School of Computer Science and Informatics, Cardiff University

MSc Computing

November 2021

PoundHelp - Financial Literacy Mobile Application

CMT404 - Dissertation

Author: Joseph Symons

Supervisor: Martin Caminada

Moderator: Carolina Fuentes Toro

Table of Contents

Title Page	**I**
Table of Contents	**II**
Acknowledgements	**III**
Abstract	**IV**
Introduction	**1**
- Aims and Objectives	**1**
- The Problem	**2**
- Literature Review	**12**
Specification	**22**
- Functional and Non-Functional Requirements	**22**
- Use Cases	**24**
Approach	**29**
- Frameworks and Programming Languages	**29**
- Assessment and Quiz Types	**36**
Implementation	**41**
- Code Examples	**42**
- Designs	**55**
Conclusions	**63**
- Conclusions of the Assessment	**63**
- Reflection	**65**
References	**67**

Acknowledgements

This past year and a half have been rather difficult for me to navigate through. It began with me heading for the US as Covid hit Europe, to me returning through a locked-down and confusing airport in San Francisco as the world closed. Nothing has been quite the same since then.

I must, though, thank Covid in some ways. It helped me reassess where I was and has shown me that a real career dawned on me. I worried that being out of work for my master's would be a hindrance, but in fact, I believe that this has been a blessing.

To my housemates this year, I wish I could see you each and every day. Your support has been so unexpected and humbling. I wish I could show my appreciation more.

To my friends and family, Covid helped me realise how much I miss speaking to your day to day and that I need to live a little more than before.

Finally, my parents, thank you for letting me talk through my code with you before quickly realising how it could be solved and leaving mid-sentence to finish my work. You enabled me to succeed this year, and I look forward to the career you've helped me to aspire to.

Abstract

The UK has been unable to deal with the issues of financial illiteracy for some time. Research shows that a significant number of Britons do not have adequate levels of financial literacy. Therefore, a solution is needed which could solve this. My solution is to provide a mobile application called “PoundHelp”, that will allow people of all ages to learn about personal finance. This project considers the business case of such an application and presents a minimum viable product to solve the issue. This project presents the research conducted on the financial literacy situation, current solutions, market place and situation of the target audience. Based on this information, the project explains the steps taken and choices made to create the product, proving that the mobile application this project produced is a suitable solution.

Introduction

This chapter is split into three sections: Aim and Objectives, The Problem and a Literature review. Firstly, the Aim and Objectives give an illustration of the intended outcome of the project. Secondly, The Problem section gives a transparent background to the reader to understand why there is a problem with the topic being addressed, with supporting evidence from a survey I conducted. Finally, the Literature Review is then used to illustrate why my solution to the problem being addressed will succeed in light of the currently available products already in the marketplace, which the application should target and the value of the market it would find itself.

Aim and Objectives

This section will address the main aims of the project and what it seeks to achieve. It will not detail how these objectives have been achieved, as this will be done in the oncoming chapters. However, it will explain the main aim and objective in order for the reader to understand the success of this project better. It will lay out the key objectives in the form of bullet points. It will then detail how these can be achieved and make sure the work produced is relevant. The main objectives were as follows:

- Demonstrate that there is a problem that needs a solution in financial literacy through researching the topic.
- Show that a mobile application is a unique solution by providing a literature review of the current market solutions available to customers and demonstrating that such a product could be profitable.
- To create a product that could successfully solve the problem of financial literacy within the UK by providing information and quizzes to take and providing feedback. It must then show that it meets these requirements as a minimum viable product.
- Justify the critical decisions made during the creation of the product and prove how the following software development frameworks enabled the successful implementation of the product.

By providing a suitable literature review that offers enough depth of current solutions for the problem being addressed, the document should provide a crystal-clear understanding of the market and therefore give an opinion on why the current solutions are not effective in solving the current problem. It will show that the current mobile application market would welcome the solution that has been proposed and therefore indicates that a product could be profitable if launched in its intended app store.

The main aim of these objectives is to successfully create a suitable product that can successfully solve the problem that is being discussed. It needed to be an application to take quizzes based on provided educational material and receive feedback to help them learn. Due to the time frame given, a more thorough application would have been impossible to implement. By using a minimum viable product, all essential requirements were met to sufficiently demonstrate that the application could successfully solve the problem of financial illiteracy.

The project will also discuss the key reasons that were made in the creation of the product. Discuss the programming the program was written in, the software framework and designs used to create the product. Helping the reader understand how successful the management of the development was.

The Problem

This section is dedicated to presenting why there is a problem in financial literacy, financial literacy is the ability to “understand and effectively use various financial skills, including personal financial management, budgeting, and investing.” (Fernando, 2021). The section will do this by discussing the contextual situation the UK finds itself in, how government and global economic issues can affect everyday people and a survey conducted to support the findings of the research. This segment therefore will highlight the issues with financial literacy in the UK. Showing how the problem affects a diverse demographic of people. It will do this by presenting research from a survey that was conducted as part of its project and its results.

Pensions

Recently, as had been warned on several occasions about an eventual suspension (Elliott, 2020), on 7th September 2021, in a speech to the House of Commons, Prime Minister Boris Johnson's government decided to break an election manifesto promise, which was not to break the Pension triple-lock (Johnson, 2021). A scheme established in 2010 that offered to increase pensions based on either:

- The annual average wage increase.
- Inflation (measured by the Consumer Prices Index).
- 2.5 per cent.

The highest figure would then dictate the increased fund in the national pensions (Finnis, 2021). It has been argued that the triple lock would eventually be ended altogether (Emmerson, 2020). However, despite election promises, Boris Johnson has let pensioners down by this suspension to this system.

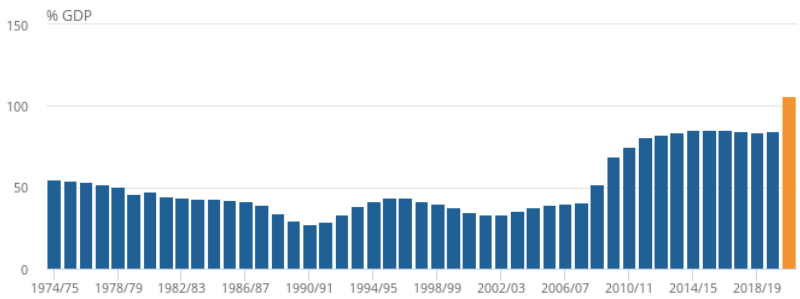


Figure 1, General government debt as a percentage of gross domestic product (GDP), UK, March 1975 to March 2021.

Source: Office for National Statistics - UK government debt and deficit. (www.ons.gov.uk, 2021)

The suspension was due to a rise in the wages in the UK; due to difficulty finding staff, many companies had to increase their salaries to attract labour to fill positions rapidly. As a result, in hospitality, for example, wages increased well over 5% in June compared to the second quarter of 2019, and other similar increases occurred in other industry areas (Cherry, 2021). The result was wage inflation of almost 9%. The election promises became impossible for a government that was already borrowing substantial amounts of money due to the COVID-19 pandemic. According to the Office of National Statistics, general government gross debt as a percentage of gross domestic product was 106% of GDP in 2020-2021 compared to 84.4% the prior year, as shown in Figure 1 (ONS, 2021).

A large portion of this money went into the economy through Furlough Schemes and loans to companies to survive (Barnes and Megaw, 2021). Therefore, in due course, consumer prices will be impacted by such a massive influx of money into the economy. The Bank of England governor Andrew Bailey has recently highlighted wage increases as one major factor that may lead to longer-term inflation rises (Saker-Clark, 2021). However, this is not the first-time pensioners have been let down by the UK government.

For example, approximately 200,000 women were granted a £2.7 bn' top-up' to their underfunded pensions earlier this year. The top-up came after the government uncovered systematic underpayments that went unpaid for almost 30 years (Cumbo, 2021). In the event these women were reliant on the state for their retirements and had been struggling without them. They could have been in severe financial and emotional difficulty due to their inability to maintain their lifestyle. Statistics show that tomorrows pensioners will be struggling as well.

Over a fifth of UK citizens have no pension savings, and around 17% of over fifty-fives have admitted to having no pension savings other than the state pension (Green, 2020). Four in ten pension pots that were accessed between 2018-2019 have worryingly less than £10,000. Over a fifth of UK citizens believe they will never retire (Raisin, 2020). A typical pension pot after retirement rests at £61,897 (Financial Solutions Team, Telegraph, 2020; FCA, 2018). The pot would provide £3,000 extra per year from 67 years onward. Putting this together with the maximum State Pension, which works out at over £12,000 a year (Financial Solutions Team, Telegraph, 2020), means that pensioners yearly budgets would be about £15,000, which is just enough for a basic retirement, which looks poor when considering the knock-on effects of a world with far higher inflation.

During the prime minister's announcement, other financial circumstances were changed, such as a rise in National Insurance (NI) of 1.25% (Johnson, 2021). For the first time, pensioners still earning after retirement are now charged NI if they are earning. Meaning a million pensioners would now face taxation during retirement (Brennan, 2021).

Working Population, Debt, Gambling and Benefits.

Nevertheless, it was not just pensioners who faced the wrath of this change but the working population too. These changes will mean that graduates with student debt will pay nearly half of their extra income in income tax, national insurance and student loan repayments (Harley-McKeown, 2021). Additionally, these tax changes shall also obviously affect anyone of working age who has a job. It would be essential to consider how the UK population can handle hikes in taxation and, retrospectively, any financial changes from taxes to inflation in goods and services that could incur shortly and shock people.

Debt is another issue that is affecting those who live in the UK. Although paying for things with credit can be beneficial for credit scores or perhaps as a simple temporary payment. If unable to pay back these debts, the effects can be apocalyptic. One example is Wonga, a payday loans company that charged high-interest rates, over 5000% Annual Percentage Rate (APR), on their debt. Some figures indicated that over 23% of its clients went into financial disaster at one time, one person even killing themselves due to their debt (Collins, 2018). It is not just from loans that people find themselves in difficulty. Gambling has a massive part to play in the financial difficulties of Britons.

The Gambling Commission indicated that 1.1% of all respondents identify as either problem gamblers, which is a person who is “Gambling with negative consequences and a possible loss of control”, or moderate gamblers, “Gamblers who experience a moderate level of problems leading to some negative consequences” (Barnfield-Tubb and Atkinson, 2021). Although 1.1% would be considered a tiny fraction of gamblers in general, the significance cannot be underestimated. According to the House of Lords report “Gambling Harm — Time for Action”, the report found that three hundred thousand Britons are “problem” or “disordered” gamblers. It is then estimated that for each of these problem or disordered gamblers, six others are impacted by them in some way, a total of around two million people (Kropshofer, 2020. Gambling Harm — Time for Action, 2020).

The people around these gamblers are affected by several adverse effects from these problems. For example, research has shown that partners and ex-partners of those with gambling problems had not just their “…financial security, their emotional, mental and physical health…” affected, some “…lost their savings, homes, belongings and established

ways of life” (Holdsworth et al., 2013). The gamblers themselves experience many adverse effects of their problems. According to the Royal College of Psychiatrists, potential harms from problem gambling are as follows:

- **Financial harms:** Overdue utility bills; borrowing from family friends and loan sharks; debts; pawning or selling possessions; eviction or repossession; defaults; committing illegal acts like fraud, theft, embezzlement to finance gambling; bankruptcy; etc...
- **Family harms:** Preoccupied with gambling so normal family life becomes difficult; increased arguments over money and debts; emotional and physical abuse, neglect and violence towards spouse/partner and/or children; relationship problems and separation/divorce.
- **Health harms:** Low self-esteem; stress-related disorders; anxious, worried or mood swings; poor sleep and appetite; substance misuse; depression, suicidal ideas and attempts; etc...
- **School/college/work harms**: Poor school, college or work performance; increased absenteeism; expulsion or dismissal.

(George and Bowden, 2014)

It is unfortunate to say that statistics show that those who are in debt struggle. Problem debt is summarised as "…when someone becomes unable to pay their debts or other household bills" (National Audit Office, 2018). According to The Money and Mental Health Institute's 'A Silent Killer: Breaking the link between financial difficulty and suicide' report in 2018, 13% of those in problem debt have thought about suicide, and 3% have attempted suicide, this means that approximately 400,000 people considered suicide in England last year and 100,000 people attempted suicide. Additionally, 23% of people who attempted suicide were in problem debt (Bond and Holkar, 2018).

It would now be beneficial to look at how others are struggling, such as those receiving Universal Credit. Universal Credit is an all-in-one payment system that has replaced such benefits or credit and supports child tax credit and housing benefit payments. The amount of money that individuals receive is dependent on a case-by-case basis. Considering the raising of national insurance by Boris Johnson, these people are most likely to suffer the

most. As Labour leader Keir Stammer tweeted, referencing an article in The Independent, "Two and a half million working families will face a double whammy of a national insurance tax rise and a cut to Universal Credit…" (Keir Starmer, 2021; Cowburn, 2021).

Starmer is correct. Many such people will also be struggling soon. Although the jobs market has been recovering since the Covid-19 pandemic, 2.4 million people are still under the furlough scheme as of 31st May 2021 (Powell and Francis-Devine, 2021). Following the curtailment of the furlough scheme in September, there are expectations for the unemployment rate to rise rather than continue being employed, which would affect many (Sierra, 2021). There are currently six million people who currently claim Universal Credit. Recently, the government has decided that Universal Credit should be cut by £20 a week to drive people to look for work. However, statistics show that 40% of these claimants are employed already. Before the pandemic, 14.5 million people were in poverty. 57% of these were workers (Klair, 2021). We can assume then that by the end of the furlough scheme and decrease in Universal Credit, alongside the rise in taxation and inflation, would increase the unemployment rate and raise the UK's poverty levels, making many households suffer. The Contextual background of the personal finance problem – Financial Literacy.

Financial Literacy

This year, Freetrade launched a survey testing respondents' financial literacy. Among their results were some alarming statistics. The pass rate for questions about retirement was 20%, a very worrying statistic. While 88% of respondents felt unconfident with their grasp of financial literacy, a third said their mental health had been affected by dealing with their finances. However, a massive 45% of those between 18-24 said their mental health had been affected. Despite their years of experience, the pass rate for over 55s was 55% while those between 18-24 was just under 50% (Lane, 2021).

According to the Financial Times, one-third of Britons are financially illiterate (Pickford, 2015). Additionally, a report commissioned by MoneyHelper (formerly Money Advice Service) in 2018 entitled "Adult Financial Capability Survey" found that over ten million people rarely or never save money (Money Advice Service, 2019, Financial Capability Survey, 2018). Consequently, eleven-and-a-half million have less than £100 in savings. It is

also estimated that nine million adults borrow money to buy food or pay bills (Financial Capability Survey, 2020). Another issue is due to the behaviour of UK Citizens. Behavioural economists have shown that people tend to make negative and poor choices, mainly due to individuals struggling to envision the future benefits of saving in the immediate short term (Poterba, Venti and Wise, 2012; McConnell, 2013; James, Price and Buffel, 2020).

It is worth noting how Britons are educated on finance. One report interviewing 15–18-year-olds by the London Banking Institute of Banking & Finance entitled found that 75% of respondents learnt their financial literacy from their parents. This could be detrimental to their learning if the statistics show that the pass rate of over 55s was only 55%, the knowledge they are receiving may be flawed (LIBF, 2021. Lane, 2021). The report also showed that 83% of respondents would like to learn their financial education from their schools. However, other statistics show that only 4/10 pupils have had some financial education in school despite this. Indeed, financial literacy is on the curriculums in England, Northern Ireland, Scotland and Wales. Despite these schools wanting to increase their time for financial education, they are unable to due to timetables, curriculums and a lack of skills and knowledge (libf.ac.uk, 2021. Financial Capability, 2021).

Survey

In order to better understand the arguments presented regarding the problem, I produced a survey that would help better understand the need for the application with my own data. The purpose of the survey was to ask a range of questions. Particular focus was set on the confidence of the participants understanding of their overall personal finance understanding and then more specific questions about a few facets of their knowledge. After these clarification questions, the same question of their general confidence was asked to check for deviations. Additionally, the survey also asked some marketing questions. Figures 2 and 3 give a summary of the participants' demographics. As the figures indicate, the participants' demographics by gender were fairly balanced; however, the figures were heavily youth-focused in terms of age ranges.

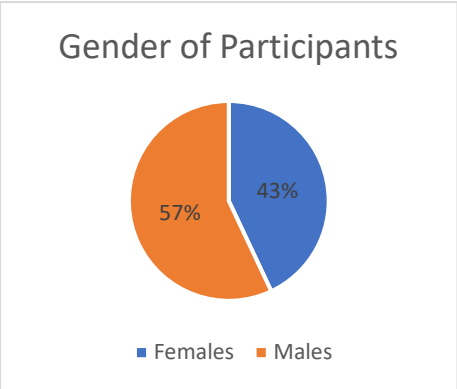


Figure 2, Pie chart indicating gender of participants (n=21).

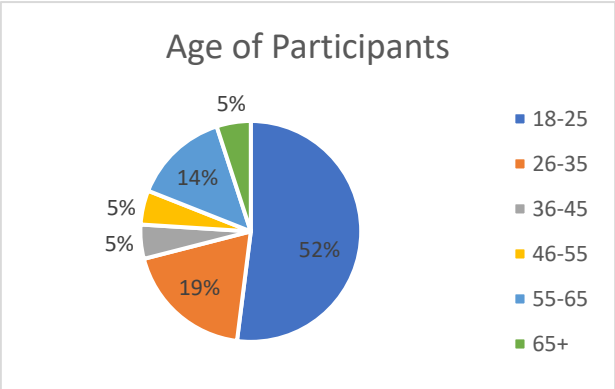


Figure 3, Age ranges of participants (n=21).

The main aim of the survey was the discussion of user's confidence in their financial literacy. The users were asked "On a scale of 1-10, 1 being unconfident, 10 being confident how competent do you believe your financial literacy to be? 1 Being Unconfident, 10 being Confident", They were then asked to rate their understanding of specific aspects of their financial literacy such as their knowledge on Everyday Money, Savings and Pensions & Retirement among others.

Notably, the research of this project has shown that gender is a strong indicator of confidence in financial literacy. The deviation between the results of Male and Female confidence shows that men are significantly more confident than women. Although this project will not speculate on the causes for such results, evidence suggests this data follows research that has preceded this projects research (Bucher-Koenen et al., 2021). As Figure 4 indicates, prior to the clarification questions, Males felt pretty confident about their understanding of financial literacy. They averaged a confidence result of 6.83/10 before the clarification questions were asked and then 6.92/10 after the clarification questions a very slight increase. While, comparatively, Females were significantly less confident. Female's results were 4/10 before the clarification questions and then 4.5/10 after. Much less than the Males scores, which showed that the difference of confidence was initially 2.83 and then 2.42. Despite showing a greater increase in their confidence females still fell well short of the men, being in the bottom half of the confidence levels.

The survey also conducted research based on these clarification questions, asking for the respondents' general understandings of topics that would be the basis of most

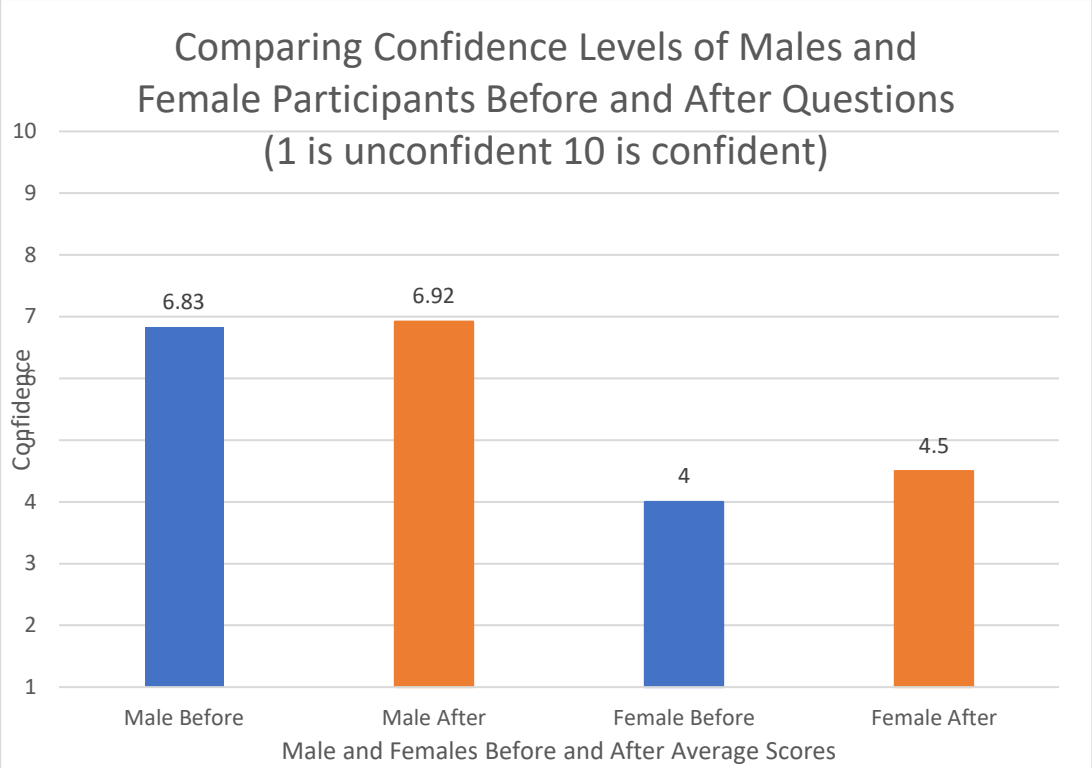
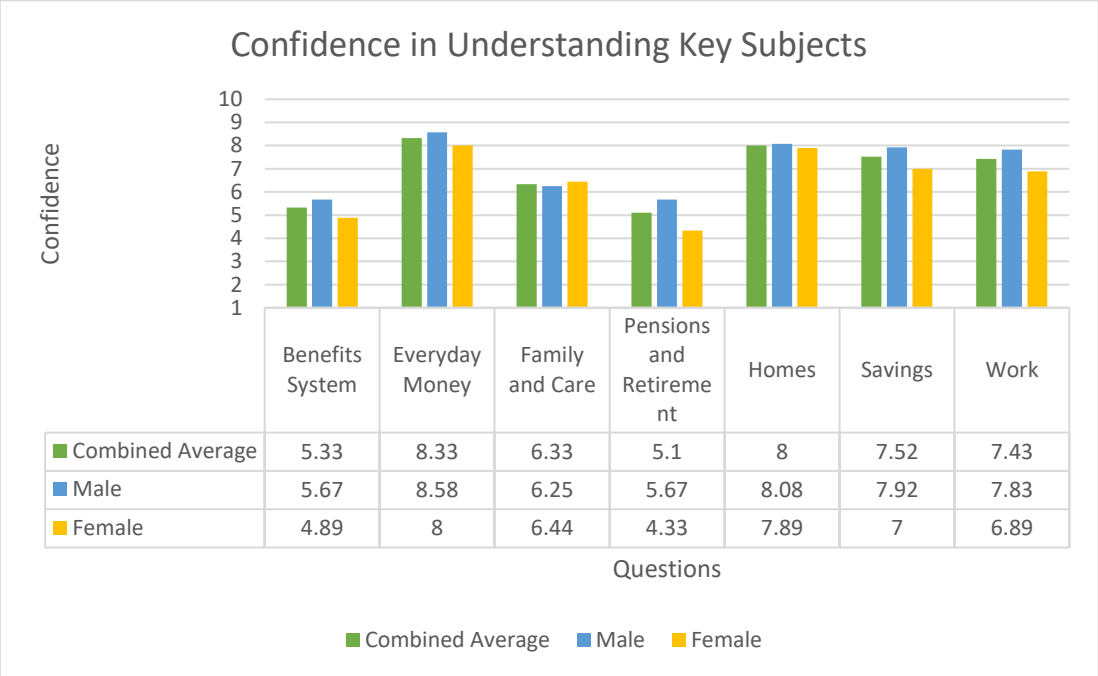


Figure 4, Comparing confidence levels of males and females



	Benefits System	Everyday Money	Family and Care	Pensions and Retirement	Homes	Savings	Work
Combined Average	5.33	8.33	6.33	5.1	8	7.52	7.43
Male	5.67	8.58	6.25	5.67	8.08	7.92	7.83
Female	4.89	8	6.44	4.33	7.89	7	6.89

Figure 5, Comparison of confidence levels of clarification questions among males and females and a combined average.

understandings. Rather than asking questions asking for mathematical problem solving, this project aimed to understand respondents' comprehension of typical subjects such as savings. These understandings are shown in Figure 5. As Figure 5 shows, Male's scored better in almost every category, women scored better in the Family and Care

understandings. The most remarkable difference in confidence was in Pensions and retirement, where Male's scored 5.67/10 compared to females 4.33/10. Males and Females had the closest confidence in their knowledge of Homes and Family. Another Interesting fact that this data shows is that both men and women align their top scores. Both genders top four results matched one another, where Everyday Money, Homes, Savings, Work, Family and care. Males scored the same for their least confident answers, Benefits and Pensions & Retirement.

Figure 5 also demonstrates that some of the prior research supports these findings. As demonstrated in The Contextual Problem of UK Finances section, pensions are one of the worst understood aspects of personal finance. Notably lower than the best performing category, Everyday Money, by over 3 points. However, unlike some of the statistics presented, the financial literacy of the UK would seem to be reasonable. Most people were somewhat confident and, after clarification, felt more confident than they had been. This is most likely due to the small sample size taken.

Problem Conclusions

What this segment has shown is that there is plenty of problems that are facing Britons. The broader economic issues are presenting serious issues to households. Decisions in government are affecting a wide range of society negatively. Gambling and debt are causing a great deal of damage to the personal lives of those suffering from them. In order for these people to be able to compensate for these changes, there is a need to improve the financial literacy of the UK. As the survey has demonstrated, the financial literacy of the UK is still not perfect. Some areas lack more than others.

Nevertheless, things can and therefore should, be improved. Diane Maxwell, the former leader of New Zealand’s state-backed financial capability drive, said: “Improving financial capability can be transformative for individuals and families… People report better sleep, feeling more in control, greater family cohesion, and are more likely to think long term. In that sense, it has a powerful upward momentum to it.” (Jenkins, 2021). A suitable solution is needed to address the various consequences surrounding the financial literacy problem appropriately.

Literature Review

This section will provide a literature review to help establish the need for a product in the financial literacy market. It will initially discuss the current products in the target market while discussing their advantages and disadvantages. It will consider the financial aspects of the market it will target and its potential profitability. Then, it will consider the target audience of the product when it is produced.

Competitors

As will become apparent, despite many financial assistance solutions, there are no mobile applications that offer financial literacy education and information to resolve the problem at the time of writing. However, many websites inform their visitors about personal finance, which are seen as the main competitors for this project. These competitors are considered so because were they to enter their own mobile applications, and then market share could be lost, to such an extent that the application would no longer be a profitable venture.

If the application is to be a success, it must move quickly to establish itself as a market leader to dominate both or either of the Android or Apple app stores. This application should become highly profitable and beneficial to society. Therefore, this review will demonstrate the competitors in the market, acknowledge their strengths and weaknesses and consider an array of different competitors for various reasons.

Investopedia.com – Website

Investopedia is a website that provides financial information on many topics focussed on investments. It receives over 60 million visitors a month on its platform (Similarweb, 2021a). One of its most significant features is a dictionary where a user will search for a particular word related to finance to understand more about it. Once found, they can read an extensive article explaining it.

Investopedia also offers educational features to learn more about topics like the economy, allowing users to learn through articles provided. It should be noted that some of these

educational products are behind paywalls on an individual basis. Investopedia would be considered a rival in the broader context. It provides valuable information that potential customers could use instead of the application this project will be implementing.

However, Investopedia is designed for the American user as opposed to the UK or any other nation. Only around 5% of its monthly users come from the UK (Similarweb, 2021a). It only provides relevant information to Americans, such as American pension plans and taxes, meaning that some of its information would misinform a British user. It charges additional premium fees in its 'Academy' tutorial for learning, meaning that a paywall effectively stops willing learners from accessing information that could help their financial literacy.

Therefore, this mobile application would be an alternative to this website, as unlike this website, it focuses on the British financial system. It needs to be connected to a network in order to be displayed. It also has to compete with other websites on Google for the top spot in related searches. Therefore, a mobile application would benefit users by reducing the number of clicks they need to make before receiving the information they want and not relying on a network connection.

MoneyHelper – Website

MoneyHelper (MH) is an organisation created by the UK government and funded by a levy on the financial services industry. It is a site that supplies financial information to its visitors, focusing entirely on UK personal finance. The information is free and impartial. Additionally, it covers financial topics such as debt & borrowing, pensions & retirement and budgeting & saving (moneyhelper.org.uk, 2021).

MH is a site that should draw many visitors. Unfortunately, it does not; Similarweb does not currently provide information on websites that do not attract more than 50,000 visitors (Silverweb, 2021c) . At the time of writing, the site has not received the required views to compare. Likely because of the rebranding, it was formerly known as Money Advice Service, which shows that the interest in this site is lower than other sites in this review., when the site was known as Money Advice Service.

Prior to this rebranding, the website itself was abysmal. It looked outdated, and the navigation was very clogged, with many links from drop-down menus being displayed. However, the site now looks very impressive. It now has a sleeker design with excellent use of colour coordination. The site now also incorporates other services called the Pensions Advice Service and Pension Wise, meaning that the site offers much more information to a broader demographic of visitors.

It is hard to gauge the impact of the site without statistics of its performance; this is a crucial indicator. As Similarweb is unable to process the site due to a lack of viewership, we can conclude that there is confusion among those seeking out help from this platform who are unaware of the change. The rebrand is most likely making these visitors go to other information sources. Another issue this site would have attempted to launch a mobile application such as this project produces is that changing tact can be slow as a government-backed organisation. Meaning that a successful launch of the project's application would mean that entry into the market for a MoneyHelper product could be slower than its other rivals, giving it a lower market share.

Therefore, this project can take note of the sound design principles of this site while considering the information it provides as an excellent example of what a user of the mobile application would need. However, its lack of visitors demonstrates that not getting the good word out about a change in branding can negatively impact the product's success. Therefore, this project must focus on successfully branding its product to stand out and retain its users.

MoneySavingExpert.com – Website

One major competitor to take note of is MoneySavingExpert (MSE). The organisation, founded by Martin Lewis, who is a well-known journalist in the British financial world. In 2015, 71% of Brits trusted his thoughts on Europe more than any other public figure in a public pole before Brexit. His site receives 22 million views a month, and his weekly Money Saving Expert's Money Tip's blog is sent to 13 million people (Similarweb, 2021c. Cohen, 2019).

The website and blog contain helpful information for users, especially personal finance, often providing the typical user's most essential and relevant information, designed for the British public. This approach is something that this application would benefit from, instead of long articles about financial news that is irrelevant for most people. Instead, this website focuses on what a British consumer needs to know, such as the costs associated with first-time buyers of houses.

The volume of readers to the website and blog makes MSE a severe competitor. It dwarfs almost all other UK competitors. Fortunately, like the other websites, MSE does not have a mobile application. The project will benefit from a portion of the market that they do not currently dominate. Considering the proportion of the British public who own smartphones, this application has a better chance of success due to making the information shorter than browsing the web.

Other factors to know about MSE is its profitability, MoneySuperMarket which trades publicly, owns 100% of the company. Therefore, research was conducted on the reports they have offered. Worryingly, MSE generates around 30 million pounds of revenue a year. It also has cash reserves of almost £100 million (Annual Report 2020, 2021). Meaning that, should they choose to create a rival product, they could efficiently utilise their financial strength to take market share.

However, one crucial factor in this project's success is that MSE does contain much information about personal finance. For example, it offers information on the best utilities that a consumer can consider when switching. This type of information is not relevant to what this application will offer, which will mean that the application will be more succinct and offer services that are different from MSE even if it should launch its mobile application.

IG Academy – Mobile Application

IG Academy is a mobile application that teaches its users how to invest in 'day trading', which is currently a fashionable investment strategy. It offers educational courses for its users that, in turn, will encourage them to use its services as a broker. It is highly rated on

the google app store with a 4.8-star rating and downloaded 100 thousand times (play.google.com, 2021).

IG Academy is one of many applications that offer this style of education in day trading. It does not teach personal finance, and its sole strategy is to encourage its users to put money on extremely 'risky' investments on its platform. Therefore, the primary education that is taking place on the app gives its users confidence in trying this investment strategy, which will generate money should they choose their platform for trading. Over 74% of retail investors lose money when trading CFDs, forex, and spread betting, which this investment style entails (Independent Investor, 2020). Meaning that the target audience for this type of application differs from the one this project aims for.

This literature review believes that this style of application and its rationale is reprehensible and immoral. This project's approach will ensure its users learn about finance to help them, rather than a high-risk, high reward strategy that IG academy teaches. As is written at the bottom of their site, "Spread bets and CFDs are complex instruments and come with a high risk of losing money rapidly due to leverage…" (IG, 2021). Although some of its users may benefit from learning these strategies, the target audience that this project is targeting would not be encouraged to be more responsible. Most would lose and therefore put the target audience of this project into potentially further financial difficulty. Considering the number of applications and websites similar to IG Academy, the market is oversaturated. Therefore, the business model of IG Academy should be avoided.

Duolingo - Mobile Application

Duolingo is a mobile application that has been incredibly successful, with over 100 million downloads on the Android App store (Google Play Store, 2021). The project has no desire to enter nor compete in the self-taught language space that Duolingo inhabits. However, the user experience of this app is something worth noting. It is bright, fun and friendly with soft edges, bright colours and friendly text font that welcomes the user. The designs are something that this project will take inspiration from and build an alternative financial application.

The other factor of Duolingo that will allow influencing the project is how it organises its learning. Small subject matters are grouped to make learning easier and bite-size, as indicated in Figure 5, a choice not to deter the customer. It avoids providing all the information at once and allows it to be more bite-size, as this project plans to replicate. This benefits the customer due to the accomplishment of completing one section, which avoids long study sessions that could become unbearable after several study sessions. This measured approach will be adopted. The application also provides many different methods to answer questions, such as a drag and drop for words and sentences and a way to answer by inputting text.

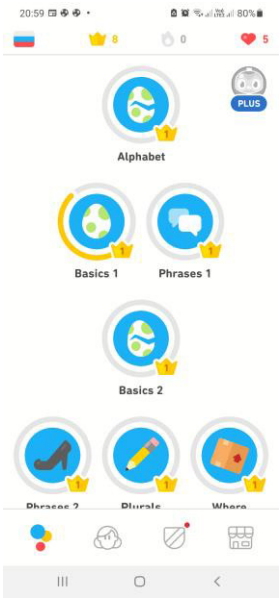


Figure 6, Duolingo mobile application. (2021)

Equally, it would be simpler for a user to focus on the specifics they are inquisitive about by using this method. For example, maybe they are more users interested in what ISAs are than how dividends work. They would feel a sense of achievement in marking a completed section that would benefit learning before trying their hand at something else.

Mint: Personal Finance Manager – Mobile Application

Mint is a mobile budgeting app. As such, it provides methods to help people control how they manage their money. This is done by the app alerting the user to: receive bill reminders, monitor expenses, track transactions and balances. The app has received excellent reviews to date, receiving 4.5 stars out of 5 from almost 200,000 reviews (play.google.com, 2021).

The app can positively impact the target audience by providing visuals that can help its users comprehend their financial situation. The application allows users to link their bank account to the application to track subscriptions and other payments. This means that it can deter users from making purchases that could be detrimental to their financial situation and prevent them from going into debt by visibly displaying this.

However, this application again does not provide the information to solve the problem this project seeks to address. However, it may help people use their money more effectively by providing tools to allow its users to save more effectively. However, it does not provide tools to help users understand how to improve their financial situation. For example, it does not deliver information about how pensions work or how to save for one. Thus, Mint should not be considered a rival. Although it offers some tools that can improve a user's financial situation, it does not provide the information they may need to become financially literate.

Competitors Conclusion

As is now clear, many solutions currently exist to solve the problem section's issues. However, none of them does this job well. These solutions do not offer accurate nor relevant advice to help customers discover how to improve their financial literacy. Instead, applications like IG Academy are predatory applications promising education to enable users to effectively 'get rich quick' despite the meagre degree of success. This review has also found that there is not currently exist a product that could offer genuine educational products on a mobile application.

Target Audience

The project will now demonstrate its research conducted on the target audience of the UK following the reflection on the current marketplace. As has been shown, a mobile application is a realistic and original solution. This segment will provide stronger reasonings of why a mobile application is a beneficial product for the problem.

In 2020, 84% of UK adults owned a smartphone, a massive portion of the population. The primary target audience, those aged between 16-54, average a 96% smartphone ownership (Cadman, 2019). That means approximately 64 million people own a smartphone. Considering this, and the statistics illuminated in the Problem segment that indicates that a third of UK adults are financially illiterate, we can determine that about 20 million people could be interested in our product, which gives a vast target audience.

Of UK parents, 83% believe it is their responsibility to educate their children about financial literacy, yet 1 in 6 of those say they are unconfident in doing so. Another illuminating statistic is that in 2017, 64% of those who sought financial advice were under 40 (Creditangel.co.uk, 2018). As this project demonstrates, an application could prove key in resolving financial illiteracy.

If this product successfully implemented a mobile app that could stand out as a unique application on the app store, it would have an enormous market section to itself in the immediate term. If developed with App Store Optimization in mind, keyword relevance, retention of installs, positive reviews and targeted UK app users (Blair, 2016), among other factors, could ensure the project's success.

It would seem clear then that an application that can utilise both the large number of consumers who own smartphones and those seeking financial information would make a mobile phone application suitable to solve the financial literacy problem in the UK. As the next segment will explore, the profitability of such an application could be high.

Market Value

After identifying a market and its potential for customers, this segment must now discuss the realistic financial capacity for such a market. It must be made clear that potential profitability has had to be an estimated metric. Not due to poor research, but what will soon become apparent, that the 'financial literacy mobile application market' does not yet exist. The estimations have meant that the project will use a selection of markets to give a general view of what size this market could be to consider its profitability. Therefore, this section will consider several statistics before making an approximate figure on the market's current value.

The most relevant market data that this project has used to validate itself is that the global personal finance software market size was valued at $1 billion in 2019 and is expected to reach $1.5 Billion by 2027 (Khan, Modi and Kumar, 2020). Representing a compound annual growth rate (CAGR) is around 5.1%, which would indicate that there is room for more growth—demonstrating that the market is significant for a particular part of the global

software market. The global software market is worth about $960 billion (Businesswire.com, 2021), which would indicate that the market right now is a tiny fraction of the software. Nevertheless, considering the niche aspect of the market shows promise for new entries to compete for market share.

This application's most general UK market it would find itself in would be the UK Software market, valued at around $24 billion in 2019. Between 2015-2019 CAGR was 12.7% (Marketresearch.com, 2020). The market value calculated with the previous 12.7% CAGR should make it worth approximately 30 billion today. Such a high growth rate would indicate that creating products for the market at this current time could mean capitalising on high periods of growth for the future.

Suppose we consider the global personal finance software market at $1 billion and compare the global software market worth almost a billion. We could attribute the total size of the UK personal finance software market. Therefore, the total UK software market size at 24 billion means the UK personal finance software market is somewhere between $20-30 million. This could be considered in some industries as minor. However, considering that this product would incorporate the mobile application market, whose revenues are expected to reach $30 billion by 2025 (Martin, 2014), we can safely say that this figure would be at the lower end of this theoretical market size.

Therefore, the research indicates that the target market's actual value is unknown but is estimated in the tens of millions. However, it is safe to assume that the product could be profitable due to similar markets used to signpost the market. It can grow in time, as the figures have indicated. Hence, this product could prove to be an excellent harbinger of a new form of application on the mobile application market.

Literature Review Conclusion

This literature review has focused on the new market in which the product would enter using several different markets to estimate the application's value and potential profitability. It has also demonstrated the lack of clear competitors. The most relevant competitors do not run mobile applications, only websites. While companies that do own

mobile apps do not demonstrate that they could be considered rival products to this project's application.

The research into the target audience showed that a significant portion of the UK population owns a mobile phone and is interested in learning about financial literacy is high, which makes the mobile application a viable outlet for this solution. Finally, the research into the market values showed that, although it currently does not exist, it is still large enough to enter a product. The literature review has shown clear evidence of the viability of a product to solve the problem. The remaining chapters will discuss the steps taken to create the product.

Specification

This chapter is designed to outline the specifications of the application in order to understand how it was implemented. This chapter will demonstrate the functional and non-functional requirements of the application and their reasonings, then use cases that will demonstrate how they are intended to be used.

Functional and Non-Functional Requirements

This section will discuss the Functional and Non-Functional aspects of this project. In software engineering, a Functional Requirement is a term that describes the functions a software must perform. The function is the inputs, the behaviour and its outputs (Martin, 2019). The objective of the function can be many different things, and some examples will be discussed below.

Non-Functional Requirements are the defined constraints that affect how the system should do the functional requirements (Rome, 2020). For example, a functional requirement on a website might be that a user can navigate to a store page. While a non-functional requirement might be that they can access this by a menu or that navigating to that page should take only one second between clicking the link and loading the page.

The both sets of requirements that have been selected below are drawn from a few sources as stated in the Aims and Objectives. Such as creating an application that can solve the problem of financial literacy by providing information and quizzes to take while also giving feedback so they can improve over time. Additionally, the survey I conducted and the results of the Literature Review influenced what the functional and non-functional requirements should be.

For example, the survey showed that no single group was willing to say they were very entirely confident in any subjects. Therefore, multiple tests should be used to aid all learners across a range of subjects, even if they are more confident about some than others. The Literature Review also demonstrated that Duolingo was an essential contextual

application to use. There should be more than one way to answer questions on a subject, so therefore, I will provide two options to answer a quiz.

Functional Requirements

FR 1: The user can choose a subject to take a test on

Acceptance Criteria:

- A user can click a link that begins the process of taking a test.

FR 2: The user can choose the type of quiz they would like to take

Acceptance Criteria:

- The user can choose an option of the type of test they would like to take.

FR 3: After choosing a subject and type of quiz, the user can access an information page that is relevant

Acceptance Criteria:

- A user can learn some information presented to learn about the subject they will be questioned on.

FR 4: The user can complete a quiz on the chosen subject

Acceptance Criteria:

- The user can complete all questions and then exit the quiz

FR 5: The user's quiz score is retained in a database

Acceptance Criteria:

- The user's score is automatically saved to a database on completion of a test

FR 6: The user can view their score in a database

Acceptance Criteria:

- The user can see the completed quiz
- The user can view all recently completed quizzes

Non-Functional Requirements

NFR 1: The user is given feedback after each question to learn from their mistakes

Acceptance Criteria:

- After submitting an answer to a question, the user can understand if they got the answer correct or incorrect.
- The user can see what the correct answer was if they were incorrect.

NFR 2: The user can see a recap of their score on a results page

Acceptance Criteria:

- The user is presented with a page that indicates how many questions they were asked, how many questions they got correct and some feedback.

NFR 3: The user can access support to charities

Acceptance Criteria:

- The user can navigate to a support page successfully.
- The user can click on links to call or go on the web to contact these services.

NFR 4: The application has a logo and title

Acceptance Criteria:

- The user can see some branding of the application.

Use Cases

This section will now demonstrate the intended use cases of the application. A 'use case' describes how a user is meant to use a process or system. It does this by describing the steps in reaching the desired outcome. For example, if a user needed to send an email, the steps might describe opening a browser and opening their email provider client. Writing an email then sending it to the recipient. The following segments will discuss a use case diagram and several use cases of the mobile application.

Use Case Diagram

This segment will use a Use Case diagram for summarising the needs of the projects users and their interactions with the system. A use case diagram does not go into much detail about how the interactions will take place. However, it does depict a high-level overview of how the projects use cases interact with one another (Lucidchart, 2019). The diagram is drawn with the Unified Modelling Language (UML), a visual language that aids software developers to visualise and then construct new systems, similar to a blueprint (Gliffy, 2020).

Within the Take a Subject function, four functions are included that link with each other in sequence:

1. Choose a Quiz enables a Quiz to be picked and taken.
2. Learn about Subject information is displayed for the user to study.
3. Complete Quiz, the user can complete a quiz.
4. Save to Database after completing a quiz. The data is saved to a database.

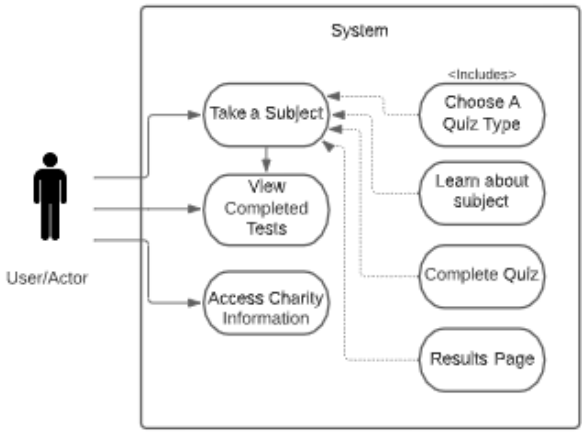


Figure 7, Use Case Diagram

Use Cases

This segment corresponds with the last segment to explain the use case diagram in greater detail. It will go over the steps that a user would take to work the application functions mentioned in the Use Case Diagram in Figure 7.

Use Case No.: 1

Description: Users can take a subject to complete a test.

Type: Must have

Goal: A user wishes to take a quiz. They will need to open the application, choose a subject from the home screen, choose a type of quiz to take, then read about the subject. They will then take a quiz in their selected test style and, once completed, will view their results.

Pre-Conditions: The application is installed on the user's device.

Steps:

1. The user opens the application.
2. They see the splash screen, which closes, and the application displays a home screen.
3. They choose a subject from the list of subjects on the home page.
4. Choice page is displayed showing Multiple Choice and Fill the Blank Options.
5. Clicking either of these will take them to the information page, which displays the relevant subject information.

6. Clicking a prompt to ask if they want to start the test, the Multiple Choice or Fill the Blank Quiz begins.
7. The user answers questions until completed. On the final question, a 'Result' button appears.
8. User clicks this button which will then display the results page showing their scores from the test.

Related Use Cases: UC 2

Diagram of steps:

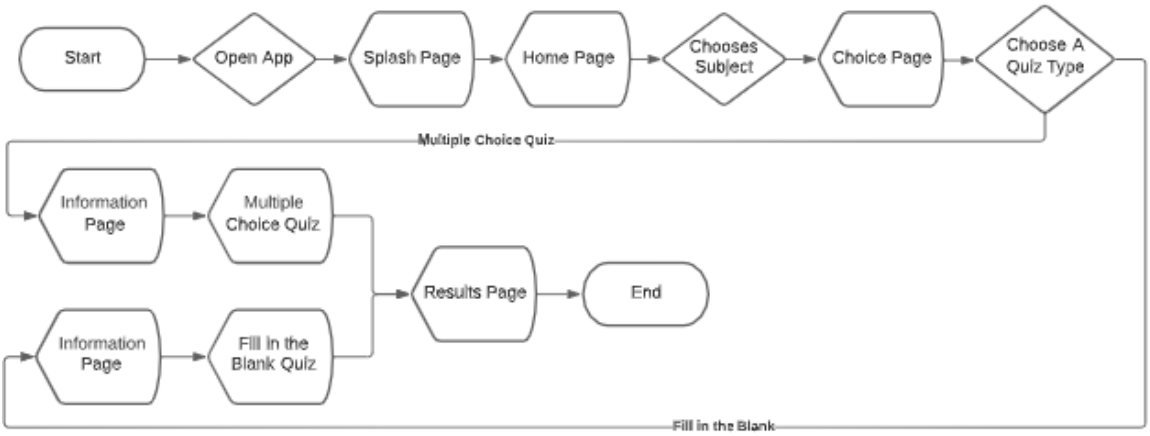


Figure 8, UC 1 steps diagram

<u>Use Case No.: 2</u>

Description: User wishes to view their past results of quizzes

Type: Must have

Goal: User can view their past results in order to track their progress in the

Pre-Conditions: The user has opened the application.

Steps:

1. User clicks on the bottom navigation menu option called 'progress'.
2. The progress page is displayed

3a. If the user has completed quizzes in the past, the user is then able to see a list of previous attempts at the tests with relevant information

3b. If the user has not completed quizzes in the past, the user can view a message that states no quizzes have been completed.

Related Use Cases: UC 1

Diagram of steps:

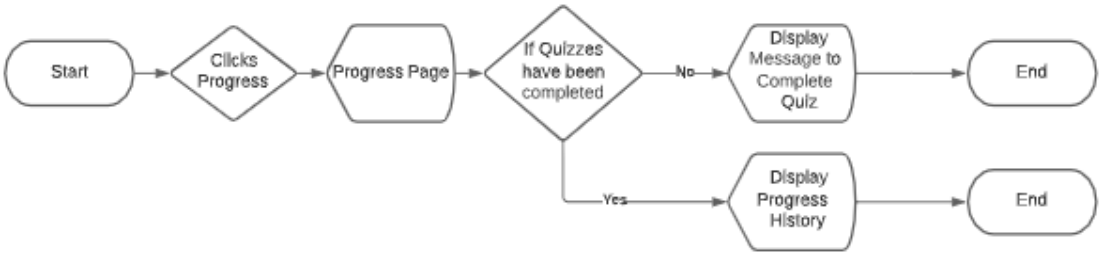


Figure 9, UC 2 steps display

<u>Use Case No.: 3</u>

Description: The user wishes to call a charity in order to get support.

Type: Optional

Goal: User can access support from charities via phone call.

Pre-Conditions: The user has loaded the application.

Steps:

1. User clicks on the bottom navigation bar option labelled ‘support’.
2. After being clicked, the support page displays.
3. The user can see various types of charities, the subject of what they do and what they are called.

4a. The user clicks on the phone number or phone icon on the page of a charity.

5a. Application closes and opens the ‘phone’ application, with the charities number displayed ready to be dialled.

4b. The user clicks the website or website icon.

5b. Application closes and opens the website in a browser application.

Related Use Cases: N/A

Diagram of steps:

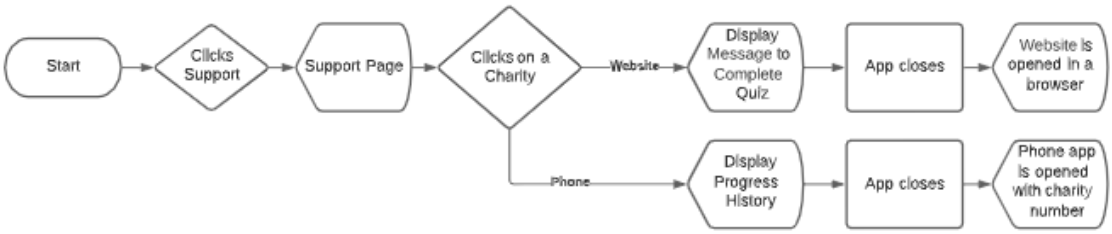


Figure 10, UC 3 steps display

Approach

This next chapter will review the approach that was taken during the implementation of this project. The chapter will discuss the decisions made during the project concerning which particular framework organised the program's production, the chosen programming language used, the type of assessment used, and the quiz styles used.

Frameworks and Programming Languages

One of the most crucial factors that would determine the success of this project was the successful choice of a Software Development Framework and the choice of programming language. The first part of this segment will discuss the Framework decision. The second segment will likewise discuss the choice of programming language.

Framework: Agile or Waterfall

When the chosen solution to the problem was reached, decisions were needed to organise workflows to develop the program efficiently. The research was conducted to examine the various styles of frameworks until just two software development frameworks remained, Waterfall and Agile.

Agile had many benefits, and it was a framework that I had already been encouraged to use before. During studies at the University, the Software Engineering module and its teachers had encouraged its use. Over 70% of companies use Agile approaches in industry, and 98% of companies report that using Agile has benefitted them (Djurovic, 2020). There are many benefits to Agile that applied to my needs:

The Quality Product	Before Agile was used, most testing would occur before launch and errors would be found and amended during this phase, potentially delaying the product's rollout. However, with Agile, "testing is integrated throughout every stage of development to ensure a quality end product" (App Partner, 2017). Because of continuous testing, Agile enables adjustments

	to be made more often and can catch issues and bugs before they manifest.
Time to Market	As Megan Cook, Head of Product – Software Teams at Atlassian, has said: "…a product is built in a series of iterations called sprints that break down big, complex projects into bite-sized pieces…" (Rehkopf, 2019). The final output of Agile development is a minimum viable product.
Flexibility	Additionally, due to the use of sprints, work is more flexible. It is flexible because the small workloads enable quick reviews of the progress of the project. Should the project change somehow, the framework enables quick pivots to new priorities in shorter periods.

The alternative framework identified was the Waterfall framework. The Waterfall methodology is a successive development structure. It acts like a waterfall through what are referred to as phases of a project, Scope and plan project, Gather and document requirements, Design application, Develop application and perform unit tests, Conduct system testing, Perform user acceptance training, Fix application as appropriate, Deploy the application. Beginning with the design of the application, each individual phase is designed to be finished before the next phase is started (Sherman, 2014, pp.449–492). The Waterfall methodology did also have several advantages that could have been applicable for the project.

Timescales	Due to the nature of the phased development cycle, Waterfall enforces discipline. Each phase has a clear and defined starting and endpoint, making monitoring progress easy and avoiding “slippage” (Kienitz, 2017).
Testing is made easy	Where relevant, testing should occur at the end of each phase of the project. The project will not prematurely advance before it is ready and allows bugs and errors to be removed only after development (Martin, 2019).
Clear Outcome	Before the software development starts, the objectives are laid out in detail, which clarifies the needs and outcome. Providing clarity to the project enables work that is focussed, with the goals in mind at all times (Cavet,

2021). Because objectives are clear, a final product is expected when development finishes.

Therefore, before proceeding, a decision needed to be made. Overall, after considering the research, this project chose to follow the Agile framework. Partially because of existing familiarity with the methodologies. Having used Agile in previous collaborations, I felt more confident that using a methodology that I was experienced with instead of a framework I was not could hinder my progress from completing the project on time. The previous experience with Agile was arguably the most crucial factor, as it meant that fewer new skills would be needed to be developed. At the same time, a deadline loomed for the development. Additionally, an Agile framework was better suited to this project than a Waterfall methodology.

In terms of fulfilling the objectives, because Waterfall uses a Linear process, a well-defined approach is needed before the project begins. At the same time, Agile can have undefined goals while tasks are completed, meaning that sprints could be used to define objectives as they were needed (Glasscubes, 2019). The added flexibility was something that the project would benefit from, especially considering the need to revisit and change various parts of the project, if necessary, since I was building a mobile application for the first time. Furthermore, the need Waterfall methodology required a completed product. In contrast, Agile requires a minimal viable product. The Agile methodologies lend themselves to the objectives of this project more than the Waterfall as due to time constraints would make a complete development impossible to achieve.

As noted, testing in both of these examples is done in very different ways. Waterfall has fewer tests at the end of each phase, while Agile has many more tests much more regularly. Both of these have their advantages. However, considering that this project was the development of a mobile application that I had never done before, testing regularly and often was more important than testing at the end of a phase. This is because due to time constraints, it would be crucial that delays should be avoided as much as possible. If delays did occur with the Waterfall methodology instead of Agile, the consequences could have been the failure to release a product rather than a product with slightly less functionality. It

decided that using the waterfall methodology would be detrimental to the overall production of the application.

To conclude, the Agile framework leant itself more successfully to dealing with the project's time constraints, the desired outcome of the project and allowed greater flexibility during its development. Learning skills and understandings from following a new methodology would have been detrimental to the successful implementation of a mobile application.

Mobile development: Dart, Java, Kotlin or Swift

Dart

Dart is a programming language that was first launched in 2011 and is used to build mobile apps. Until 2018, Dart was a reasonably small language, but with the launch of Flutter – a software development kit (SDK) – the content was made much greater. Flutter gave Dart access to an extensive UI library with a set of development tools that support developers to create, among other things, mobile apps (Björklund, 2021).

Researching Dart gave excellent reasoning for learning the language. Developers say it is easy to learn, it is fast, reliable, efficient, and backed by a tech giant (Fuad, 2019). One of the critical reasons Dart was considered was its ability to create cross-platform applications. The application code could have been run over both Android and iOS devices. This would be enormously beneficial to creating a minimum viable product and would have contributed to solving this project's problem.

Notable statistics of Dart is that it is one of the most loved languages. It came 7th, with just under 64% of respondents claiming to love it to hate it. However, other statistics were troublesome. Its community was smaller than some of the other languages being considered, and it was used by only 6% of Stack Overflow users. It also was one of the lowest-paid languages to learn (Stack Overflow, 2021).

Most mobile applications are designed to be small as mobile storage space is smaller compared to a computer. However, a minimum app size of 4MB for Flutter applications compared to Java's 539KB or Kotlin's 550KB (AltexSoft, 2021) makes the mobile apps many times larger to download. In turn, this could cause users to delete the app in favour of

downloading or keeping other applications when necessary for storage. Customer retention would be a vital objective of the project and a measure of its success. A large application, therefore, should be avoided.

Java

Java is another programming language, first developed by Sun MicroSystems beginning in 1991 and released in 1995, designed to be used for the internet. Java allowed greater interactivity and displayed multimedia, which showed that it was particularly well suited for the Web (Encyclopaedia Britannica, 2021).

Stack Overflow's 2021 annual report placed Java as the 5th most popular programming language (Stack Overflow, 2021). As a result, there are a great many online resources and books available to learn. Should developers get stuck, it is easier to find solutions than newer languages like Kotlin (Wintrich, 2020). Java is used on around 7 billion devices by 9 million developers with the Java Virtual Machine (JVM). The JVM turns Java code into bytecode for the computer to interpret, meaning Java can be used on almost any device that can install the JVM. Since the release of Android in 2008, Java has been Androids primary programming language (GeeksforGeeks, 2019. Jones, 2021). Java is a reasonably sensible language to develop in considering its establishment.

There are, however, many reasons to not develop in Java. One such reason is that among developers, Java is one of the least enjoyed, with just over 47% enjoying using the language (Stack Overflow, 2021). The lowest in this comparison. Java is declining in popularity in the broader development world, with languages like Python overtaking it (Barron, 2021). However, in the more relevant Android development world, Kotlin is seen as the successor of mobile development, "Kotlin vs Java…makes Kotlin the future language for enterprise applications and Mobile" (Singh Gill, 2020). As of Google I/O 2019, it was announced as increasingly Kotlin-first (Android Developers, 2021), indicating that the future role of Java was likely to diminish over time.

Kotlin

Kotlin is a programming language developed in 2010 by JetBrains. JetBrains uses Kotlin in many of its products, including its flagship IntelliJ IDEA. Kotlin was designed for the Java Virtual Machine (JVM) to run Android programs (Heller, 2020).

Kotlin, like Dart, has been significantly supported by Google, developer.android.com/kotlin is dedicated to the language, providing guides and learning materials for developers. As it uses the JVM, Kotlin can be used on any device. Kotlin can be used interoperability with Java. It can be used in programs that feature both languages (Shneyveys, 2021). Kotlin is easy to learn, important as I was approaching mobile development for the first time. It has excellent performance compared to Java and is safer than Java. Kotlin prevents common programming mistakes by design (Beladiya, 2019). The Stack Overflow Annual Developer Survey 2021 rated Kotlin as 14th most beloved programming language at almost 62% (Stack Overflow, 2021).

In terms of community, Kotlin is growing, but compared to Java, it is dwarfed. It had the 15th largest community on Stack Overflow at approximately 8% (Stack Overflow, 2021), which means that asking questions for information when problems inevitably arose would be less likely to be answered than other languages.

Swift

Swift is a programming language developed by Apple and released in 2014. Swift replaced Objective-C, a slower object-orientated language developed in the 1980s. Swift, by contrast, is over 2.7 times faster than Objective-C (Apple, 2021).

Swift is considered easier to learn than other programming languages such as Javascript, Java, and Python. It’s designed in such a way that development is limited and structured to stay on track. Therefore, allowing its programmers to develop products at a faster pace (David, 2021). It is a language which is also loved, sitting just under Dart in 8th in most loved/hated at over 63%. It is also a high paying language to learn, sitting highest amongst the other languages being considered.

Swift has a few issues. First, Swift needs to be written on a computer that runs Mac, the Apple Operating System, which was unavailable to this project. A solution could have been to run a virtual machine as a mac device and program this way. However, this would have

been a great deal of time to set up. Second, Swift is not cross-compatible with Android, which meant testing the system would be more difficult as this project did not have access to an iOS device either. Third, problem with the Swift programming language was its reach. Apple may have a significant overall portion as a company in the mobile phone market. It still has much fewer users than Android. Androids share of mobile phone operating systems is approximately 73% (StatCounter, 2021) worldwide. In the UK, the gap is smaller but still significant. Currently, there are 28.9 million Britons with Android devices, compared to iOS with 19.7, a gap of 9.2 million. By 2023, this gap is predicted to increase to 10.1 million (eMarketer, 2021). Swift is also not a cross-compatible language with Android, meaning that it could not be used to create Android products.

Conclusion

As is evident, none of the programs that have been examined stood out as being the clear choice to go forward within the project. However, Swift was dropped due to its reach and the ease with which it could be developed. Java was omitted because the future of Android is moving away from it. Languages like Kotlin and Dart are replacing it and have the support of Google. Finally, Dart and Kotlin, this was probably the most challenging choice that was made. Both languages are the future of Android. Arguably, however, Dart is slightly more favoured as Google itself has built it, unlike Kotlin. However, Kotlin and Dart are similarly loved by developers, with Dart being just ahead. I chose to use Kotlin, however. Kotlin has a smaller minimum app size, meaning that fewer resources are needed to install the application, making it unintrusive to a user's phone. Kotlin also has a larger community than Dart. Kotlin is regarded as the future of Android, but its present, "Kotlin is not only the future of Android app development but its present" (Scully, 2020). Therefore, Kotlin was chosen over Dart.

Assessment and Quiz types

This segment will discuss the decisions made to choose the type of assessment and the type of quizzes the mobile application would use. The assessment type was vital because it would help provide the correct type of learning that was needed and the feedback that would go with it. While the quizzes selected were critical because they needed to be engaging and the

correct type of quiz for the type of application being produced. The first section will discuss the assessment type, and the second part will discuss the Quiz types.

Formative or Summative Assessment

In order to produce this mobile application, decisions needed to be made on the type of assessment style it would use. There are two main types of assessment, formative and summative. This section will discuss the decision on the summative assessment style chosen.

The objective of formative assessments is to assess and monitor a student's learning, feedback that can be used by an instructor to improve their own teaching but also used by students to improve their learning on an ongoing basis. Therefore, formative assessments can help a student recognise their strengths and weaknesses on a topic and help the assessors identify their faults in the curriculum as it is being taught (Carnegie Mellon University, 2019).

The objective of summative assessments is to review students' learnings at the end of a learning period. Such as GCSE assessments. The amount of time between assessments can vary. Tests could occur at the end of topics, at the end of the month or end of a school year. The result of these tests is a summary of what has been attained. This can provide information on tracking progress (NFER, 2018).

The two methods have their benefits and disbenefits. Some studies have shown that formative assessments are one of the best motivational factors in students' learning, making them more aware of what they have learned. They need to study to improve their understanding (Weurlander et al., 2012). In terms of a learner's preferences, some studies have shown that sometimes learners will prefer formative or summative depending on the contextual situation of what they are learning (Iannone and Simpson, 2016). In contrast, others argue that there is no significant difference between formative and summative assessments. Stating that formative is a summative assessment with some feedback that a student can use (Taras, 2005).

To conclude, the summative assessments were chosen for a few reasons. As this product is a minimum viable product, the complexity of tracking progress for all users and regularly updating the product, albeit a good idea, would be impractical to this project, especially considering the time concerns of the project. Formative assessments would be something that would be incredibly beneficial to the project should it be launched as a complete product. However, the nature of this project lends itself better to summative assessment because an assessment takes place each time a quiz is completed. This demonstrates how well a user understands the project to themselves. A formative assessment would require a large team to track progress to tailor resources that this project could not realistically achieve.

Quiz Types

The application had to consider the best methods that a user could use to learn the resources presented to them. There are many types of ways that a quiz could have worked. However, ultimately, two choices types of quizzes were made. The section explores the considerations made when choosing the types of quizzes and why others were omitted.

It should be evident that given this project is based upon the work of a mobile application, the primary function of which is to have quizzes to check the users understanding. Interactivity is a necessity. The program must be engaging enough for a user to enjoy and wish to try again. Many types of quizzes could have been used. Below are eight possible examples that were considered:

Multiple Choice	Questions pop up to the user, and they have several options to select before moving on to the next question.
Spin the Wheel	A random topic and question are selected to test the user on a variety of subjects.
Who Wants to be a Millionaire	Based on the famous TV show, the user is presented with a series of questions from a low value, trying to reach the

	final question at a high value. Questions are getting more challenging each time.
Assessment	A literal assessment whereby a user fills in a document of questions.
Pairing-Up	Similar to a connect the answer to the question. Questions on one side with answers on the other, the user must connect these correctly.
Fill in the Blanks	A user is presented with either the beginning, end or both parts of a sentence. The user must fill the blank with the correct answer.
Noughts and Crosses	The user plays a game of Noughts and Crosses, also known as Tic Tac Toe. A user selects a square to play their nought or cross. They are given a question to answer. If successful, they play the move. If not, they lose a turn. The winner needs three in a row.
Crossword	A user is presented with a crossword puzzle with blank spaces to fill out. The missing words are based on related questions.

There were pros and cons to each of these answers. The initial assessment on these options was the engagement factor, would a user be willing to learn with the particular quiz type. This immediately negated the assessment as an option. It was too similar to school and was not fun. Additionally, although some people enjoy crosswords, to some people, they can be intimidating. For this reason, this option was dropped.

The following options that were dropped were the Spin the Wheel and Who Wants to Be a Millionaire. The Spin the Wheel option was not deemed an appropriate learning technique. The difficulty is that the randomness would mean that users could not focus on a particular subject matter. They would instead be perhaps frustrated by the random nature of their learning. The Who Wants to Be a Millionaire option was dropped for one particular reason, among others. Realistically, licensing issues would make the project harder and expensive to launch. Launching a similarly named version such as a 'Billion-Pound Quiz' or perhaps, 'Quiz for Big Bucks' would feel cheap and tacky as the user would probably know what show was being alluded to. Additionally, the idea of encouraging people to learn that taking the risk for monetary reward seemed extremely counter-intuitive to the objective of this application.

Despite being one of the best options, the Noughts and Crosses quiz was dropped because the game is normally intended for two players. Therefore, the options were to create a system to have a computer play against a player or have two players play against each other. Firstly, the game vs the computer, the computer sometimes needs to get questions wrong or right. The difficulty would be that algorithms would need to be developed to allow the computer to win the game. Therefore, further complexities about game balancing would need to be implemented. Again, time constraints were considered, which meant this option was dropped. Secondly, the user vs user option seemed like a very fitting way for the project to help gain traction, encourage users to play. Another person who may not have the application would spread knowledge of the app. However, the data indicates that this may not be appropriate. According to one report, discussing money is not just a taboo subject. It is more uncomfortable to discuss than sex, religion or politics (Lloydsbankinggroup.com, 2019). Therefore, it would likely mean that such a feature would go unused as people would not wish to seem 'stupid' or discuss their knowledge with the other player.

Pair-Up was another excellent idea. Getting a user to rearrange or draw lines from the questions to the correct answers could be considered engaging instead of simply selecting an answer. It was dropped as an option, however, for a few reasons. This type of quiz would have meant that more questions with definitively different answers would need to be written. Because the questions and answers could be four at a time, if questions were drawn randomly, assurances would need to be made to ensure the user did not have

situations where similar questions or answers wound up on the same question slide. If this was not prevented, a user would become frustrated with the product. Most likely uninstalling the application due to such an error.

Leaving two options, Multiple Choice and Fill in the Blank. These were considered the best options for the application. Multiple Choice Question was chosen as it is arguably one of the best-known ways a quiz can be run. It is easy to keep engaged as answers and feedback can be immediate. There is also little time to dwell on a mistake as the following question can be quickly brought on. It would be less discouraging for the userbase. Fill in the Blank was then chosen because it differed significantly from the multiple-choice option. While most of the other quizzes involved choosing an option, Fill in the Blank asked for direct input from the user to type the correct answer, making it both interactive and differentiated enough to give two clear types of quizzes that it could offer.

Implementation

This chapter will discuss the implementation of the project. It will consider what was learned in the previous chapters of the dissertation and how they were applied. It shall demonstrate the development of the functional and non-functional requirements and its handling of the use cases. It will be discussed by using code examples and designs.

Please find a bullet point explanation of how to access all relevant files if needed from the application:

<u>To access the modified 'AndroidManifest':</u>

- Open the 'app' folder.
- Open the 'manifest' folder.
- Open file entitled 'AndroidManifest.xml'.

<u>To access the design aspects of the files:</u>

- Open the 'app' folder.
- Open 'res', not the "res (generated)" folder.
    - Fonts:
        - Open 'font', 4 files are in this folder.
    - Colours:
        - Open 'values', open 'colours.xml'.
    - Layout:
        - Open 'layout', many files are found in this folder.
    - Images/symbols/backgrounds:
        - Open 'drawable', many files are found in this folder.
    - Navigation:
        - Files are fond in 'menu' and 'navigation' folder.

<u>To access all other files:</u>

- Open the 'app' folder.
- Open 'java', not the 'java (generated)' folder.
- Open 'com' not the 'com (androidTest)' or 'com (test)' folders.
- Open 'example' folder, then 'finacialapp' folder.
- All main files that should be examined are found in the folders displayed in this folder:
    - 'adapter' contains 'ResultsAdapter' file used for the database.
    - 'db' contains the 'QuizDao' and 'QuizDb' for the database.
    - 'fragments' contains 'main_page_fragments' which include the module, progress and support fragments.

- ‘models’ contains the data classes used to create the quiz such as questions and inputting to database.
- ‘questions’ contains two folders full of questions for the user.
- ‘view’ contains the most important activities of the application.
- ‘Info’ is the source of the information pages.

Code Examples

The following segment will consist of explaining the way in which the application works. It will take the key parts of code and explain how they work, rather than the entire code as this would be far too long to include. The code selected will be relevant to the use cases that were discussed in the specification chapter.

Below in Figure 11 shows the overall flow and look of the application. This can be used to help understand the code being discussed in the following segments.

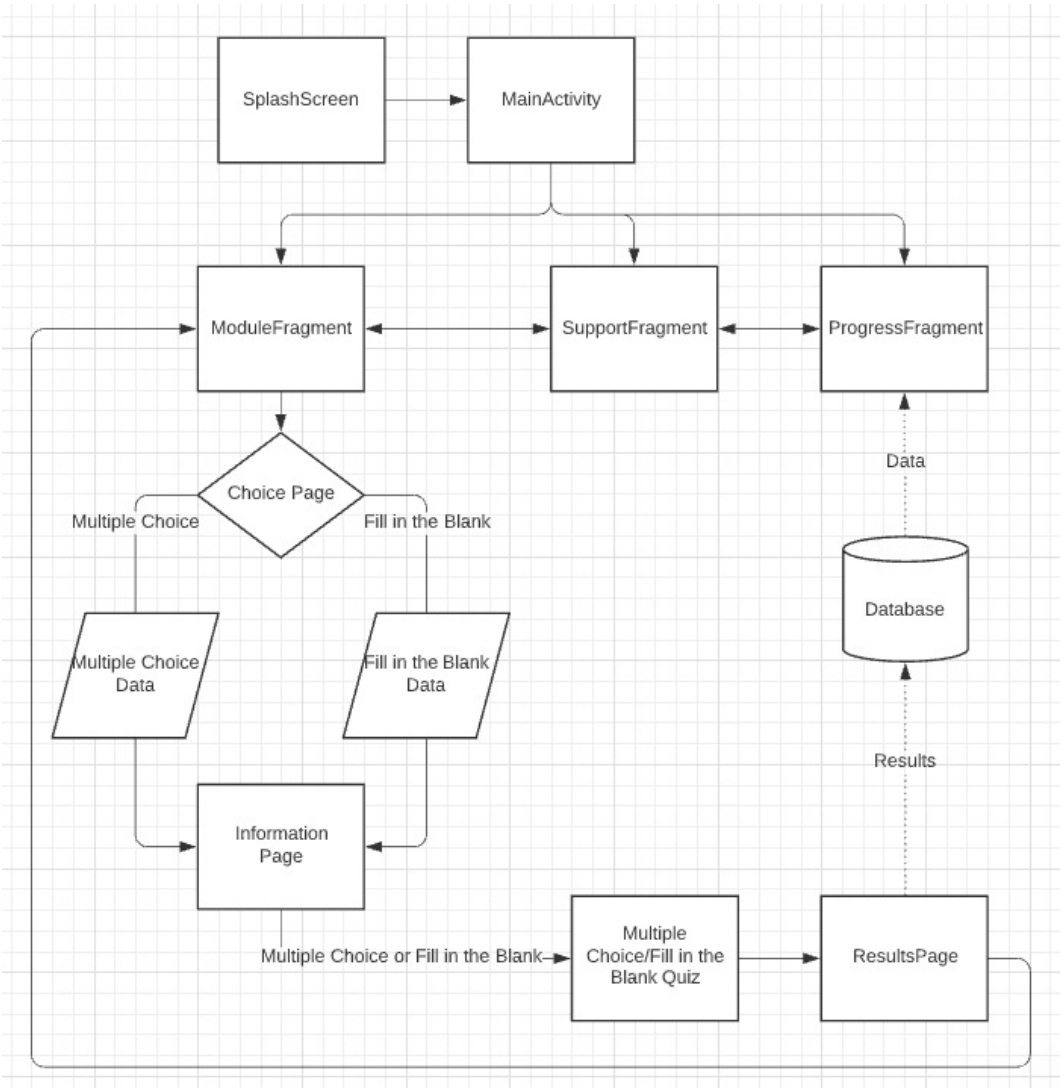


Figure 11, Diagram of flow of the application.

Taking a Quiz

This part is based on the Use Case 1 discussions, which is the application's main flow and purpose. This Use Case shows the product can open. A subject is chosen to be taken from a list of subjects. The user can choose a type of quiz they wish to answer, an information page is displayed on that subject, they can then answer the relevant questions. Finally, a results page is displayed, and their score is saved to a database.

1. **AndroidManifest and SplashScreen files**

Section 1 discusses the 'SplashScreen' and alterations to the 'Android Manifest'. This code enables a window other than the 'MainActivity' to be opened when the application is launched. It is intended to be visible for a few seconds before disappearing.

To begin, the user needs to open the application. In order to display a splash page, an edit must be made to the 'AndroidManifest' file, as depicted in Figure 12 shows. Within the SplashScreen Activity lines 24 to 30, I have adjusted the code always to start the 'SplashScreen' first when launched, as indicated by lines 26 and 28.

The key code for the splash screen is displayed in Figure 13. This clearly shows a portion of code that enables the delay of a runnable thread. As can be seen, the code on lines 18-20 is inside of the brackets for the code on lines 17 and 21. The latter clearly shows a one thousand-five-hundred millisecond, or one and a half-second delay to being run. Inside the 'postDelayed' on line 18 is an attribute called intent, assigned the class 'Intent'. An Intent is an abstract description of a method that needs to be performed. In this example, the code will start a new activity for a file called 'MainActivity'. Line 19 executes the intent by calling the 'startActivity' method with the 'intent' variable. It is then closed on line 20 with 'finish()'. The code is then executed after the delay has been completed and launches the 'MainActivity'.

```
24        <activity android:name=".view.SplashScreen">
25            <intent-filter>
26                <action android:name="android.intent.action.MAIN" />
27
28                <category android:name="android.intent.category.LAUNCHER" />
29            </intent-filter>
30        </activity>
```

Figure 12, snippet of 'AndroidManifest' file

```
17        Handler(Looper.getMainLooper()).postDelayed({
18            val intent = Intent( packageContext: this, MainActivity::class.java)
19            startActivity(intent)
20            finish()
21        }, delayMillis: 1500)
```

Figure 13, snippet of 'SplashScreen' file

2. MainActivity and ModuleFragment files

Section 2 explains how the 'MainActivity' and 'ModuleFragment' interact with one another. The code shows a portion of how the bottom navigation works and how a quiz can be taken.

The way that 'MainActivity' class is organised is to use three fragments to use the same activity at the same time. These are the Module, Progress and Support Fragments. Therefore, the 'MainActivity' is used as a starting point for the rest of the application and merely provides a menu system for the user to navigate along the bottom of each of the fragments. In Figure 14. It shows the items on the menu: Home, Progress and Support.

Figure 15 shows the program turning some items displayed in the design of the module as becoming clickable. Line 29 sets a variable called benefits equal to the layout page with the id of 'tvBenefits'. It enables this variable to be linked directly to the displayed item. The variable is then turned into a button on line 30 with the 'setOnClickListener'. It then uses passes a variable that is passed to a function in Figure 16.

Figure 16 shows a private function called 'sendIntent', which is used to launch the following activity to launch the 'ChoicePage' as shown in line 55, designed for a fragment rather than an activity, 'putExtra' items manipulates the 'intent' variable. It simply passes additional information that can be attained and used when the next activity is started. On line 56, a key called "information" is passed to be intent, which is checked by a when loop. As Figure 16 shows, the benefits button passes a 0, so when the function 'sendIntent' is called, the 0 is passed in for the intent.

```xml
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/moduleFragment"
        android:title="Home"
        android:icon="@drawable/ic_baseline_home_24"
        />
    <item
        android:id="@+id/progressFragment"
        android:title="Progress"
        android:icon="@drawable/ic_baseline_done_24"
        />
    <item
        android:id="@+id/supportFragment"
        android:title="Support"
        android:icon="@drawable/ic_help"
        />
</menu>
```

Figure 14, 'bottom_nav' code snippet used for 'MainActivity'

```
        // For navigating to quizzes
        val benefits = binding.tvBenefits
        benefits.setOnClickListener { sendIntent( passedVariable: 0) }

        val everydayMoney = binding.tvEverydayMoney
        everydayMoney.setOnClickListener { sendIntent( passedVariable: 1) }

        val familyCare = binding.tvFamilyAndCare
        familyCare.setOnClickListener { sendIntent( passedVariable: 2) }
```

Figure 15, Snippet of 'ModuleFragment'

```
    private fun sendIntent(passedVariable: Int) {
        val intent = Intent(requireContext(), ChoicePage::class.java)
        intent.putExtra( name: "information", passedVariable)
        when (passedVariable) {
            0 -> {
                intent.putExtra( name: "subjectName",  value: "Benefits")
            }
            1 -> {
                intent.putExtra( name: "subjectName",  value: "Everyday Money")
            }
            2 -> {
                intent.putExtra( name: "subjectName",  value: "Family & Care")
            }
            3 -> {
                intent.putExtra( name: "subjectName",  value: "Homes")
            }
```

Figure 16, Snippet of 'ModuleFragment' code

3. ChoicePage file

Section 3 is a 'ChoicePage' simple page that takes the information passed from the 'ModuleFragment' and allows a user to choose a type of quiz to take that will be displayed in subsequent pages the information page, then passes this information on.

Next, the 'ChoicePage' is opened. As Figure 17 shows, on lines 18-20, the code extracts the extra variables passed to it by the intent in the module fragment. It assigns some of these extras to new variables. From here, lines 22-24 and 33-34 create two variables to act as buttons to correspond to the display, which is done by the same methods described prior.

Within both of the 'setOnClickListener', an Intent is used to begin the following activity for the 'InformationPage'. Then the intent is passed the value and subject name values previously extracted. This page features two buttons that can route the user to two different pages. Clicking either button corresponds with either button created from the Listeners. If they choose the Multiple Choice option, the code in lines 25-30 is executed, and intent sent. Otherwise, lines 33-40 are executed when they choose the Fill in the Blank options. The difference is lines 28 and 38, which send different Booleans in the intents.

```
17
18          val extras = intent.extras
19          val value = extras?.getInt( key: "information")
20          val subjectName = extras?.getString( key: "subjectName")
21
22          val optionMultipleChoice = binding.tvMultipleChoice
23
24          optionMultipleChoice.setOnClickListener{  it: View!
25              val intent = Intent( packageContext: this, InformationPage::class.java)
26              intent.putExtra( name: "information", value)
27              intent.putExtra( name: "subjectName", subjectName)
28              intent.putExtra( name: "choice",  value: true)
29              startActivity(intent)
30              finish()
31          }
32
33          val optionFillInTheBlank = binding.tvFillInTheBlank
34          optionFillInTheBlank.setOnClickListener{  it: View!
35              val intent = Intent( packageContext: this, InformationPage::class.java)
36              intent.putExtra( name: "information", value)
37              intent.putExtra( name: "subjectName", subjectName)
38              intent.putExtra( name: "choice",  value: false)
39              startActivity(intent)
40              finish()
```

Figure 17, snippet of 'ChoicePage' code

4. InformationPage file

Section 4 is an 'InformationPage' which allows a user to read information about their chosen subject based on the information passed to it. The page then passes the information on again to the quiz pages.

As Figure 19 shows, the information passed to the file by the intent is extracted. On lines 30 and 31, the extracted variables are used to grab information from a data class named Info, which has all the questions that are asked stored inside it, Info uses a function called 'getInformation' that grabs all the information that will be displayed on the Information Page dependent on the variables passed in the intent. Line 31 gives a variable the type of 'InformationSheet' that is the method by which the information sheet is designed to be displayed. The relevant information that the user wants from the passed 'value' variable. This information is grabbed by line 31 in 'finder'. Lines 33-35 then allow this information to be displayed on the screen.

The program then checks the value of bool. If it is true, then the user wishes to try a multiple-choice question. The intent then sends them to that question page. Otherwise, they wish to try the Fill in the Blank Question. In either case, the button to take them to the questions page seems the same but is determined by the prior bool. The intent is fired when the button is clicked.

```
26        if (extras != null) {
27            val value = extras.getInt( key: "information")
28            val subjectName = extras.getString( key: "subjectName")
29            val bool = extras.getBoolean( key: "choice")
30            val articleList = Info.getInformation()
31            val finder: InformationSheet = articleList[value]
32
33            binding.tvTitle.text = finder.title
34            binding.tvPara1.text = finder.paraOne
35            binding.tvPara2.text = finder.paraTwo
36
37            if(bool){
38                val button = binding.btnQuestionClick
39                button.setOnClickListener {  it: View!
40                    val intent = Intent( packageContext: this, MCQuestionPage::class.java)
41                    intent.putExtra( name: "key", value)
42                    intent.putExtra( name: "subjectName", subjectName)
43                    startActivity(intent)
44                    finish()
45                    }
46
47            } else {
48                val button = binding.btnQuestionClick
49                button.setOnClickListener {  it: View!
50                    val intent = Intent( packageContext: this, FBQuestionPage::class.java)
51                    intent.putExtra( name: "key", value)
52                    intent.putExtra( name: "subjectName", subjectName)
53                    startActivity(intent)
54                    finish()
```

Figure 18, snippet of 'InformationPage' code

5. Question Page Part A – FBQuestionPage

Section 5 is the 'FBQuestionPage'. This page is designed for the Fill In The Blank Quiz. It takes the passed information from the intents, gets Questions in the appropriate format and through the quiz calculates how well the user did while giving a small amount of feedback before sending these results on to the Quiz Result Page.

If the user has selected to fill in the Blank quiz, the following code will be exercised. Questions with the Fill the Blank specification are acquired from a specific modular file in the program. For example, the Benefits or Savings questions are stored in 'FBBenefits' and 'FBSavings' respectively, which are objects using an array of the data class file 'FBQuestion' to store the questions for the user. Figure 19 shows the code randomly order the code so that it is not the same each time it is answered. This is performed through lines 75 to 85. On line 85, the 'fbQuestionList' variable is set to equal this random array.

Then the 'setQuestion' function is called. This function shown in Figure 20 is used to set the question that are being asked initially. It checks that the user's current position in the quiz question list is not greater than the total size of the quiz itself. If it is, then the quiz must end, parts of the layout are hidden, and the submit button no longer submits questions but starts the intent for the 'ResultsPage'. Otherwise, the questions are formatted with the next question.

When the quiz is not finished, the 'btnSubmit' checks that the inputted value is not empty when clicked. Suppose it contains an input, then the 'answerChecker' function checks if the answer is correct. This is found in Figure 21 which shows that, if it does not contain an input, it asks for an answer and does not submit the answer until there is some input. If the user and correct answers are the same, the 'fbCorrectAnswers' is iterated by 1. Feedback is given to the user with a green text message to let them know it was correct on lines 106-107. If it is incorrect, Lines 110-112 show a red message that will display and give the correct answer as feedback.

```
75      val randomArray: ArrayList<FBQuestion> = ArrayList()
76      while (fbQuestionList!!.size != 0) {
77          val random = (0 until ((fbQuestionList!!.size))).random()
78          if (randomArray.size == totalQuestionSize) {
79              break
80          } else {
81              randomArray.add(fbQuestionList!![random])
82              fbQuestionList!!.removeAt(random)
83          }
84      }
85      fbQuestionList = randomArray
86      binding.pbProgress.max = fbQuestionList!!.size
87
88      setQuestion()
```

Figure 19, snippet of 'FBQuestionPage' code

```
121     private fun setQuestion() {
122         binding.teInput.hint = ""
123         binding.pbProgress.progress = fbCurrentPosition
124         if (fbCurrentPosition > fbQuestionList!!.size) {
125             binding.tvFirstSentence.visibility = View.GONE
126             binding.tvSecondSentence.visibility = View.GONE
127             binding.pbProgress.visibility = View.GONE
128             binding.teInput.visibility = View.GONE
129             binding.btnSubmit.text = "Results"
130             binding.btnSubmit.setOnClickListener { it: View!
131                 Toast.makeText( context: this,  text: "Calculating Final Results", Toast.LENGTH_SHORT).show()
132                 Handler(Looper.getMainLooper()).postDelayed({
133                     sendIntent()
134                 },  delayMillis: 1000)
135             }
136         } else {
137
138             finder = fbQuestionList!![fbCurrentPosition - 1]
139
140             if (fbCurrentPosition == fbQuestionList!!.size) {
141                 binding.btnSubmit.text = "Submit"
142             } else {
143                 binding.btnSubmit.text = "Submit"
144             }
145             if (finder.firstString == "") {
146                 binding.tvFirstSentence.visibility = View.GONE
147             } else {
148                 binding.tvFirstSentence.visibility = View.VISIBLE
149                 binding.tvFirstSentence.text = "\"" + finder.firstString + "..."
150             }
151             if (finder.secondString == "") {
152                 binding.tvSecondSentence.visibility = View.GONE
153             } else {
154                 binding.tvSecondSentence.visibility = View.VISIBLE
155                 binding.tvSecondSentence.text = "..." + finder.secondString + "\""
156             }
157             fbAnswer = finder.answer
158         }
159     }
```

Figure 20, snippet of 'FBQuestionPage' code

```
90          val textInput = binding.teInput
91
92          binding.btnSubmit.setOnClickListener { it: View!
93              if (binding.teInput.text.toString().isNotEmpty()) {
94                  answerChecker()
95                  textInput.text?.clear()
96              } else {
97                  textInput.hint = "Please provide an answer"
98              }
99          }
100     }
101
102     private fun answerChecker() {
103         val answer = binding.teInput.text.toString().trim()
104         if (answer.uppercase() == fbAnswer?.uppercase()) {
105             fbCorrectAnswers++
106             binding.tvFeedback.setTextColor(Color.parseColor( colorString: "#64dd17"))
107             binding.tvFeedback.text = "Correct, well done!"
108
109         } else {
110             binding.tvFeedback.setTextColor(Color.parseColor( colorString: "#c30000"))
111             binding.tvFeedback.text =
112                 "Incorrect, the answer was: ${finder.answer.replaceFirstChar { finder.answer[0].uppercase() }}"
113         }
114
115         if (fbCurrentPosition <= fbQuestionList!!.size) {
116             fbCurrentPosition++
117             setQuestion()
118         }
119     }
```

Figure 21, snippet of 'FBQuestionPage' code

6. Question Page Part B – MCQuestionPage

Section 5 comes from a different selection of the 'ChoicePage', 'MCQuestionPage' is a piece of code that enables a multiple-choice quiz to be had. The program takes the information from the intent sent by 'informationPage'. Gets questions in the correct format and displays them, enabling the user to answer the questions. The complexity is slightly more significant, but in principle, the code is the same as the 'FBQuestionPage'.

The beginning of the code in retrieving data and sorting the questions is essentially the same as Section 4. However, how they deviate is pretty complex and will be explained to the best possible extent. The differences begin when 'setQuestion' is called. As Figure 22 shows line 206, 'defaultAnswers' is called, which is a function that removes any feedback and adds the correct options so the user can successfully answer the question. Two variables called 'submit' and 'selectAnswer' are set to false and true, respectively. These will be used throughout the code for the program to operate. These variables are used to check the stage of answering the question. The rest of the function sets the display.

The next part of the program to understand is the changes to the default functions of Android. The 'onClick' function was overwritten as shown in Figure 23 and 24 so that when the 'selectAnswers' variable equals true, the function allows the optional answers to call a function called 'selectedAnswer', which takes the clicked answer as an argument. The function allows the user to change their mind on an answer by setting all answers to default before highlighting the selected option in yellow, with bold text. The 'onClick' function also makes the submit variable equal true. These features are shown in Lines 129-131. Now that an answer has been selected and submitted is true, lines 135-137 are reachable. If the user clicks the button to submit their answer, a series of checks are carried out.

Line 138 indicates that a check works out if 'selectAnswer' is false. If it is, then the current position is incremented by one. It then checks the current position of the quiz by the quiz question array. If it is, it calls the 'setQuestion' function to reset the variables and change the question. Otherwise, they are equal and will exit the application with an intent. If 'selectAnswer' is true, a variable called option will get the question object from the 'mQuestionList' array at the correct index. It will then check if the answer is correct and use an 'answerResult' function to give the correct feedback.

```
201     private fun setQuestion() {
202         submit = false
203         selectAnswer = true
204         finder = mQuestionList!![mCurrentPosition - 1]
205         binding.pbProgress.progress = mCurrentPosition
206         defaultAnswers()
207
208         if (mCurrentPosition == mQuestionList!!.size) {
209             binding.btnSubmitButton.text = "Submit"
210         } else {
211             binding.btnSubmitButton.text = "Submit"
212         }
213
214         binding.tvQuestionString.text = finder.question
215         binding.tvAnswer1.text = finder.optionOne
216         binding.tvAnswer2.text = finder.optionTwo
217         binding.tvAnswer3.text = finder.optionThree
218         binding.tvAnswer4.text = finder.optionFour
219     }
220 }
```

Figure 22, snippet of 'MCQuestionList' code.

```
114     override fun onClick(v: View?) {
115         if (selectAnswer) {
116             when (v?.id) {
117                 R.id.tv_answer1 -> {
118                     selectedAnswer(binding.tvAnswer1,  selectedAnswer: 1)
119                     submit = true
120                 }
121                 R.id.tv_answer2 -> {
122                     selectedAnswer(binding.tvAnswer2,  selectedAnswer: 2)
123                     submit = true
124                 }
125                 R.id.tv_answer3 -> {
126                     selectedAnswer(binding.tvAnswer3,  selectedAnswer: 3)
127                     submit = true
128                 }
129                 R.id.tv_answer4 -> {
130                     selectedAnswer(binding.tvAnswer4,  selectedAnswer: 4)
131                     submit = true
132                 }
133             }
134         }
135         if (submit) {
```

Figure 23, snippet of 'MCQuestionList' code.

```
                R.id.tv_answer4 -> {
                    selectedAnswer(binding.tvAnswer4, selectedAnswer: 4)
                    submit = true
                }
            }
        }
        if (submit) {
            when (v?.id) {
                R.id.btn_submit_button -> {
                    if (!selectAnswer) {
                        mCurrentPosition++
                        when {
                            mCurrentPosition <= mQuestionList!!.size -> {
                                setQuestion()
                            }
                            else -> {
                                val intent = Intent( packageContext: this, QuizResult::class.java)
                                intent.putExtra( name: "subjectName", mSubjectName)
                                intent.putExtra( name: "correctAnswers", mCorrectAnswers)
                                intent.putExtra( name: "totalQuestions", mQuestionList!!.size)
                                intent.putExtra( name: "multipleChoice", value: true)
                                startActivity(intent)
                                finish()
                            }
                        }
                    } else {
                        val option = mQuestionList?.get(mCurrentPosition - 1)
                        if (option!!.correctAnswer != mSelectedAnswer) {
                            answerResult(mSelectedAnswer, R.drawable.question_wrong_answer)
                            answerResult(option.correctAnswer, R.drawable.question_correct_answer)
                        } else {
                            mCorrectAnswers++
                            answerResult(option.correctAnswer, R.drawable.question_correct_answer)
                        }

                        if (mCurrentPosition == mQuestionList!!.size) {
                            binding.btnSubmitButton.text = "Results!"
                        } else {
                            binding.btnSubmitButton.text = "Next Question"
                        }
                        selectAnswer = false
                        mSelectedAnswer = 0
                    }
```

Figure 24, snippet of 'MCQuestionList' code.

7. ResultPage

The 'ResultsPage' is a page that is designed to display to the user some summative feedback. It takes the scores from both quizzes and displays the results while also adding this information to the database.

Following the submission of the quizzes, the 'ResultPage' immediately handles the intent sent by grabbing the data as well as creating some time variables to take the current time the activity was started. It works out the ordering of the completed quizzes in the database when displayed to the user. The program takes several of these variables and displays the results to the user as feedback.

A value called dao is designed to allow access to the databases functions and invoke an instance of the quiz. When loops are used to give the correct id for the database by using the subject name, it is passed to indicate the correct id number for the database. The program uses the Boolean value to determine which block of the When loop to assign the id number for the database. As Figure 25 shows, the database then uses the variables to input data into a value called 'quizResult', and this is then inserted into the database through the dao value. The 'insertOrUpdate' object is displayed in Figure 26 and inserts the data or replaces it. When ready, the user will click on a button that will launch an intent to go back to the home screen.

```
100         val quizResult = QuizResult(
101             primaryId = id,
102             subjectName = subjectName,
103             correctQuestionNumber = correct,
104             totalQuestionNumber = totalQuestions,
105             checkTime = compareDate,
106             typeOfTest = typeOfTest,
107             displayTime = displayTime
108         )
109
110         dao.insertOrUpdate(quizResult)
```

Figure 25, snippet of 'MCQuestionList' code.

```
6   @Dao
7   interface QuizDao {
8
9       @Insert(onConflict = OnConflictStrategy.REPLACE)
10      fun insertOrUpdate(quizResult: QuizResult)
11
12      @Query( value: "Select * From results")
13      fun getAllResults(): List<QuizResult>
14
15
16  }
```

Figure 26, snippet of 'QuizDao' code.

Progress and Support Fragments

This next section is much shorter than the last segment and outlines Use Case 2 and 3. The program demonstrates how users can review their results and track their progress by using a fragment that displays the database. It will then show how the support system works.

1. ProgressFragment

Section 1 is designed to demonstrate how the application allows a user to track their progress. The application is assumed as started already. When looking at the 'ModuleFragment', the default fragment opens when the application is launched. The user must click the "Progress" tab from the bottom navigation menu. Once there, the user will be able to see one of two things.

If they have not completed a quiz, the screen will display a message by default in the layout for the frame, which will alert the user that a quiz has not been completed and, therefore, no data can be displayed. The database works this out by invoking an instance of the database. As shown by line 30 in Figure 27, the check occurs on lines 32-38. If the results are empty, then the warning is displayed. Otherwise, it is hidden and vice versa for the database results if they have been created.

Then the results are displayed on lines 40 and 42. First, a value of 'resultAdapter' is called. It is assigned as a ResultsAdapter and calls the previously invoked database variable with a function that displays all results featured in Figure 26. It displays all the database results and sorts them into descending order, making the most recent quiz appear at the top.

```
30          dao = QuizDb.invoke(requireContext()).getDao()
31
32          if (dao.getAllResults().isEmpty()) {
33              binding.tvWarning.visibility = View.VISIBLE
34              binding.recyclerView.visibility = View.GONE
35          } else {
36              binding.tvWarning.visibility = View.GONE
37              binding.recyclerView.visibility = View.VISIBLE
38          }
39
40          val resultsAdapter = ResultsAdapter(dao.getAllResults().sortedByDescending { it.checkTime })
41
42          binding.recyclerView.adapter = resultsAdapter
43
44          return binding.root
45      }
46
```

Figure 27, snippet of 'QuizDao' code.

2. SupportFragment

Section 2, the last part of the application, is designed to allow users to contact support if they have money problems and related problems such as Mental Health. It provides methods to call and visit the websites of various charities.

Like the 'ProgressFragment, to reach this part of the app, the bottom navigation in the 'MainActivity' must be used. By using the "support" tab, the fragment page opens. As shown in Figure 28, one charity is assigned some variables between lines 148 and 151. These variables display the charity information to the user and turn the text to several buttons through several 'setOnClickListener's. These Listeners use the variables information in two intents. Unlike the earlier intents, these intents effectively close the application. The 'phoneIntent' opens the "Phone" app that allows calls to be made by the mobile phone. The intent inputs the number to be instantly called by the user. Meanwhile, similarly, the 'webIntent' actions a web search, enabling the passed variable website to be found by the user.

```
147
148         val char6Name = "Samaritans"
149         val char6Phone = "tel:116 123"
150         val char6PhoneDisplay = "116 123"
151         val char6Web = "samaritans.org/"
152
153         binding.tvCharitySix.text = char6Name
154         binding.tvCharitySixPhone.text = char6PhoneDisplay
155         binding.tvCharitySixPhone.setOnClickListener {  it: View!
156             phoneIntent(char6Phone)
157         }
158         binding.imPhoneSix.setOnClickListener {  it: View!
159             phoneIntent(char6Phone)
160         }
161         binding.tvCharitySixWebsite.text = char6Web
162         binding.tvCharitySixWebsite.setOnClickListener {  it: View!
163             webIntent(char6Web)
164         }
165         binding.imEmailSix.setOnClickListener {  it: View!
166             webIntent(char6Web)
167         }
168     }
169
170     private fun phoneIntent(phoneNumber: String) {
171         val intent = Intent(Intent.ACTION_DIAL)
172         intent.data = Uri.parse(phoneNumber)
173         startActivity(intent)
174     }
175
176     private fun webIntent(website: String) {
177         val intent = Intent(Intent.ACTION_WEB_SEARCH)
178         intent.putExtra(SearchManager.QUERY, website)
179         startActivity(intent)
180     }
```

Figure 28, snippet of 'SupportFragment' code.

Designs

It is essential to know showcase the application's design that the user views. The detail will be kept minimal as many layouts are several hundred lines long. Therefore, screenshots of the application and a short description of what they do and their page will be used. It will demonstrate the successful implementation of the Functional and non-Functional Requirements where relevant.

It would be beneficial to remind the reader at this time of the functional and non-functional requirements of the application.

Functional Requirements

FR 1: The user is able to choose a subject to take a test on

FR 2: The user is able to choose the type of quiz they would like to take
FR 3: After choosing a subject and type of quiz, the user is able to access an information page that is relevant
FR 4: The user is able to complete a quiz on the chosen subject
FR 5: The user's quiz score is retained in a database
FR 6: The user is able to view their score in a database

Non-Functional Requirements

NFR 1: The user is given feedback after each question to learn from their mistakes
NFR 2: The user is able to see a recap of their score on a results page
NFR 3: The user's is able to access support to charities
NFR 4: The application has a logo and title

SplashScreen

The SplashScreen is designed to make the application seem like a professional application. It fulfils NFR 4, which is to satisfy the branding elements of the application. As shown in Figure 29, the display is not very complicated. It features the applications name and its logo in the centre of the screen. The font used for the application logo and name here is known as Brittanic. The reasoning for this is to demonstrate the "British" marketing focus of the application. The name itself is also designed to convey this factor. The Pound Sterling is colloquially associated with the United Kingdom. Therefore, an application with the name "PoundHelp" and an iconic logo can help the marketing strategy play to its specific target audience.

ModuleFragment

The 'ModuleFragment' is effectively the application's home screen and is called so in the bottom navigation bar, as shown in Figure 30. It enables the user to pick from various subjects, eight in total, which shows it fulfils FR 1. Has suitable branding to continue from 'SplashScreen'. This page is where the user is taken when they complete a test. Notably, there is a high degree of contrast across the entire application for the user to see all the information.

ProgressFragment

This fragment is used to display the user's past scores in the quizzes. It is designed to show a default message when the database is empty. As shown by Figure 31, the message is large enough and readable for the user to understand the page's purpose. Figure XX shows what the progress bar looks like with one entry. Figure 32 shows the progress bar with one entry. Notice that the default message is no longer displayed. It also demonstrates some details about the product.

Figure 33 demonstrates the differences between each database entry. For example, the subject head is clear and large enough to be viewable for the user to identify which database entry corresponds to which email quickly. They are informed when the last attempt of the quiz type was and the type of quiz. Which also shows that the database is placed in date order. Other information

is shown relevant to the user's scores, then at the bottom is a 5-star system that helps users visually understand how well they did on a test. It demonstrates the successful implementation of FR 5 and FR 6.

SupportFragment

The design shown in Figure 34 demonstrates the support page. The page is intended to allow users to contact support from charities. As can be seen, it briefly mentions the type of support it gives. It also allows for a user to touch both images and text of the charities to contact them. This design corresponds to fulfilling NFR 3.

ChoicePage

This page is pretty simple. It follows the choice of subject of the 'ModuleFragment'. It is designed to be minimal. It allows a user to choose the type of quiz they wish to take. Clicking either button will take them down the correct path, showing that the app successfully meets FR 2 requirements. It is displayed in Figure 35.

InformationPage

The 'InformationPage' is intended to show the chosen subject's information and display it to the user. An example of the information is shown in Figures 36 and 37. The figures show that the relevant selected subject, in this instance, Homes, can be chosen and the information retained to be displayed after the ChoicePage. The information is scrollable to allow the reader to view all the content showing that it adheres to FR 3.

FBQuestionPage

The 'FBQuestionPage' page displays the chosen quiz type if selected by the user. The design will be bullet-pointed to reference several figures to discuss the general flow of the page:

- In Figure 38 we can see an example of the page when opened answering the 'Homes' subject. It shows a page where a user can input an answer amongst a sentence. Depending on the question asked, it will determine where the answer is placed, start, middle or beginning.
- Figure 39 demonstrates the appearance of the application when typing the answers.
- Figure 40 then shows what the page looks like after a correct response with feedback.
- Figure 41 contrastingly shows the page after an incorrect response with feedback.
- Figure 42 shows the display after the last question has been asked.

Overall, the 'FBQuestionPage' successfully demonstrates that the application enables the feedback of answers and the completion of a quiz. Showing it meets the requirements of FR 4 and NFR 1.

MCQuestionPage

Similarly to the ‘FBQuestionPage’, ‘MCQuestionPage’ is the page which displays when the multiple-choice option is selected. The design will again be bullet-pointed and referenced with figures about how it flows.

- In Figure 43 we can see an example of the page when opened answering the ‘Homes’ subject. Four options are available to be clicked with a question visible.
- Figure 44 demonstrates the appearance of the application when selecting an answer.
- Figure 45 then shows what the page looks like after a correct answer.
- Figure 46 contrastingly shows the page after an incorrect response with feedback informing the user of the correct answer.

As is evident, like, ‘FBQuestionPage’, the multiple choice quiz can also show the application meets the requirements of FR 4 and NFR 1.

ResultsPage

After clicking the relevant buttons in whichever the last quiz they took, the user is brought to the results page. This page is intended to inform the user of their results in the form of summative feedback. As Figures 47 and 48 show, the scores depend on the user's scores. Written feedback also changes per the scores, and this shows the successful implementation of NFR 2. The user is given a page that will also help them return home.

Figure 29, 'SplashScreen' display

Figure 30, 'ModuleFragment' display

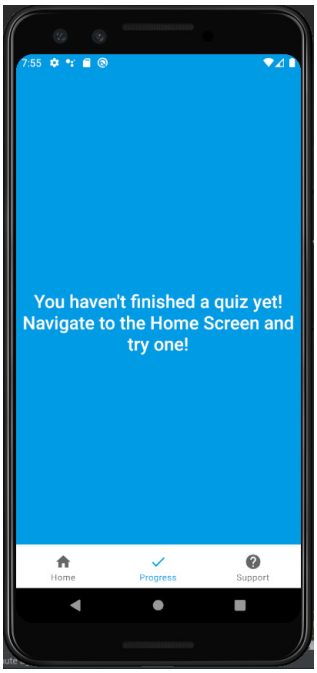


Figure 31, 'ProgressFragment' display

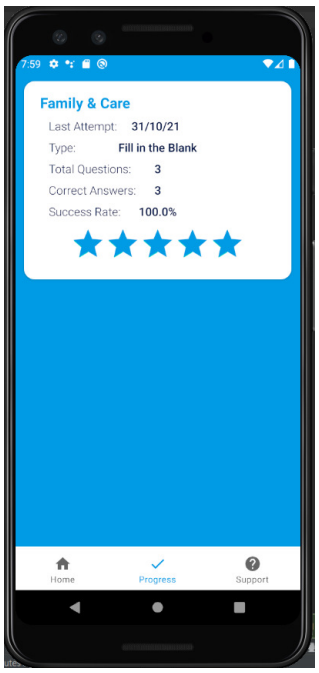


Figure 32, 'ProgressFragment' display

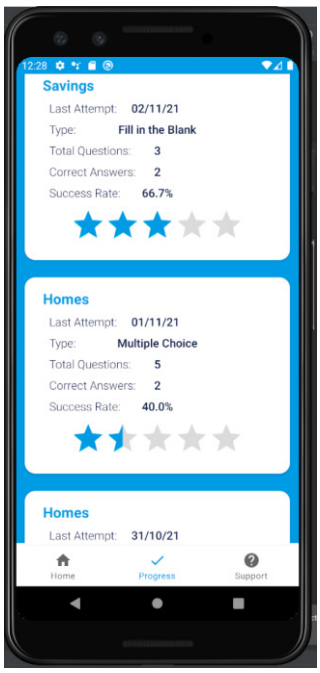


Figure 33, 'ProgressFragment' display

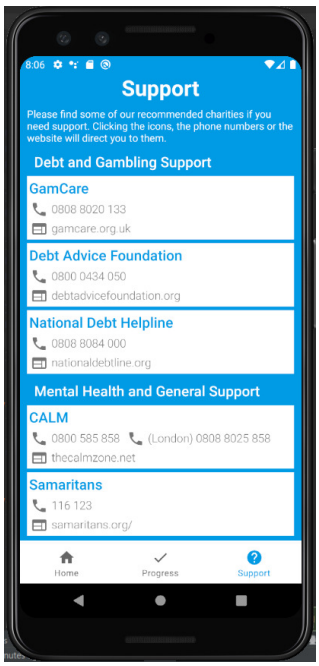


Figure 34, 'SupportFragment' display

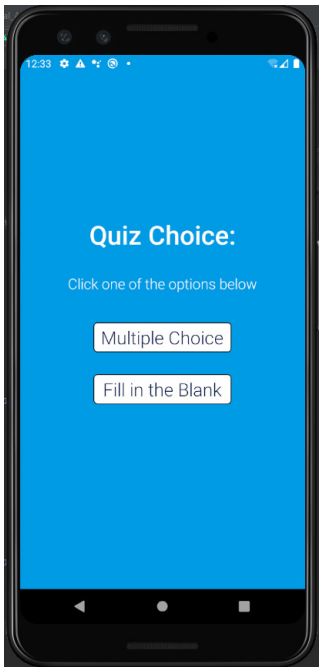


Figure 35, 'ChoicePage' display

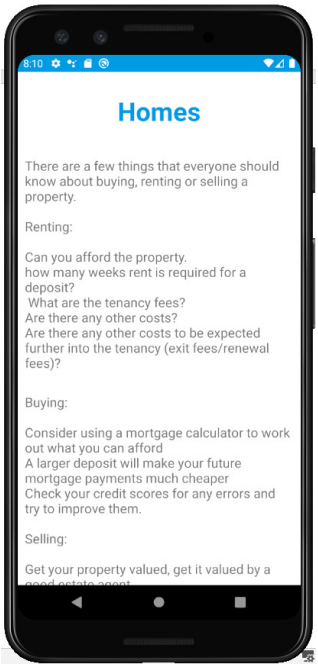


Figure 36, 'InformationPage' display

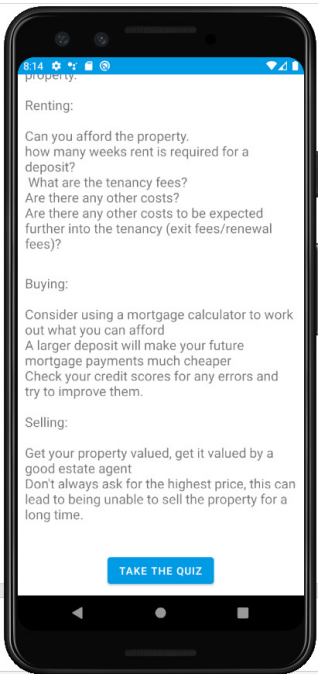


Figure 37, 'InformationPage' display

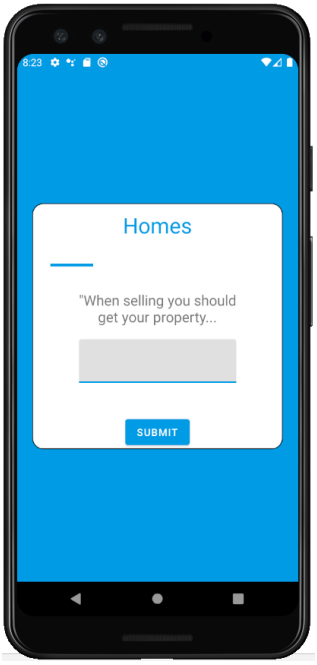


Figure 38, 'FBQuestionPage' display

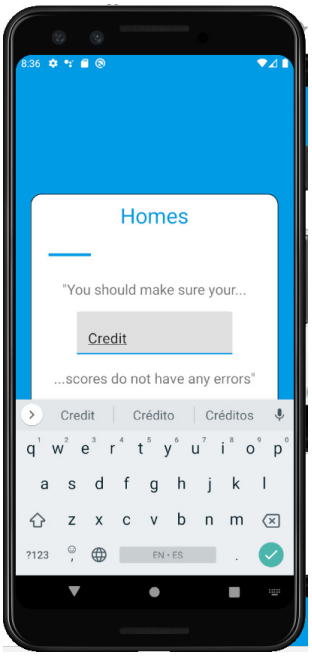


Figure 39, 'FBQuestionPage' display

Figure 40, 'FBQuestionPage' display

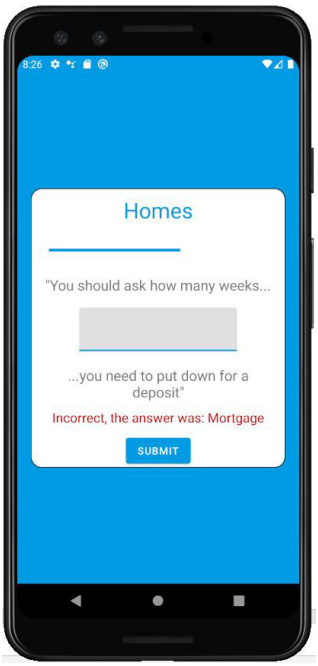


Figure 41, 'FBQuestionPage' display

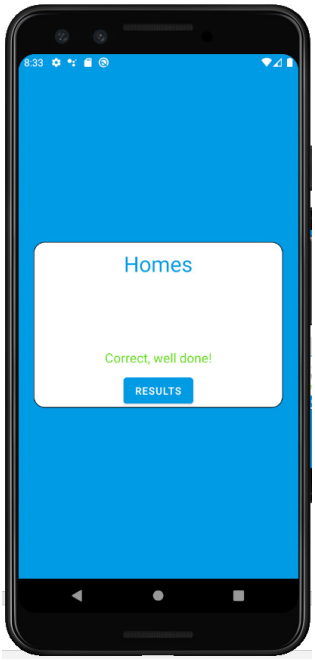


Figure 42, 'FBQuestionPage' display

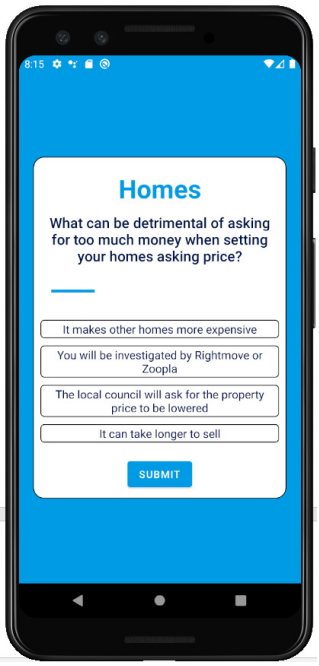


Figure 43, 'MCQuestionpage' display

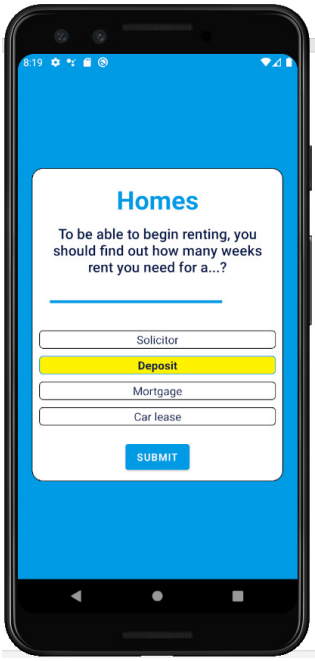


Figure 44, 'MCQuestionpage' display

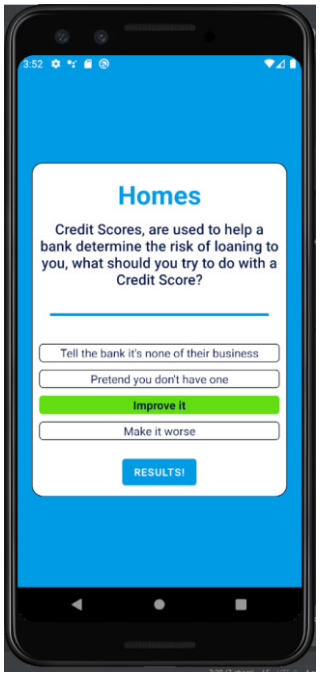


Figure 45, 'MCQuestionpage' display

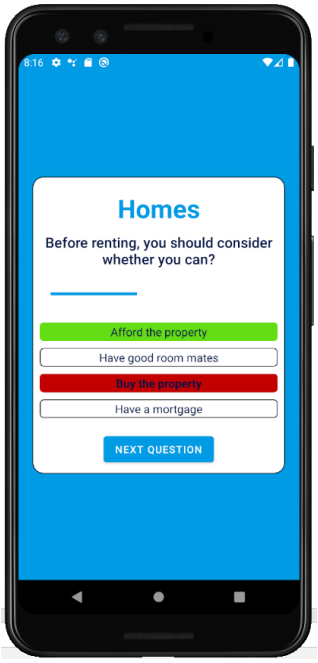


Figure 46, 'MCQuestionpage' display

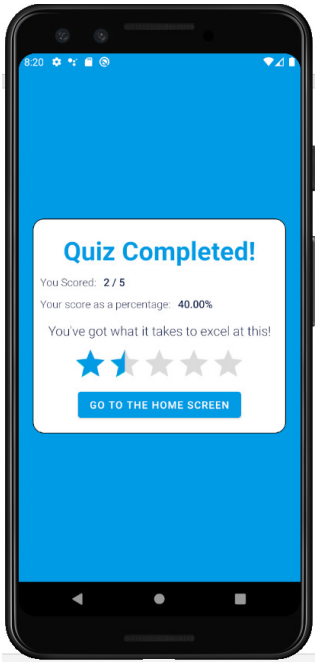


Figure 47, 'ResultPage' display

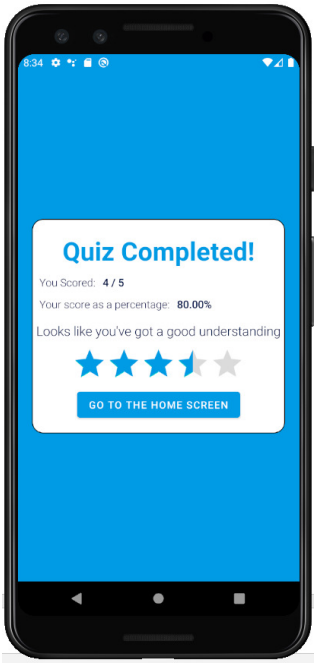


Figure 48, 'ResultPage' display

Conclusions

This last and final chapter is split into two sections. The first segment will summarise the achievements of the project. It will objectively consider both the successes and the failures of the project, reflect on what could or should have been done in different circumstances. It will consider the contribution of the work to the existing current solutions to the problem. The second section will be a self-reflection in the form of a critical assessment of the skills developed in the undertaking of this dissertation.

Conclusions of the Assessment

This section will now summarise the achievements of the work undertaken in this dissertation. Equally, it will also critically assess its deficiencies. It will reflect on what could have been done if more time had been available or if things had gone differently. Therefore, this evaluation will be objective in its reasonings, both describing the triumphs and disappointments.

Successes

There were many successes in the output of this project. In order to appropriately understand the successes of the dissertation, we should remind ourselves about the objectives of this application. Therefore, we should consider how each of these objectives has been met. These are listed below:

- Demonstrate that there is a problem that needs a solution in financial literacy through researching the topic.
- Show that a mobile application is a unique solution by providing a literature review of the current market solutions available to customers and demonstrating that such a product could be profitable.
- To create a product that could successfully solve the problem of financial literacy within the UK by providing information and quizzes to take and providing feedback. It must then show that it meets these requirements as a minimum viable product.
- Justify the critical decisions made during the creation of the product and prove how the following software development frameworks enabled the successful implementation of the product.

To begin, I believe this application has demonstrated more than enough research on the problem of financial literacy. It has used relevant and up to date sources wherever possible to make the research reliable and as accurate as possible. It has argued authoritatively with the reader informing them about the problem and its need for a solution.

Because I was able to understand the need for a solution. I conducted a literature review that went into great detail to explore the financial literacy solution market. From this research, I understood that the ‘financial literacy mobile application market’ does not exist. I estimated the value of such a market and worked out that it was large enough to be profitable. I was able to examine the target audience then that such a market would have. I observed the current competitors in the field and understood a business opportunity for a mobile application.

I was then successfully able to demonstrate that I had successfully created a solution to the financial literacy problem by taking the knowledge from the literature review to create a product that could meet the requirements of a solution to the financial literacy problem. This dissertation was able to justify the decisions made in how the work was conducted, such as explaining why the Agile software development framework was chosen to organise the project management.

Deficiencies

Although the main objectives as listed in the successes segment have shown, the overall work has been very successful in meeting the project's aims. It should be clear that in some aspects, there have been some minor failures. Most of these problems are more specific than the achievements of the successes.

One deficiency was the failure to investigate enough that I could have programmed the application to work over both iOS and Android devices. Initially, this did not cross my mind as I became aware that iOS and Android used different coding languages. Swift is only for iOS and Android uses several, including such as my chosen language Kotlin. However, I eventually realised that according to kotlinlang.org, Kotlin allows this. That being said, this would have required a MacOS device to have programmed with, I might have been able to use a Virtual Machine to do this as I do not own an Apple computer nor an iPhone to test on, but the performance of using a Virtual Machine probably would have made programming difficult. I also did not feel confident in using MacOS or a Virtual Machine for the first time.

Despite providing links to charities so that users in distress can find help in their time of need. I have also failed them. The reason for this is that I have been unable to correct an error successfully, this error does not crash the application, but it does minimise it. To be more specific, when clicked, the application will often launch the relevant browser of the phone and then close it before it finds the relevant charity. As mentioned, this is not a crash. I'm not sure what the cause is. Especially as the application can open the link and the website during the times it works.

I also feel as though the outputted product is very minimal, even for a minimal viable product. For example, the database is not as fleshed out as it could be. In most applications where you might want to track progress, you would want to see the results of all attempts

over time and have this visualised in some form of a graph. Instead, my application only saves and updates the most recent attempt. If there has never been an attempt, then an attempt is inserted because of my failure to successfully implement a database as quickly as I would have liked. It took the longest of all of my developments.

The survey I conducted was poor, and I realised this too late. Most apparent is the phrasing of questions. For example, I asked how confident people were in their financial literacy, 0 being unconfident and 10 being confident. What I meant to say was, 0 very unconfident, 10 very confident. This error I repeated several times, with the following questions compounding the problem. As a result, I am unsure of whether these statistics can be considered valid. Either the participants knew what I meant to ask and would score themselves low or high in the question, or they felt they were just being asked on a smaller scale and so mostly felt just above average in their confidence.

Reflection

There was plenty of skills and knowledge gained on an extensive degree of subjects while undertaking this project. This section will critically consider the lessons learned—both in terms of the process of research and about myself.

Process of Research

Probably the most crucial factor that determined the outcome of this application was the Time Constraints. Therefore, the primary consideration in my decision-making process involved its effects on the time I had left. For example, time helped determine that the project must be undertaken using the Agile software development framework. As explored in the Approach chapter, Agile is designed to be time flexible and therefore more malleable to changes if delays occur. This is because by creating a minimal viable product as opposed to a fully developed application that other frameworks like Waterfall would require, the application can afford to lose a small amount of functionality when a sprint is not completed on time. However, in the case of Waterfall, a problem in development would cause the outright delay of the release of the application until it was fully functional, something that I needed to avoid. By appreciating the time available and my initial skills, I developed good time management skills and further improved my project management skills for future projects.

I also had to learn how mobile applications were created. Having never experienced any form of development in this field, I had a lot to learn, and this was an ongoing process. I believe that I failed myself in my ability to do this aspect of my research. Early on, I did not find the most useful tutorials to learn Kotlin that I could have. These had out of date information that hindered both my understanding and my progress in completing the application. That being said, I do feel very competent in my understanding of Kotlin. I enjoyed the language, and it's perhaps my favourite programming language. Following this project, I will look to become a mobile application developer.

Lessons Learned

Due to the nature of the project, I was able to identify several key aspects of my personal development that I was able to identify and will look to improve. Probably the most important was my determination. At times I struggled with the project due to its breadth, having never created any software before. However, I was able to recognise that this skill was crucial in this type of project. Equally and previously touched upon were my time-management skills. I have realised that this is something I need to improve significantly on. I find myself distracted and unable to utilise the time I have available. I was fortunately managed to complete the dissertation on time. However, it led to several stressful nights where I worked for long periods of time. Therefore, I know that I need to utilise my time more efficiently.

In terms of the skills I had already, I found that I was more resilient than I had previously thought. I had several setbacks due to developing a language I was unfamiliar with and with incomplete and sometimes imprecise guidelines on achieving my objectives. Yet, I found that I did not give up. For example, I struggled to create a unified database for the application, and the guides I followed seemed to be inaccurate. Yet, my resilience to both continues to try and remind myself that many programmers had come before me and learnt what I was experiencing means that I was more resilient than I thought.

In tandem with this, I found that my adaptability was very useful by allowing myself to not rigidly keep trying methods that failed and being open to different ways of trying things allowed my skills as a developer to improve. While coding, I decided to change the already working code on a few occasions to do other things as I saw fit. I improved my understanding of what I was writing and therefore improved my understanding of the Kotlin programming language.

Probably the best and most important thing I learnt about myself was that I do enjoy development. Taking on such a large project helped me learn that development was more than just code. That theory and research are equally if not more important, and without them, projects can be executed in the wrong way.

This project has certainly helped me realise a few things about myself. It has aided my personal development by giving me the confidence to know that, when well organised, big projects like this one can be achieved. It’s shown that I have natural entrepreneurial skills to identify business opportunities in real-life issues and formulate them into coherent reasonings for these ideas to be successful.

References

Introduction – The Problem

Barnes, O. and Megaw, N. (2021). *UK banks expect deluge of 'pay as you grow' requests from SMEs*. [online] www.ft.com. Available at: https://www.ft.com/content/b6e50a11-437d-48c9-8610-f9c5f9e3dde3 [Accessed 6 Oct. 2021].

Barnfield-Tubb, J. and Atkinson, G. (2021). *Gambling Participation: Activities and Mode of Access - July 2021*. [online] Gambling Commission, Gambling Commission, pp.5–5b. Available at: https://www.gamblingcommission.gov.uk/statistics-and-research/publication/statistics-on-participation-and-problem-gambling-for-the-year-to-june-2021 [Accessed 12 Sep. 2021].

Bond, N. and Holkar, M. (2018). *A Silent Killer: Breaking the Link between Financial Difficulty and Suicide*. [online] Money and Mental Health Institute, London: Money and Mental Health Institute, pp.5–42. Available at: https://www.moneyandmentalhealth.org/wp-content/uploads/2018/12/A-Silent-Killer-Report.pdf [Accessed 12 Sep. 2021].

Brennan, H. (2021). *A Million Pensioners hit with National Insurance Tax Bills for First Time Ever*. The Telegraph. [online] 7 Sep. Available at: https://www.telegraph.co.uk/pensions-retirement/tax-retirement/million-pensioners-hit-national-insurance-tax-bills-first-time/ [Accessed 10 Sep. 2021].

Bucher-Koenen, T., Alessie, R., Lusardi, A. and van Rooij, M. (2021). *Fearless Woman: Financial Literacy and Stock Market Participation*. [online] Global Financial Literacy Excellence Center. Available at: https://gflec.org/wp-content/uploads/2021/03/Fearless-Woman-Research-Final.pdf?x63881 [Accessed 25 Oct. 2021].

Cherry, L. (2021). *U.K. Wage Inflation Emerges With Post-Lockdown Staff Shortages*. Bloomberg.com. [online] 8 Jul. Available at: https://www.bloomberg.com/news/articles/2021-07-08/u-k-wage-inflation-emerges-with-post-lockdown-staff-shortages [Accessed 13 Sep. 2021].

Collins, P. (2018). *Wonga Made Borrowing Too Easy – None of us Will Mourn its Demise* | Patrick Collinson. [online] the Guardian. Available at:

https://www.theguardian.com/business/2018/aug/31/wonga-borrowing-payday-lender [Accessed 10 Sep. 2021].

Cowburn, A. (2021). *Starmer says Benefit Cut and National Insurance "double whammy attack" on Workers*. [online] The Independent. Available at: https://www.independent.co.uk/news/uk/politics/starmer-workers-johnson-national-insurance-b1918701.html [Accessed 13 Sep. 2021].

Cumbo, J. (2021). *UK Women Win £2.7bn Top-up to Underpaid State pensions*. [online] www.ft.com. Available at: https://www.ft.com/content/29a77d92-5047-414e-b43a-e42c0faa99ce [Accessed 13 Sep. 2021].

Elliott, L. (2020). *Furlough Effect Leaves Rishi Sunak with "Triple Lock" Pensions Dilemma*. [online] The Guardian. Available at: https://www.theguardian.com/politics/2020/jun/17/furlough-effect-leaves-rishi-sunak-with-triple-lock-pensions-dilemma [Accessed 10 Sep. 2021].

Emmerson, C. (2020). *COVID-19 Will Bring Forward the Date When the Pensions Triple Lock is Unpicked*. [online] ifs.org.uk. Available at: https://ifs.org.uk/publications/15132 [Accessed 10 Sep. 2021].

Financial Conduct Authority (FCA). (2018). *Retirement Income Market Data 2019/20*. [online] Available at: https://www.fca.org.uk/data/retirement-income-market-data [Accessed 21 Apr. 2021].

Fernando, J. (2021). *Financial Literacy*. [online] Investopedia. Available at: https://www.investopedia.com/terms/f/financial-literacy.asp [Accessed 13 Sep. 2021].

Financial Capability (A 2021). *Financial Capability for Schools* | Financial Capability - FinCap. [online] www.fincap.org.uk. Available at: https://www.fincap.org.uk/en/articles/schools [Accessed 25 Oct. 2021].

Financial Solutions Team, Telegraph (2020). *What Is a Good Pension Pot?* The Telegraph. [online] 6 Jul. Available at: https://www.telegraph.co.uk/financial-services/pensions-advice-service/what-is-a-good-pension-pot/#:~:text=What%20is%20the%20average%20UK [Accessed 30 Aug. 2021].

Finnis, A. (2021). *What the Pensions Triple Lock is, and Why it's Being Suspended Despite Manifesto Pledge*. [online] inews.co.uk. Available at: https://inews.co.uk/news/politics/triple-lock-state-pension-scrapped-double-meaning-suspended-one-year-explained-1188709 [Accessed 10 Sep. 2021].

Gambling Harm — Time for Action. (2020). [online] parliament.uk. House of Lords. Available at: https://publications.parliament.uk/pa/ld5801/ldselect/ldgamb/79/7902.htm [Accessed 12 Sep. 2021].

George, S. and Bowden, H. (2014). *Problem Gambling*. [online] Royal College of Psychiatrists. Available at: https://www.rcpsych.ac.uk/mental-health/problems-disorders/problem-gambling [Accessed 12 Sep. 2021].

Green, N. (2020). *One in Six Over-55s Have No Pension Savings Yet*. [online] unbiased.co.uk. Available at: https://www.unbiased.co.uk/news/financial-adviser/one-in-six-over-55s-have-no-pension-savings-yet [Accessed 7 Apr. 2021].

Harley-McKeown, L. (2021). *Tax Rate on Graduates' Additional Pay Heads Towards 50%*. [online] uk.finance.yahoo.com. Available at: https://uk.finance.yahoo.com/news/uk-graduate-tax-rate-national-insurance-increase-092249491.html [Accessed 10 Sep. 2021].

Holdsworth, L., Nuske, E., Tiyce, M. and Hing, N. (2013). *Impacts of Gambling Problems on Partners: Partners' Interpretations*. Asian Journal of Gambling Issues and Public Health, [online] 3(1), p.11. Available at: https://ajgiph.springeropen.com/articles/10.1186/2195-3007-3-11 [Accessed 12 Oct. 2021].

Jenkins, P. (2021). *Why Financial Literacy Is a Passport to Financial Freedom*. [online] www.ft.com. Available at: https://www.ft.com/content/80480742-9853-4144-9c91-238021414bc8 [Accessed 13 Sep. 2021].

Johnson, B. (2021). *PM Statement to the House of Commons on Health and Social care*: 7 September 2021. [In-Person] Available at: https://www.gov.uk/government/speeches/pm-statement-to-the-house-of-commons-on-health-and-social-care-7-september-2021 [Accessed 10 Sep. 2021].

Klair, A. (2021). *Universal Credit Cut Will Hit Millions of Working Families and Key Workers*. [online] www.tuc.org.uk. Available at: https://www.tuc.org.uk/blogs/universal-credit-cut-will-hit-millions-working-families-and-key-workers [Accessed 21 Aug. 2021].

Kropshofer, K. (2020). *The Pandemic has Triggered a British Online Gambling Crisis*. [online] Wired UK. Available at: https://www.wired.co.uk/article/gambling-uk-online-sites-addiction.

Lane, D. (2021). *How Much do Brits Understand About Their Own Finances?* [online] Freetrade. Available at: https://freetrade.io/news/uk-financial-literacy-test [Accessed 13 Sep. 2021].

London Institute of Banking & Finance (LIBF) (2021). *Young Persons' Money Index* | The London Institute of Banking & Finance. [online] Available at: https://www.libf.ac.uk/study/financial-education/young-persons-money-index.

Money Advice Service, 2019, Financial Capability Survey, 2018, [data collection], UK Data Service, 2nd Edition, Accessed 18 August 2021. SN: 8454, http://doi.org/10.5255/UKDA-SN-8454-2

National Audit Office, House of Commons, UK Government. 2018. *Tackling Problem Debt*. HC 1499, Session 2017-19. Available at: https://www.nao.org.uk/wp-content/uploads/2018/09/Tackling-problem-debt-Report.pdf [Accessed 12 September 2021].

Office for National Statistics (ONS). (2021). *UK Government Debt and Deficit* - Office for National Statistics. [online] Available at: https://www.ons.gov.uk/economy/governmentpublicsectorandtaxes/publicspending/bulletins/ukgovernmentdebtanddeficitforeurostatmaast/march2021 [Accessed 6 Oct. 2021].

Pickford, J. (2015). *One-third of Adults "financially illiterate."* [online] Financial Times. Available at: https://www.ft.com/content/4b0f2656-8f99-11e5-8be4-3506bf20cc2b [Accessed 10 Sep. 2021].

Poterba, J., Venti, S. and Wise, D. (2012). *Were They Prepared for Retirement? Financial Status at Advanced Ages in the HRS and AHEAD Cohorts*. National Bureau of Economic Research, (17824).

Powell, A. and Francis-Devine, B. (2021). *Coronavirus: Impact on the labour market*. [online] Commons Library. Available at: https://commonslibrary.parliament.uk/research-briefings/cbp-8898/ [Accessed 18 Aug. 2021].

Raisin. (2020). *Are you Saving Enough Money to Retire?* | Raisin UK. [online] Available at: https://www.raisin.co.uk/newsroom/articles/retirement-statistics-2020/ [Accessed 30 Apr. 2021].

Saker-Clark, H. (2021). *UK wages soar by 8.8% Year-on-year after Workers Taken off Furlough*. [online] The Independent. Available at: https://www.independent.co.uk/business/uk-wages-soar-by-8-8-yearonyear-after-workers-taken-off-furlough-b1903896.html [Accessed 13 Sep. 2021].

Sierra, R. (2021*). End of Furlough Schemes to Push UK and EU Unemployment Rates Higher*. [online] Fitchratings.com. Available at: https://www.fitchratings.com/research/sovereigns/end-of-furlough-schemes-to-push-uk-eu-unemployment-rates-higher-02-07-2021 [Accessed 21 Aug. 2021].

Starmer, Keir. 2021. "Two and a half million working families will face a double whammy of a national insurance tax rise and a cut to Universal Credit. This is the same old Tories – putting the very wealthiest ahead of working people who have to pick up the bill." [Twitter] 13 September. Available at: https://twitter.com/Keir_Starmer/status/1437319622815068165?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Etweet [Accessed: 13 September 2021].

Introduction – Literature Review

Annual Report 2020. (2021). [online] MoneySuperMarket, p.145. Available at: https://corporate.moneysupermarket.com/~/media/Files/M/Moneysupermarket-V3/result-and-presentations/moneysupermarket-ar20-dps-new-1.pdf [Accessed 13 Oct. 2021].

Blair, I. (2016). *10 Powerful Strategies to Boost Mobile App Downloads - BuildFire*. [online] BuildFire. Available at: https://buildfire.com/ways-boost-mobile-app-downloads/. [Accessed 16 Apr. 2021].

Businesswire.com. (2021). *Software Products Global Market Report 2021: COVID-19 Impact and Recovery to 2030* - ResearchAndMarkets.com. [online] Available at: https://www.businesswire.com/news/home/20210909006012/en/Software-Products-Global-Market-Report-2021-COVID-19-Impact-and-Recovery-to-2030---ResearchAndMarkets.com [Accessed 13 Oct. 2021].

Cadman, J. (2019). *Mobile Internet Statistics 2019* | Finder UK. [online] Finder UK. Available at: https://www.finder.com/uk/mobile-internet-statistics [Accessed 17 Apr. 2021].

Cohen, D. (2019). *The Money Saving Expert: how Martin Lewis Became the Most Trusted man in Britain*. [online] the Guardian. Available at: https://www.theguardian.com/money/2019/jan/31/money-saving-expert-how-martin-lewis-became-most-trusted-man-in-britain-consumer-finance-advice [Accessed 14 Apr. 2021].

Creditangel.co.uk. (2018). *Is the UK Financially Illiterate?* - Credit Angel. [online] Available at: https://www.creditangel.co.uk/blog/is-the-uk-financially-illiterate [Accessed 5 Apr. 2021].

Duolingo. (2021). [Mobile Application] available at: Google Play Store (This image was screenshot on the author of this documents phone) [Accessed 13 Oct. 2021]

Independent Investor. (2020). *How And Why CFD Traders Lose?* [online] Independent Investor. Available at: https://www.independentinvestor.com/cfd/how-to-lose [Accessed 23 Apr. 2021].

IG. (2021). IG Academy. [online] Available at: https://www.ig.com/uk/learn-to-trade/ig-academy [Accessed 13 Oct. 2021].

Khan, S., Modi, V. and Kumar, V. (2020). *Personal Finance Software Market Size, Share* | Industry Growth - 2027. [online] Allied Market Research. Available at: https://www.alliedmarketresearch.com/personal-finance-software-market [Accessed 22 Apr. 2021].

Marketresearch.com. (2020). *Software in the United Kingdom.* [online] Available at: https://www.marketresearch.com/MarketLine-v3883/Software-United-Kingdom-13230698/.

Martin, S. (2014). *UK App Industry to Be Worth £30bn by 2025*. [online] International Business Times UK. Available at: https://www.ibtimes.co.uk/uk-app-industry-be-worth-30bn-by-2025-1454116 [Accessed 13 Oct. 2021].

Moneyadviceservice.org.uk. (2020). [online] Available at: https://www.moneyadviceservice.org.uk

play.google.com A. (2021a). *Mint: Personal Finance Manager – Apps on Google Play*. [online] Available at: https://play.google.com/store/apps/details?id=com.mint&hl=en_GB&gl=US [Accessed 14 Oct. 2021].

play.google.com. (2021b). *IG Academy – Learn to Trade – Apps on Google Play*. [online] Available at: https://play.google.com/store/apps/details?id=com.ig.android.ta&hl=en_GB&gl=US [Accessed 13 Oct. 2021].

Similarweb. (2021a). investopedia.com *Traffic Ranking & Marketing Analytics for September 2021*. [online] Available at: https://www.similarweb.com/site/investopedia.com/#traffic [Accessed 13 Oct. 2021].

Similarweb. (2021b). moneyhelper.org.uk *Traffic Ranking & Marketing Analytics for September 2021*. [online] Available at: https://www.similarweb.com/site/moneyhelper.org.uk/#overview [Accessed 4 Nov. 2021].

Similarweb. (2021c). moneysavingexpert.com *Traffic Ranking & Marketing Analytics for September 2021*. [online] Available at: https://www.similarweb.com/site/moneysavingexpert.com/#overview [Accessed 13 Oct. 2021].

Specification

Gliffy (2020). *What is UML? Everything You Need to Know About Unified Modeling Language*. [online] Gliffy by Perforce. Available at: https://www.gliffy.com/blog/what-is-uml-everything-you-need-to-know-about-unified-modeling-language.

Lucidchart (2019). *UML Use Case Diagram Tutorial | Lucidchart*. [online] Lucidchart.com. Available at: https://www.lucidchart.com/pages/uml-use-case-diagram [Accessed 20 Oct. 2021].

Martin, M. (2019). *Functional Requirements Vs Non Functional Requirements: Key Differences*. [online] Guru99.com. Available at: https://www.guru99.com/functional-vs-non-functional-requirements.html [Accessed 20 Oct. 2021].

Rome, P. (2020). *What Are Non Functional Requirements - with Examples*. [online] Perforce. Available at: https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples [Accessed 20 Oct. 2021].

Approach – Frameworks and Programming Languages

AltexSoft. (2021). *The Good and the Bad of Flutter App Development*. [online] Available at: https://www.altexsoft.com/blog/engineering/pros-and-cons-of-flutter-app-development/.

Android Developers. (2021). *Android's Kotlin-First Approach*. [online] Available at: https://developer.android.com/kotlin/first.

Apple (United Kingdom). (2021). *Swift - Apple (UK).* [online] Available at: https://www.apple.com/uk/swift/ [Accessed 12 Aug. 2021].

Barron, B. (2021). *What is the Best Programming Language to Learn in 2021?* [online] Kinsta. Available at: https://kinsta.com/blog/best-programming-language-to-learn/#java [Accessed 12 Aug. 2021].

Beladiya, K. (2019). *Top 10 Reasons to Choose Kotlin for Android App Development*. [online] TheOneTechnologies. Available at: https://theonetechnologies.com/blog/post/top-10-reasons-to-choose-kotlin-for-android-app [Accessed 13 Aug. 2021].

Björklund, V. (2021). *What Is Dart and What Is It Used for?* [online] proxify.io. Available at: https://proxify.io/articles/what-is-dart-and-what-is-it-used-for [Accessed 14 Oct. 2021].

App Partner (2017). *10 Reasons Why Agile is Best for App Development & How We Use It*. [online] Medium. Available at: https://medium.com/app-partner-academy/10-reasons-why-agile-is-best-for-app-development-how-we-use-it-36c53bd0cf9c.

Cavet, N. (2021). *4 Advantages and Disadvantages of the Waterfall Model*. [online] appvizer.co.uk. Available at: https://www.appvizer.co.uk/magazine/operations/project-management/advantages-of-waterfall-model [Accessed 14 Oct. 2021].

David, D. (2021). *5 Reasons Why You Should Learn iOS Development in 2021*. [online] Geek Culture. Available at: https://medium.com/geekculture/5-reasons-why-you-should-learn-ios-development-in-2021-c88a431d4246 [Accessed 15 Oct. 2021].

Djurovic, A. (2020). *20+ Astonishing Agile Adoption Statistics for 2020 (Updated).* [online] goremotely. Available at: https://goremotely.net/blog/agile-adoption/ [Accessed 14 Oct. 2021].

eMarketer. (2021). *iPhones are Gaining Market Share in the US, but Losing in the UK*. [online] Available at: https://www.emarketer.com/content/iphones-gaining-us-market-share-losing-uk [Accessed 12 Aug. 2021].

Encyclopaedia Britannica (2021). *Java | Definition & Facts. In: Encyclopædia Britannica*. [online] Available at: https://www.britannica.com/technology/Java-computer-programming-language [Accessed 15 Oct. 2021].

Fuad, N. (2019). *10 Good Reasons to Learn Dart*. [online] Medium. Available at: https://medium.com/hackernoon/10-good-reasons-why-you-should-learn-dart-4b257708a332 [Accessed 15 Oct. 2021].

GeeksforGeeks (2019). *Top 10 Reasons to Learn Java*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/top-10-reasons-to-learn-java/ [Accessed 16 Oct. 2021].

Glasscubes (2019). *Agile vs. Waterfall: Pros & Cons, Use Cases, & More* | Glasscubes. [online] Glasscubes. Available at: https://www.glasscubes.com/agile-vs-waterfall/ [Accessed 14 Oct. 2021].

Heller, M. (2020). *What is Kotlin? The Java Alternative Explained.* [online] InfoWorld. Available at: https://www.infoworld.com/article/3224868/what-is-kotlin-the-java-alternative-explained.html [Accessed 16 Oct. 2021].

Jones, P. (2021). *Why Java is the Best Coding Language for Mobile Development*. [online] iTech Post. Available at: https://www.itechpost.com/articles/105416/20210422/why-java-is-the-best-coding-language-for-mobile-development.htm [Accessed 15 Oct. 2021].

Kienitz, P. (2017). *Pros and Cons of Waterfall Software Development* | DCSL GuideSmiths Ltd. [online] DCSL GuideSmiths. Available at: https://www.dcsl.com/pros-cons-waterfall-software-development/ [Accessed 14 Oct. 2021].

Martin, M. (2019). *What is Waterfall Model in SDLC? Advantages & Disadvantages*. [online] Guru99.com. Available at: https://www.guru99.com/what-is-sdlc-or-waterfall-model.html [Accessed 14 Oct. 2021].

Rehkopf, M. (2019). *Sprints | Atlassian*. [online] Atlassian. Available at: https://www.atlassian.com/agile/scrum/sprints [Accessed 14 Oct. 2021].

Scully, E. (2020). *Kotlin vs Java: Compared and Contrasted*. [online] Career Karma. Available at: https://careerkarma.com/blog/kotlin-vs-java/ [Accessed 13 Aug. 2021].

Sherman, R. (2014). *Project Managemen*t. 1st ed. [online] Business Intelligence Guidebook, Morgan Kaufmann, pp.449–492. Available at: https://www.sciencedirect.com/science/article/pii/B9780124114616000186 [Accessed 14 Oct. 2021].

Shneyveys, K. (2021). *Why Learn Kotlin? | The Kotlin Blog*. [online] JetBrains Blog. Available at: https://blog.jetbrains.com/kotlin/2021/04/why-learn-kotlin/ [Accessed 16 Oct. 2021].

Singh Gill, N. (2020). *Kotlin vs Java: Which is Better for Android App Development?* [online] www.xenonstack.com. Available at: https://www.xenonstack.com/blog/kotlin-andriod [Accessed 12 Aug. 2021].

Stack Overflow. (2020). *Stack Overflow Developer Survey 2020.* [online] Available at: https://insights.stackoverflow.com/survey/2020 [Accessed 13 Aug. 2021].

StatCounter. (2021). *Mobile Operating Systems' Market Share Worldwide from January 2012 to June 2021*. [online] Available at: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/ [Accessed 12 Aug. 2021].

Wintrich, D. (2020). *Java: Everything a Beginner Needs to Know*. [online] www.coursereport.com. Available at: https://www.coursereport.com/blog/what-is-java-programming-used-for [Accessed 16 Oct. 2021].

Approach – Assessment and Quiz types

Carnegie Mellon University (2019). *Formative vs Summative Assessment - Eberly Center - Carnegie Mellon University*. [online] Cmu.edu. Available at: https://www.cmu.edu/teaching/assessment/basics/formative-summative.html.

Lloydsbankinggroup.com. (2019). *The M Word Is Britains Biggest Taboo*. [online] Available at: https://www.lloydsbankinggroup.com/media/press-releases/2019/lloyds-bank/the-m-word-is-britains-biggest-taboo.html [Accessed 13 Sep. 2021].

Taras, M. (2005). ASSESSMENT – SUMMATIVE AND FORMATIVE – SOME THEORETICAL REFLECTIONS. *British Journal of Educational Studies*, [online] 53(4), pp.466–478. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8527.2005.00307.x.

Iannone, P. and Simpson, A. (2016). University students' perceptions of summative assessment: The role of context. *Journal of Further and Higher Education*, 41(6), pp.785–801.

NFER (2018). *An introduction to formative and summative assessment*. [online] NFER. Available at: https://www.nfer.ac.uk/for-schools/free-resources-advice/assessment-hub/introduction-to-assessment/an-introduction-to-formative-and-summative-assessment/.

Weurlander, M., Söderberg, M., Scheja, M., Hult, H. and Wernerson, A. (2012). *Exploring formative assessment as a tool for learning: students' experiences of different methods of formative assessment*. *Assessment & Evaluation in Higher Education*, 37(6), pp.747–760.