
| Task 2 – Gain an in game reward through the use of the applications fitness elements | 100% - Task was completed by all participants | 50%  Three critical errors discovered within the usability test in separate instances. In two cases this was due to the participant needing an explanation on how to interact with the system. In one case this was due a bug in the quest button, which didn’t allow it to be clicked. | 0%  No non-critical errors found | 13 seconds | “It is a little unclear as to where to press for this, maybe make the game tell the user what this section is available for?” |
| Task 3 – Purchase an in game reward and clearly identify what it was they purchased | 100% - Task was completed by all participants | 0  No critical errors found | 0  No non-critical errors found | 12  seconds | “Pretty simple, though how do I get more coins.” (after a response from the facilitator) “Well maybe it needs to say this somewhere” |

### Interpreting summary table

Three tasks were monitored during the usability analysis, results have been recorded and are present in the table abovethe following can be determined through examination of this table;

* All participants completed the tasks set
* Three critical errors were found during the test, all of which were found within task two
* No non-critical errors were discovered during the usability test
* The rate of critical errors reduced to zero as the usability test progressed
* The average time to complete a task fluctuated throughout the tasks, task 1 presented the greatest range of results, though this can be attributed to the nature of the task
* User feedback from task one was generally positive
* Feedback from the remaining tasks was generally negative as a result of the application not providing enough feedback to the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | User 1 | User 2 | User 3 | User 4 | User 5 | User 6 |
| Task 1 | 33 | 41 | 45 | 30 | 37 | 50 |
| Task 2 | 11 | 9 | 17 | 11 | 18 | 10 |
| Task 3 | 10 | 17 | 13 | 15 | 7 | 11 |

## User testing task time distribution

## SUS test results

|  |  |
| --- | --- |
| User | SUS score |
| User 1 | 72.5 |
| User 2 | 72.5 |
| User 3 | 67.5 |
| User 4 | 62.5 |
| User 5 | 62.5 |
| User 6 | 30 |
| Mean score | 61.25 |

#### Interpreting SUS score results

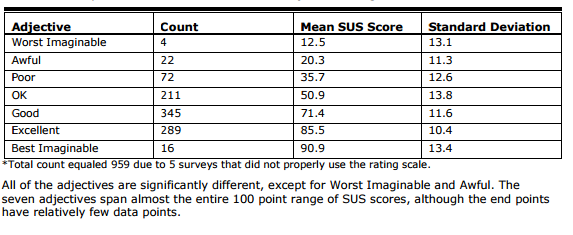
SUS scores fall on a scale of 0-100 with 0 being the worst score possible and 100 the best, analysis of the results from the Usability test suggest a system that for the most part is on the positive side of the scale, when SUS scores are compared to adjective measurements the scores on both correlate well as seen below in table 1(Bangor, Kortum, and Miller, 2009). When compared to an adjective scale the SUS score gained from the usability analysis suggests this application sits between an average design and a good one, this suggests that the design needs to be reworked as not all users considered it “good”. 

Table 1 – SUS score to adjective rating comparision of 959 tests (Bangor, Kortum, and Miller, 2009)

# References

Dgkanatsios (2015). Building a match-3 game (like Candy Crush) in Unity. [online] Dimitris-Ilias Gkanatsios. Available at: https://dgkanatsios.com/2015/02/25/building-a-match-3-game-in-unity-3/ [Accessed 6 Jan. 2017].

Martin, C. (2016) .NHS System Proposal. Unpublished manuscript, Nottingham Trent University.

Google Developers Android API. (2017). Android APIs | Google Fit | Google Developers. [online] Available at: https://developers.google.com/fit/android/ [Accessed 26 Apr. 2017].

Developer.android.com. (2017). Dashboards | Android Developers. [online] Available at: https://developer.android.com/about/dashboards/index.html [Accessed 26 Apr. 2017].

Vensi, Inc. (2017). Benefits of Apache Cordova Cross Platform Mobile App Development. [online] Available at: http://www.vensi.com/benefits-of-phonegap-cross-platform-mobile-app-development/ [Accessed 23 Feb. 2017].

Rajput, M. (2017). Top 10 Reasons to Use HTML5 for Mobile App Development - Developer's Feed. [online] Developer's Feed. Available at: http://www.developersfeed.com/top-10-reasons-use-html5-mobile-app-development/ [Accessed 23 Feb. 2017].

dzone.com. (2017). What are the Benefits of Node.js? - DZone Performance. [online] Available at: https://dzone.com/articles/what-are-benefits-nodejs [Accessed 25 Feb. 2017].

Tilkov, S. and Vinoski, S. (2010). Node.js: Using JavaScript to Build High-Performance Network Programs. IEEE Internet Computing, 14(6), pp.80-83.

Craighead, J., Burke, J. and Murphy, R. (2007). Using the Unity Game Engine to Develop SARGE: A Case Study. 1st ed.

Unity. (2017). Unity - Fast Facts. [online] Available at: https://unity3d.com/public-relations [Accessed 26 Feb. 2017].

Bramer, B. (2014). Blueprint Networking Tutorials. [online] Unrealengine.com. Available at: https://www.unrealengine.com/blog/blueprint-networking-tutorials [Accessed 15 Mar. 2017].

Tudor-Locke, C. and Bassett, D. (2004). How Many Steps/Day Are Enough?. Sports Medicine, 34(1), pp.1-8.

Paczkowski, L. (2017). MonoDevelop Roadmap – Unity Blog. [online] Unity Technologies Blog. Available at: https://blogs.unity3d.com/2015/10/22/monodevelop-roadmap/ [Accessed 10 Apr. 2017].

Martin, C. (2017). Project design report. Unpublished manuscript, Nottingham Trent University.

Dumas, J.S. and Redish, J.C. (1999) A practical guide to usability testing. Exeter, England: Intellect Books.

John Brooke. (1996) SUS- A quick and dirty usability scales (Usability evaluation in industry). Edited by Patrick W. Jordan and Bruce Thomas. Bristol, PA: Florence, Kentucky, U.S.A.: Taylor & Francis. (pg. 189-194)

# Bibliography

Nielsen, J. (2012) How many test users in a usability study? Available at: https://www.nngroup.com/articles/how-many-test-users/ (Accessed: 28 November 2016).

Dunn, T. (2017) Focus groups vs. Usability testing - what, when and why? Available at: http://www.webcredible.com/blog/focus-groups-vs-usability-testing-what-when-and-why/ (Accessed: 28 November 2016).

Bergstrom, J.R. (2013) Moderating usability tests. Available at: https://www.usability.gov/get-involved/blog/2013/04/moderating-usability-tests.html (Accessed: 28 November 2016).

Paulhus, D.L. (1991) ‘Measurement and control of response bias’, in Measures of personality and social psychological. San Diego: San Diego: Academic Press, pp. 17–59.

Callaham, J. (2015) Google says there are now 1.4 billion active Android devices worldwide. Available at: http://www.androidcentral.com/google-says-there-are-now-14-billion-active-android-devices-worldwide (Accessed: 14 November 2016).

IDC: Smartphone OS market share (2016) Available at: http://www.idc.com/prodserv/smartphone-os-market-share.jsp (Accessed: 15 November 2016).

# Appendix A - Lean UX implementation

Adapted from the works of Gothelf and Seiden, (2013).

## Problem statement

Members of the older generation have been proven to receive a greater benefit from taking part in medium intensity exercise when compared to younger adults. A problem exists in which a lack of a routine prevents members of this demographic regularly partaking in medium intensity exercise, the root cause of this can be attributed to a lack general motivation. I shall use this problem statement as a base to create a solution to help produce a novel solution that attempts to provide motivation for older adults to take part in regular bouts of medium intensity exercise.

## Assumptions

### Assumptions questionnaire

I made use of an assumptions questionnaire in order to consider many different aspects of my project and used the questions in order to pull out any assumptions I have regarding my customers, their users and my proposed solution. Each assumption has been given a data tag, this is to make it easier to identify within the prioritisation matrix below in table 1.

#### Business assumptions

**BA1 - I believe my customers have a need to:**

Be motivated to take part in physical activity.

Enjoy their free time.

**BA2 - These needs can be solved with:**

An application the combines an enjoyable activity such as a game with an element that encourages the user to take part in some form of activity.

**BA3 - My initial customers are (or will be):**

My initial customers will be older adults (35+).

**BA4 - The number one value a customer wants to get out of my service is:**

The ability to enjoyably pass the time.

**BA5 - The customer can also get these additional benefits:**

To be given some motivation to take part in physical activity.

To inform them on their progress towards fitness goals.

To give feedback on their performance.

**BA6 - My primary competition in the market will be:**

Other technical solutions that offer more in depth experiences targeted at user enjoyment, or applications focused around motivation for user fitness.

**BA7 - We will beat them due to:**

Producing a solution that combines the best elements of both the competitors in order to make a hybrid solution that appeals to members of both audiences.

**BA8 - My biggest product risk is:**

The biggest risk to the project is the potential lack of knowledge my target demographic will have about this solution.

**BA9 - We will solve this through:**

Testing the application with real users, both to improve upon its design and spread knowledge about the solution

Communication with the communities of the applications competitors to try and gather interest in the new solution

Creating a dialog between the project team and their friends and families in order to spread knowledge about the application.

**BA10 - What other assumptions do we have that, if proven false, will cause our project to fail?**

That a solution of this nature will generate any interest from the general public.

The application will provide suitable motivation for the target audience to proceed with a fitness activity.

#### User assumptions

**UA1 - Who is the user?**

Generally older adults but not limited to this, the application will also be target towards the female demographic based on previous work which suggested this was an appropriate demographic to target.

**UA2 - Where does our product fit in to their work or life?**

The product fits into their everyday life, the goal of the product is to provide a general sense of enjoyment to the user, a secondary goal is to encourage the user to push themselves a little further wherever possible to achieve a few more steps in their day, which will ultimately create positive habitual behaviours and improve upon the users overall fitness.

**UA3 - When and how is our product used?**

The product is designed to be used when and where the user feels like using it, as the application will generally take up the users attention whilst in use, the application will be used during the users down time.

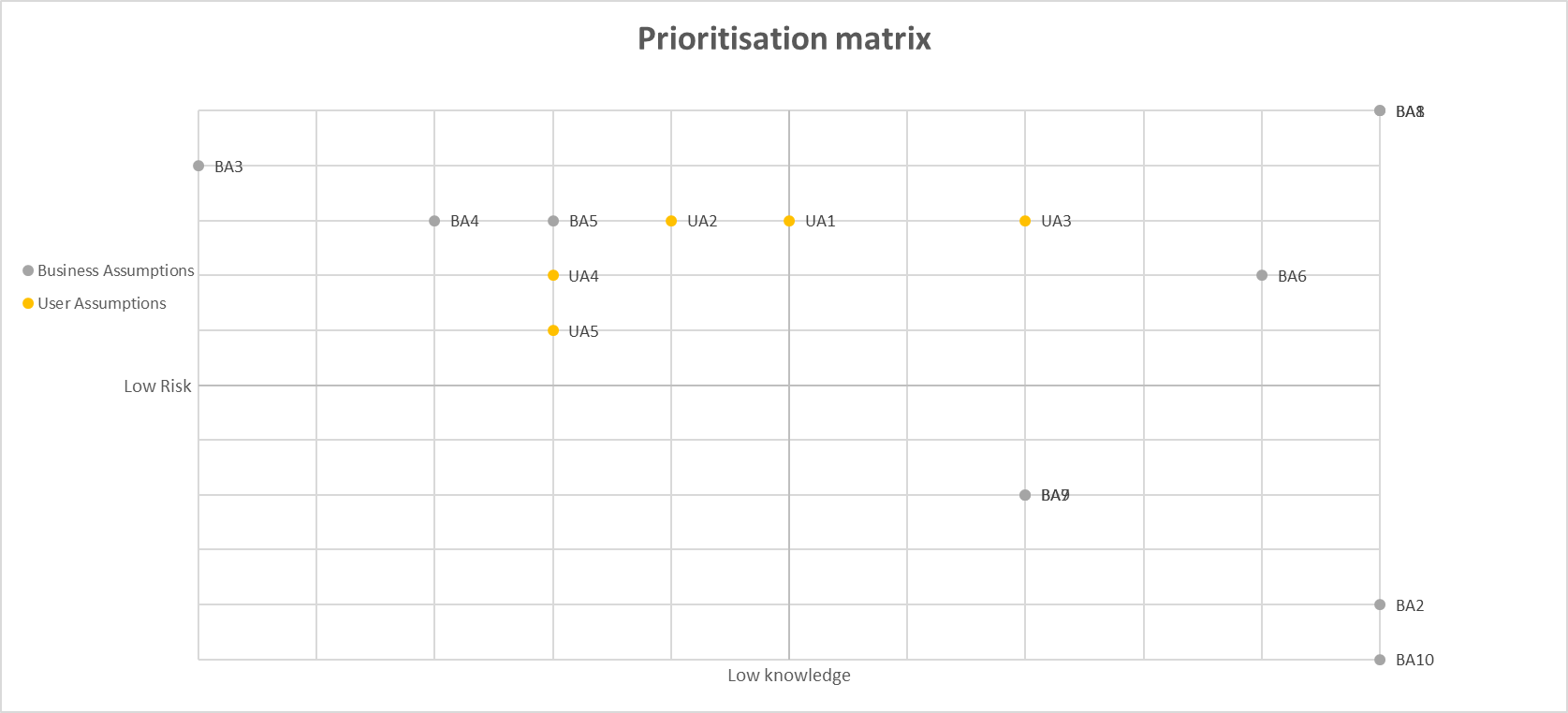
**UA4 - What features are important?**

fitness monitoring of users, enjoyability, accessibility and motivation.

**UA5 - How should out product look and behave?**

The product should look intuitive and be simple to use, its design should be based around the design of the Candy Crush Saga, a popular mobile application among the target audience of the proposed application.

### Prioritisation matrix



High risk

High knowledge

### Prioritised list of assumptions

From the above list of assumptions and prioritisation matrix, I have picked out several which I believe present the highest risk to the success of my project. This has been determined by examining the consequences of whether my assumptions are wrong and how much knowledge I have around the subject that I have based my assumptions on.

* One of my solutions biggest risk of failures comes from the lack of knowledge as to whether the solutions is warranted as well as a lack of knowledge as to whether the application will be able to sufficiently motivate people into taking part in medium intensity fitness. This has been selected as a high-risk assumption as without a general want to participate in the features of the application, customers will move onto applications that better fit their needs.
* The largest risk to the success of the application is the lack of knowledge the target audience have of the solution, it is difficult to complete when competition is so fierce and larger competitors are able to advertise using many channels, this assumption has been selected as high risk as if it is proven correct the application will be doomed to fail as it will not have a userbase in which social interaction can take place
* The applications novel ideas on how to engage with its target audience seem good in practice, however it is difficult to say whether they will be proven true, this is why the testing of the applications novel ideas is a high requirement as without this proof, the applications key selling point will be unwanted by the userbase.
* My customers have a need to be motivated to take part in physical activity and a need to enjoy their free time has been selected as a high priority assumption as this is the general goal of the application, to passively provide motivation to take part in physical activity through a medium that is considered fun by the user base, if this proves to be false the application is doomed to fail

## Hypotheses

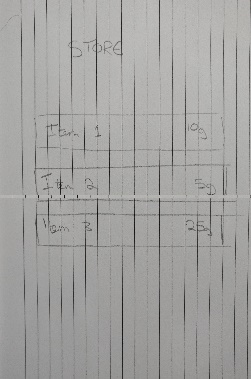
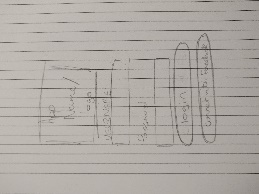
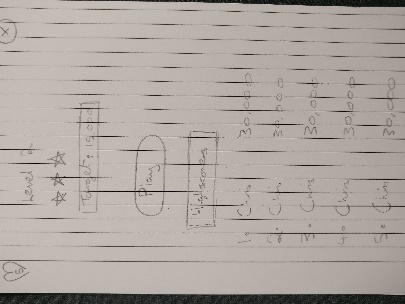
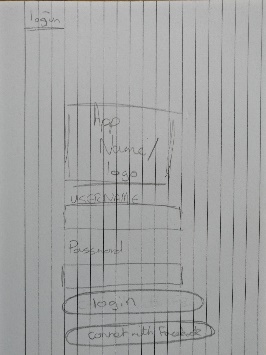
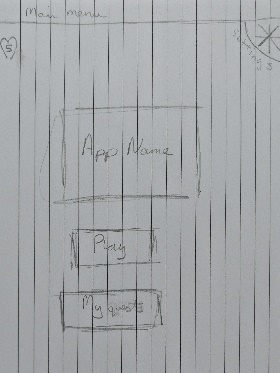
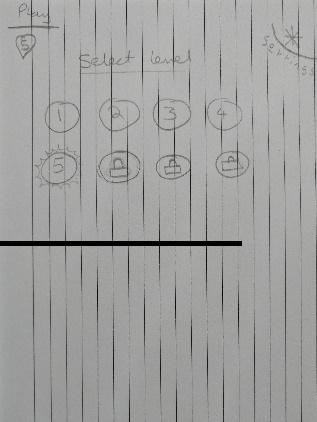
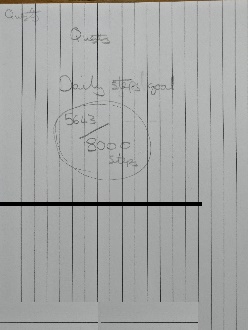
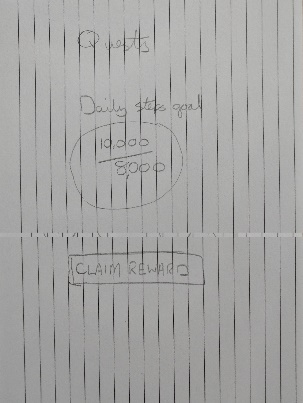
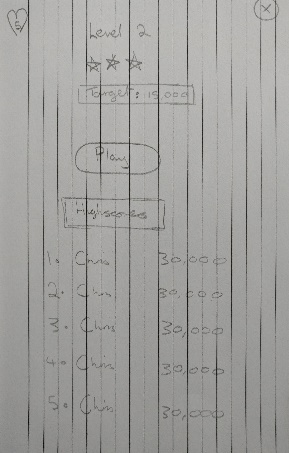
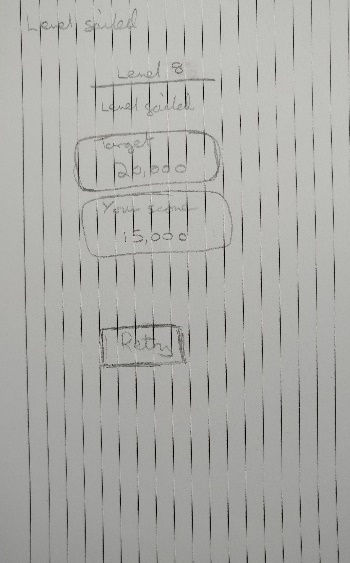
### Hypothesis one

I believe my target audience would make use of an application that aims to produce an enjoyable experience but that offers them the opportunity to gain an in-game rewards if they complete a fitness based challenge. I shall be able to test this by producing a prototype application which combines the above elements, which I can then use to test with members of the target demographic

### Hypothesis two

I believe one of the biggest risks to the success of this project is a lack of knowledge of whether my target audience want an application that aims to achieve the gaols laid out in this report, I can produce a paper prototype that will allow for a rapid testing session with my user base to conform or invalidate this hypothesis.

# Appendix B – paper design



# Appendix C – Usability plan template

Created using the Usability test plan template (Usability test plan template, 2013).

Usability test plan template (2013) Available at: https://www.usability.gov/how-to-and-tools/resources/templates/usability-test-plan-template.html (Accessed: 21 January 2016).

## Overview

This document describes a test plan for conducting a usability test during the development of a mobile fitness game named Sweet Sensation. The goals of usability testing include establishing a baseline of user performance and identifying potential design concerns to be addressed to improve the efficiency, productivity, and end-user satisfaction.

The usability test objectives are:

* To determine design inconsistencies and usability problem areas within the user interface and content areas. Potential sources of error may include:
  + Navigation errors – failure to locate functions, excessive keystrokes to complete a function, failure to follow recommended screen flow.
  + Presentation errors – failure to locate and properly act upon desired information in screens, selection errors due to labelling ambiguities.
* Exercise the application under controlled test conditions with representative users. Data will be used to access whether usability goals regarding an effective, efficient, and well-received user interface have been achieved.
* Establish baseline user performance and user-satisfaction levels of the user interface for future usability evaluations.

The application is targeted at the older generation, users who are likely to want to play games such as Candy Crush. Usability testing will be conducted on who fit this profile (2) and users who do not (4), in total six participants will take part in the study.

## Executive Summary

The goal of this usability analysis is to test whether the system can be used and navigated easily and that functionality within the application makes sense to the user.

Specifically, participants will be tested to see whether they can achieve a high score within the game and then view it on the game leader board, gain an in game reward through the use of the applications fitness elements and whether they can Purchase an in game reward and clearly identify what it was they purchased.

Upon review of this usability test plan, including the draft task scenarios and usability goals for the Sweet Sensation, documented acceptance of the plan is expected.

## Methodology

There will be six participants taking part in a study over two sessions conducted at the premises primarily used for the design of this product and an offsite location, users will interact with the system through the use of a mobile application that will be installed on a number of test devices. The test aims to collect data on user demographic, satisfaction with the system and collect any suggested improvements that could be made.

## Participants

Six participants will take part in this trial, some have been chosen due to their knowledge of the use of similar applications, others due to their knowledge of software design principles and a final subset of users were selected due to them being in the target demographic the application is aimed at. The participants' responsibilities will be to attempt to complete a set of representative task scenarios presented to them in as efficient and timely a manner as possible, and to provide feedback regarding the usability and acceptability of the user interface. The participants will be directed to provide honest opinions regarding the usability of the application, and to participate in post-session subjective questionnaires and debriefing.

## Procedure

Participants will take part in the usability test. Several test mobile phones hosting the application will be used in a typical home environment. The participant’s interaction with the application will be monitored by the facilitator seated in the same location.

The facilitator will brief the participants on the application and instruct the participant that they are evaluating the application, rather than the facilitator evaluating the participant. Participants will sign an informed consent that acknowledges: the participation is voluntary, that participation can cease at any time. The facilitator will ask the participant if they have any questions.

Participants will complete a pretest demographic and background information questionnaire. The facilitator will explain that the amount of time taken to complete the test task will be measured and that exploratory behavior outside the task flow should not occur until after task completion. At the start of each task, the participant will read aloud the task description from the printed copy and begin the task. Time-on-task measurement begins when the participant starts the task.

The facilitator will instruct the participant to ‘think aloud’ so that a verbal record exists of their interaction with the application. The facilitator will observe and log user behavior, user comments, and system actions.

After each task, the participant will answer a post-task questions and elaborate on the task session with the facilitator. After all task scenarios are attempted, the participant will complete the post-test satisfaction questionnaire.

## Roles

The roles involved in a usability test are as follows. An individual may play multiple roles and tests may not require all roles.

### Trainer

* Provide training overview prior to usability testing

### Facilitator

* Provides overview of study to participants
* Defines usability and purpose of usability testing to participants
* Assists in conduct of participant and observer debriefing sessions
* Responds to participant's requests for assistance

### Data Logger

* Records participant’s actions and comments

### Test Participants

* Provides overview of study to participants
* Defines usability and purpose of usability testing to participants
* Assists in conduct of participant and observer debriefing sessions
* Responds to participant's requests for assistance

### Ethics

All persons involved with the usability test are required to adhere to the following ethical guidelines:

* The performance of any test participant must not be individually attributable. Individual participant's name should not be used in reference outside the testing session.
* A description of the participant's performance should not be reported to his or her manager.

## Usability Tasks

The usability tasks were derived from test scenarios developed from the assistance of a subject-matter expert. Due to the short time for which each participant will be available, the tasks are the most common and relatively complex of available functions. The tasks are identical for all participants of a given user role in the study.

The task descriptions below are required to be reviewed by the application owner to ensure that the content, format, and presentation are representative of real use and substantially evaluate the total application. Their **acceptance is to be documented** prior to usability test.

**Task 1 – The user shall navigate through the application and start playing the first available level, upon completion of the level the user will attempt to find their high score**

**Task 2 – The user will attempt to claim the daily award available for reaching a fitness milestone, note for this task the users progress towards the milestone will be set just off from what is required to achieve the award, in order to test how intuitive this is for the user.**

**Task 3 – The user shall attempt to purchase an item from the in-game store with their reward for completing Task 2 above, the user must be able to clearly define what they have purchased, and how this purchase has changed the in game experience.**

## Usability Metrics

Usability metrics refers to user performance measured against specific performance goals necessary to satisfy usability requirements. Scenario completion success rates, error rates, and subjective evaluations will be used. Time-to-completion of scenarios will also be collected.

### Scenario Completion

Each scenario will request, that the participant obtains or inputs specific data that would be used in course of a typical task. The scenario is completed when the participant indicates the scenario's goal has been obtained (whether successfully or unsuccessfully) or the participant requests and receives sufficient guidance as to warrant scoring the scenario as a critical error.

### Critical Errors

Critical errors are deviations at completion from the targets of the scenario. Obtaining or otherwise reporting of the wrong data value due to participant workflow is a critical error. Participants may or may not be aware that the task goal is incorrect or incomplete.

Independent completion of the scenario is a universal goal; help obtained from the other usability test roles is cause to score the scenario a critical error. Critical errors can also be assigned when the participant initiates (or attempts to initiate) and action that will result in the goal state becoming unobtainable. In general, critical errors are unresolved errors during the process of completing the task or errors that produce an incorrect outcome.

### Non-critical Errors

Non-critical errors are errors that are recovered from by the participant or, if not detected, do not result in processing problems or unexpected results. Although non-critical errors can be undetected by the participant, when they are detected they are generally frustrating to the participant.

These errors may be procedural, in which the participant does not complete a scenario in the most optimal means (e.g., excessive steps and keystrokes). These errors may also be errors of confusion (ex., initially selecting the wrong function, using a user-interface control incorrectly such as attempting to edit an un-editable field).

Noncritical errors can always be recovered from during the process of completing the scenario. Exploratory behaviour, such as opening the ski trail while completing a task, will not be coded as a non-critical error.

### Subjective Evaluations

Subjective evaluations regarding ease of use and satisfaction will be collected via questionnaires, and during debriefing at the conclusion of the session. The questionnaires will utilize free-form responses and rating scales.

### Scenario Completion Time (time on task)

The time to complete each scenario, not including subjective evaluation durations, will be recorded.

## Usability Goals

The next section describes the usability goals for Sweet Sensation.

### Completion Rate

Completion rate is the percentage of test participants who successfully complete the task without critical errors. A critical error is defined as an error that results in an incorrect or incomplete outcome. In other words, the completion rate represents the percentage of participants who, when they are finished with the specified task, have an "output" that is correct. Note: If a participant requires assistance in order to achieve a correct output then the task will be scored as a critical error and the overall completion rate for the task will be affected.

**A completion rate of 100% is the goal for each task in this usability test.**

### Error-free rate

Error-free rate is the percentage of test participants who complete the task without any errors (critical **or** non-critical errors). A non-critical error is an error that would not have an impact on the final output of the task but would result in the task being completed less efficiently.

**An error-free rate of 80% is the goal for each task in this usability test.**

### Time on Task (TOT)

The time to complete a scenario is referred to as "time on task". It is measured from the time the person begins the scenario to the time he/she signals completion.

### Subjective Measures

Subjective opinions about specific tasks, time to perform each task, features, and functionality will be surveyed. At the end of the test, participants will rate their satisfaction with the overall system. Combined with the interview/debriefing session, these data are used to assess attitudes of the participants.

## Problem Severity

To prioritize recommendations, a method of problem severity classification will be used in the analysis of the data collected during evaluation activities. The approach treats problem severity as a combination of two factors - the impact of the problem and the frequency of users experiencing the problem during the evaluation.



### Impact

Impact is the ranking of the consequences of the problem by defining the level of impact that the problem has on successful task completion. There are three levels of impact:

* High - prevents the user from completing the task (critical error)
* Moderate - causes user difficulty but the task can be completed (non-critical error)
* Low - minor problems that do not significantly affect the task completion (non-critical error)

### Frequency

Frequency is the percentage of participants who experience the problem when working on a task.

* High: 75% or more of the participants experience the problem
* Moderate: 26% - 74% of participants experience the problem
* Low: 25% or fewer of the participants experience the problem

### Problem Severity Classification

The identified severity for each problem implies a general reward for resolving it, and a general risk for not addressing it, in the current release.

**Severity 1** - High impact problems that often prevent a user from correctly completing a task. They occur in varying frequency and are characteristic of calls to the Help Desk. Reward for resolution is typically exhibited in fewer Help Desk calls and reduced redevelopment costs.

**Severity 2** - Moderate to high frequency problems with moderate to low impact are typical of erroneous actions that the participant recognizes needs to be undone. Reward for resolution is typically exhibited in reduced time on task and decreased training costs.

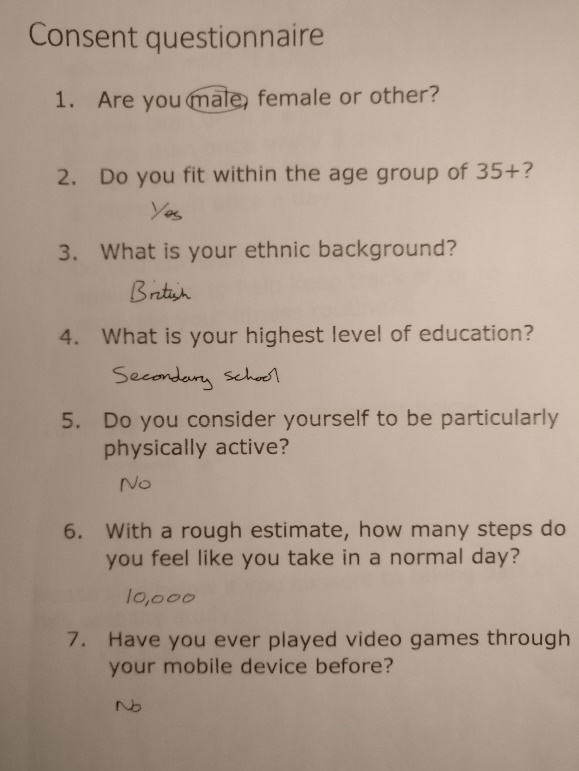
**Severity 3** - Either moderate problems with low frequency or low problems with moderate frequency; these are minor annoyance problems faced by a number of participants. Reward for resolution is typically exhibited in reduced time on task and increased data integrity.

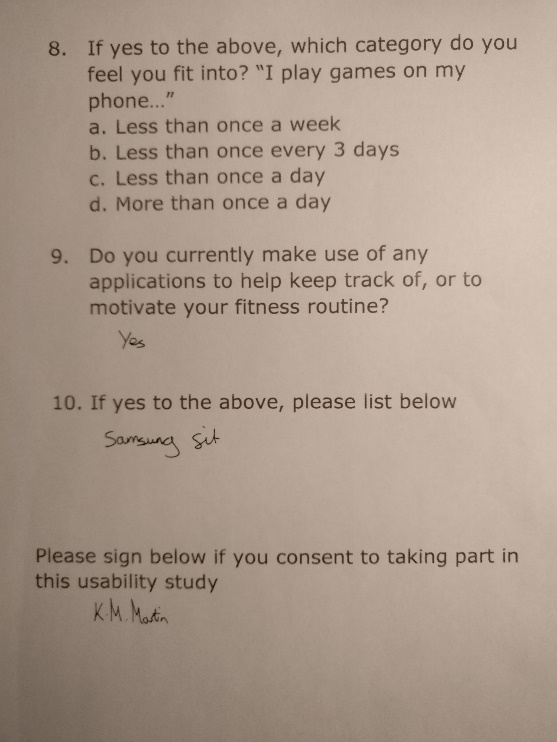
**Severity 4** - Low impact problems faced by few participants; there is low risk to not resolving these problems. Reward for resolution is typically exhibited in increased user satisfaction.

## Reporting Results

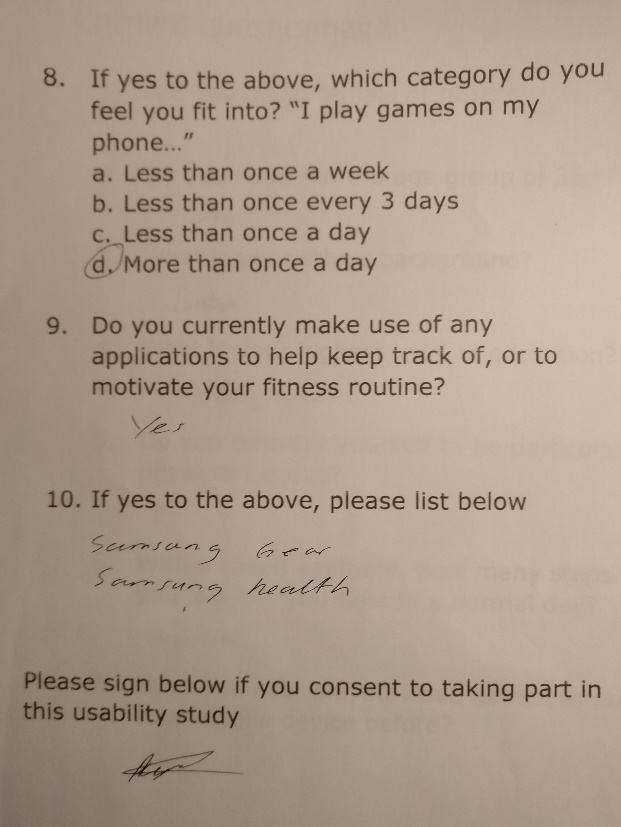
The Usability Test Report will be provided at the conclusion of the usability test. It will consist of a report and/or a presentation of the results; evaluate the usability metrics against the pre-approved goals, subjective evaluations, and specific usability problems and recommendations for resolution. The recommendations will be categorically sized by development to aid in implementation strategy.

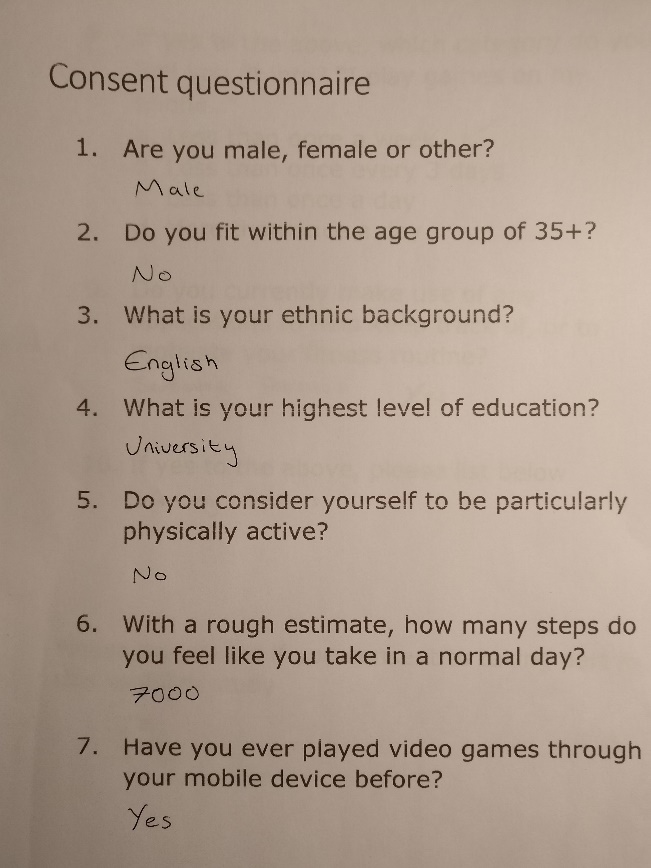
# Appendix D – Usability testing consent questionnaires

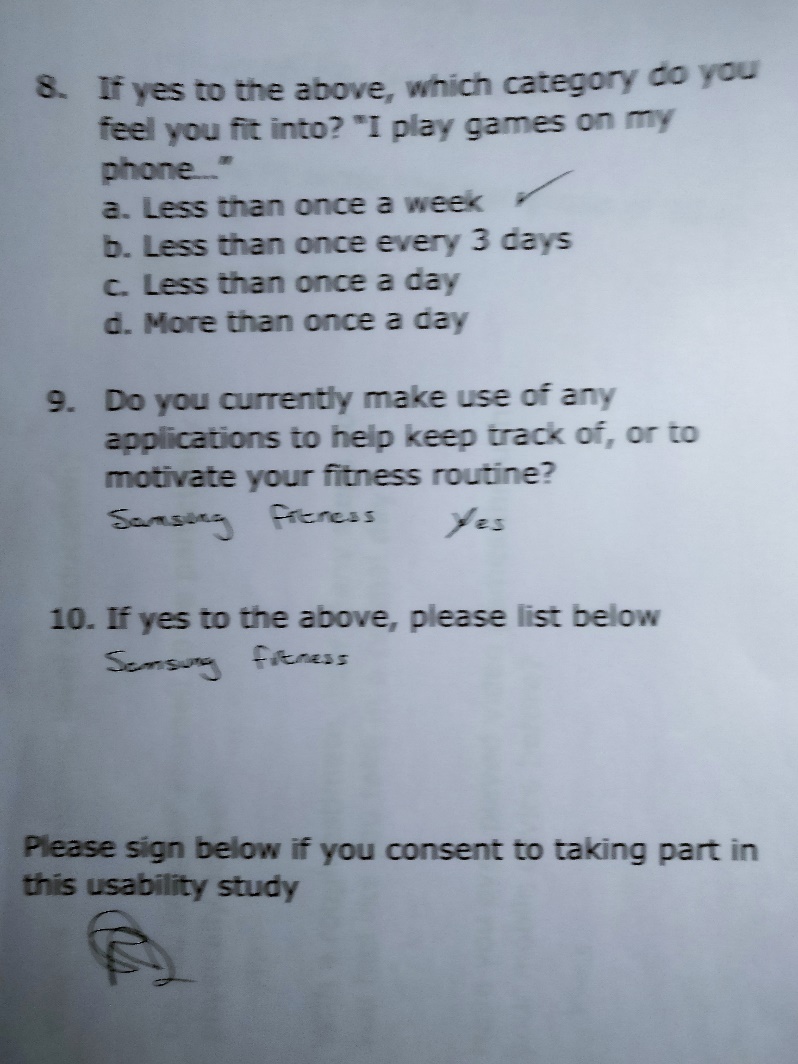


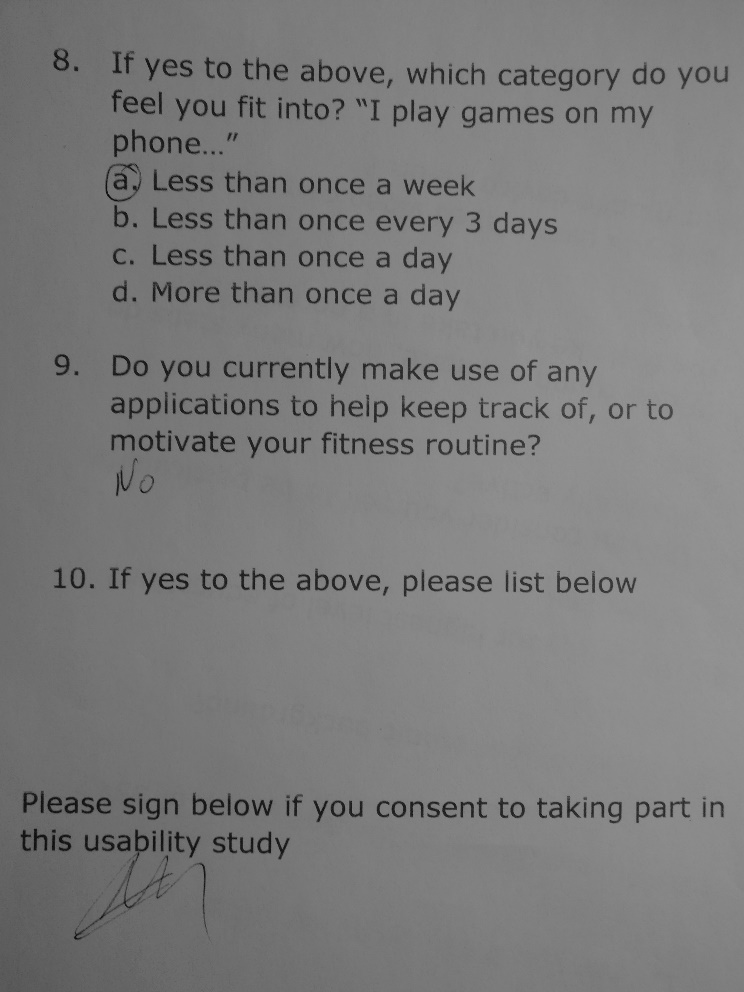
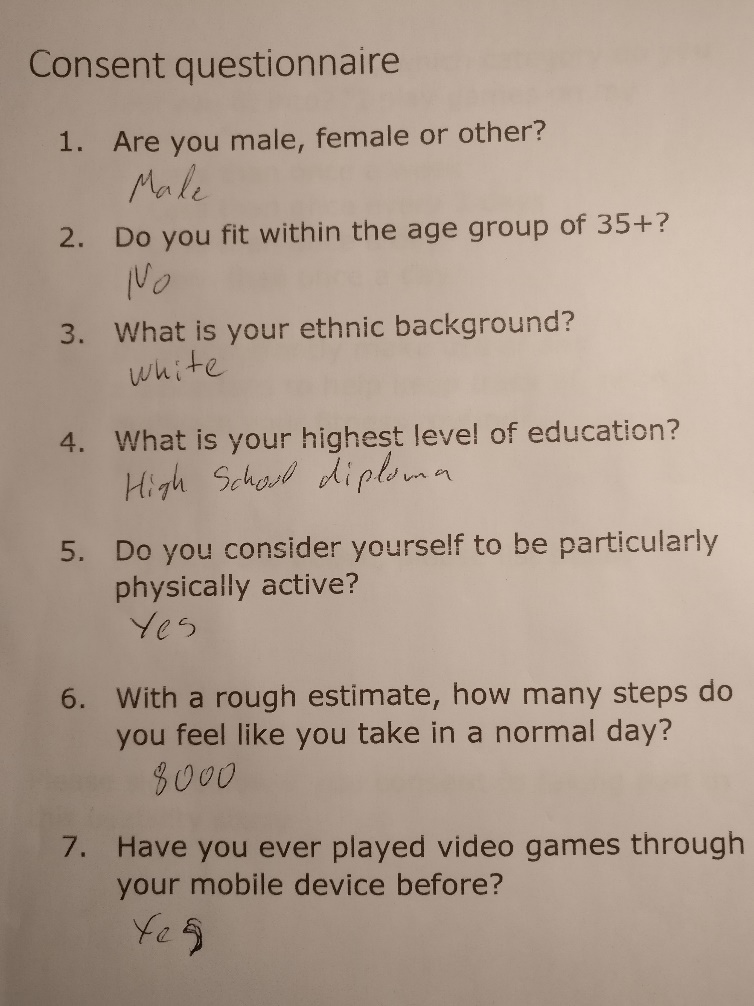


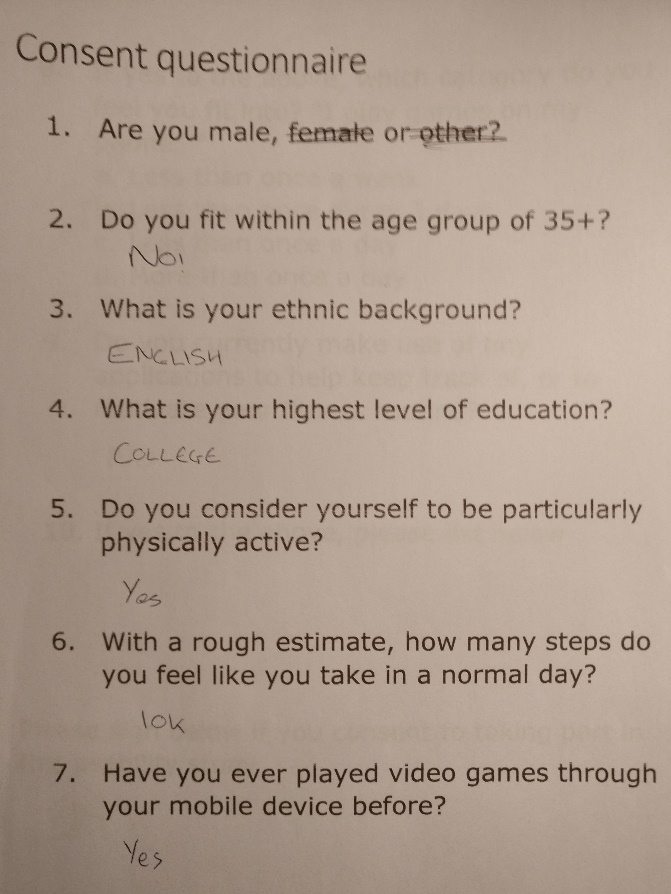


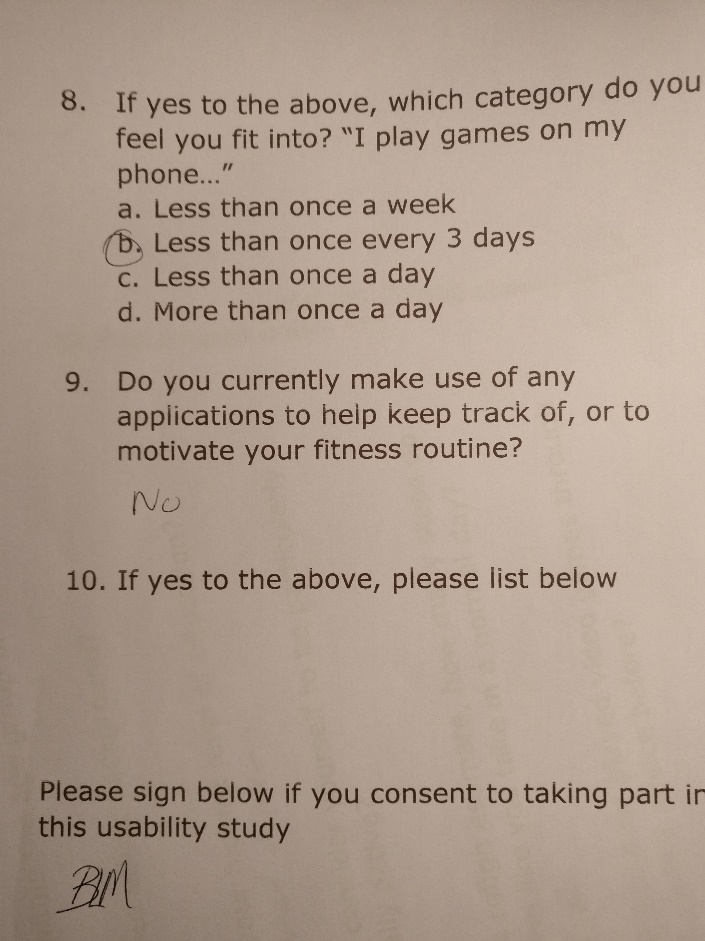


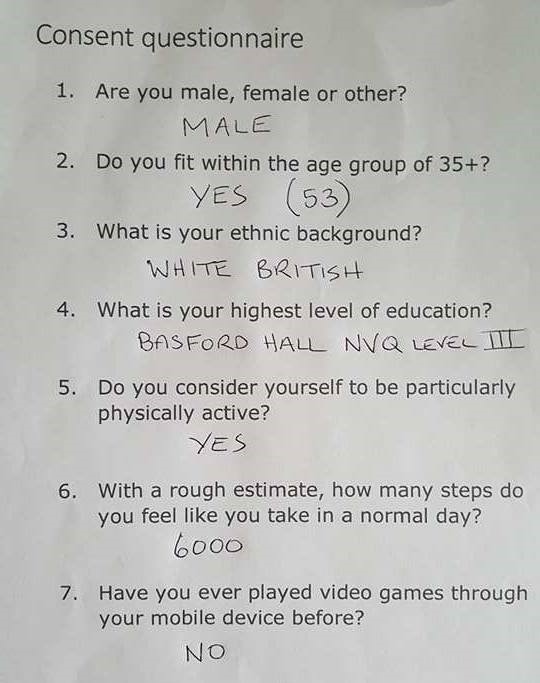


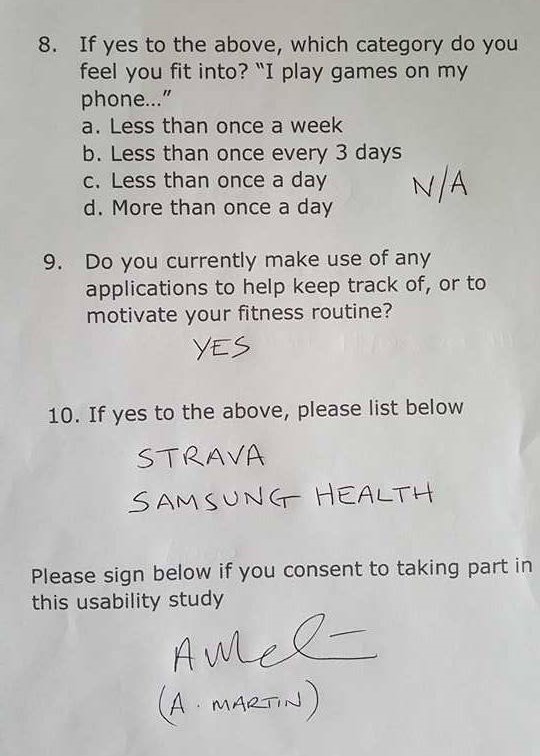












# Appendix E – System Usability Scale results

