Gym Analytics and Personal Training Application

Research Manual

Author: Jamie Hawthorne

Supervisor: Greg Doyle

Submission Date: 30/04/2021

Abstract

The purpose of this “Gym Analytics and Personal Training application” is to develop a mobile application for Android and iOS with two sections available to the public. Firstly, we have a client section to help the average person hire a personal trainer, purchase a workout plan whether it is a pre-made plan or a requested plan. The client also has the option to set motivational personal goals/ milestones for them to achieve.

Secondly, we have the trainer side of this application, it is here where a Personal Trainer can grow their business by posting pre-made plans, create custom plans for clients, reach a wide range of potential clients and receive payments directly into their PayPal accounts.

Contents

Table of Figures

Introduction

My Project is a “Gym analytics and Personal training application” that will work for two types of people. Firstly, we have the average user as a “client” the client side of this application will allow the client to view all trainers registered within the app, select a trainer as their trainer, purchase a plan (payment included), CRUD goals and view their daily workout. The other side of this application will be focused on the “trainer” who will have the ability view all their clients, CRUD workout plans, receive payments, view payments received, receive requests for plans and view client stats/progress.

Goals

- Create application to work on major mobile platforms (Android & iOS).
- Create two main logins for the application (client, trainer).
- Ensure database security (password hashing).
- Verification process for trainers (ensure they are certified).
- Allow Administrator access to data analytics.

The need for this project

The reason I choose to do this project and feel it is a valuable project is made even more evident in the current Global pandemic. The reason is both mental and physical health. With the current pandemic we cannot go to the gym during full lockdown, nor can we meet up with other people to help encourage us or motivate us to workout, therefore it is becoming increasingly hard to stay in shape and feel healthy and happy. With the pandemic it has also become increasingly hard to hire a personal trainer. It is important to keep activate and set ourselves goals during these times as we all know a healthy body means a healthy mind.

This project will also be a massive help to personal trainers as many of them are losing their jobs during this pandemic and are unable to work or, in extreme circumstances, support their families.

Overview of Areas Covered in This Paper

In the following research document, I am going to look at several areas and technologies for potential use in this project such as:

1. Development environments/platforms
2. Programming languages
3. Databases
4. Data Analytics
5. Payment systems

Development Environments/Platforms

The requirements for the project are to launch a web dashboard accompanied by an application available on a mobile device. I have opted to release this application on the two major platforms being Android and iOS. Android and iOS have similar but different architectures which might limit the ability to be able to develop both in the same environment which can cause release and version control issues along with doubling the development time.

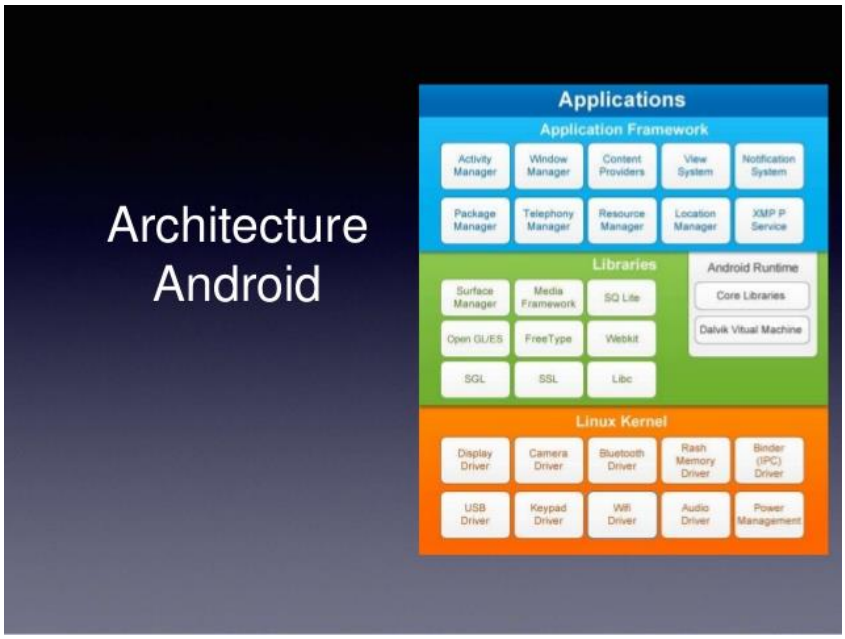

Figure 1. Android Architecture [7]

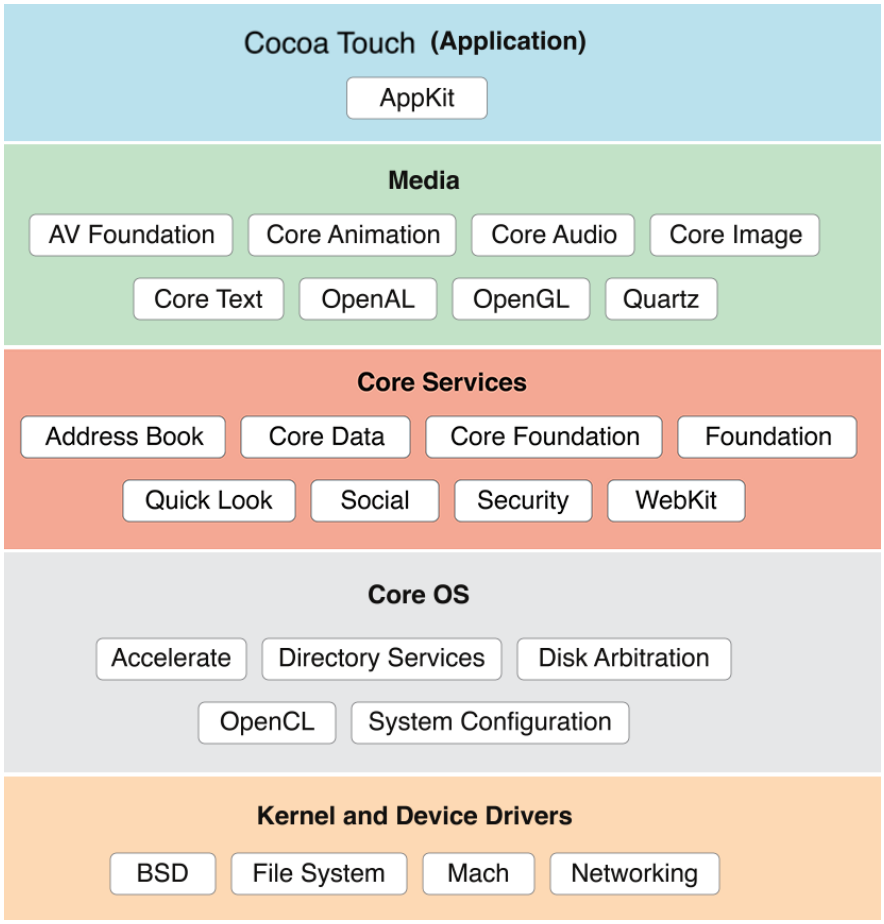

Figure 2. iOS Architecture [8]

To complete this I require a development environment to produce this application. The following environments came into consideration during my research:

- Android Studio
- XCode
- Xamarin

Below I have given a brief overview of each environment, what they offer and my conclusion and chosen environment based on my findings.

Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development. It is based on IntelliJ IDEA and on top of IntelliJ's powerful code editor and developer tools, Android Studio even offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices.
- Apply Changes to push code and resource changes to your running app without restarting your app.
- Code templates and GitHub integration to help you build common app features and import sample code.
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems.
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine [1]

Throughout the duration of this project version control will be important. Having the ability to roll back to previous versions or update to newer versions throughout is a must with this and most projects. Android Studio supports a variety of version control systems (VCS's), including Git, GitHub, CVS, Mercurial, Subversion, and Google Cloud Source Repositories.

Throughout my research I continued to look for the main selling point for each environment, what made it stand out above the rest and would make it ideal for my project.

"Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug." [1]

In conclusion android studio would be the ideal environment ticking all the boxes required for this development if I were to only launch onto android. However, I am also planning on

releasing to iOS Apple store which the code cannot be developed within Android studio so I would require 2 different environments which would cause issues with version control.

XCode

XCode is an IDE for MacOS, iOS, watchOS and tvOS platforms developed by Apple. XCode includes most of the developer documentation from Apple and Interface Builder - an application used to create graphical interfaces.

As an entire suite, XCode can be used for designing the user interface, writing the code for apps, compiling the code, testing the code and app, and debugging. On completing an app to a quality where it can be accepted for distribution, XCode is also used to submit the app to Apple's assorted App Store marketplaces.

Some of the XCode features offered are:

- SwiftUI
- Minimap
- Rich Documentation
- Swift Package Manager
- Inline Diff
- Transform iPad Apps to Mac.
- Debugging and Simulators [2]

The main selling point that puts XCode ahead of some of the other environments is the power of SwiftUI.

Apple strongly promotes Swift as its language of choice across all its platforms. A key element of this is that SwiftUI apps are native on all Apple-owned platforms. The core logic of a SwiftUI app for an iPhone can easily be transplanted across to a macOS app and vice versa, with platform-specific elements able to be accounted for in each version.

This is most apparent in one feature where ticking a single checkbox can enable XCode to convert an iPad app into a native Mac app, one that can be used with a keyboard and mouse rather than a touchscreen. The intent is to simplify the porting process of iPad apps for macOS, though developers have found the option does still require further refinement of the interface for iPad apps to be truly useful on macOS.

SwiftUI uses a declarative syntax, where the developer states what the user interface should do, such as listing items in a text field then defining the formatting of each field. This also extends to animations and other areas, with SwiftUI handling most of the work to get the code running for many different built-in effects.

In conclusion XCode would also be the ideal environment ticking all the boxes required for this development if I were to only launch onto iOS.However, since I am also planning on releasing to Android app store this two would require an additional environment causing the same issues as Android Studio.

Xamarin

Xamarin is an open-source platform for building modern and performant applications for iOS, Android, and Windows with .NET. Xamarin is an abstraction layer that manages communication of shared code with underlying platform code.

Xamarin enables developers to share an average of 90% of their application across platforms. This pattern allows developers to write all their business logic in a single language (or reuse existing application code) but achieve native performance, look, and feel on each platform. [3]

So, I asked myself why use Xamarin and the answers just blew me away with how powerful this platform really is. Below you can see the diagram on how Xamarin works.

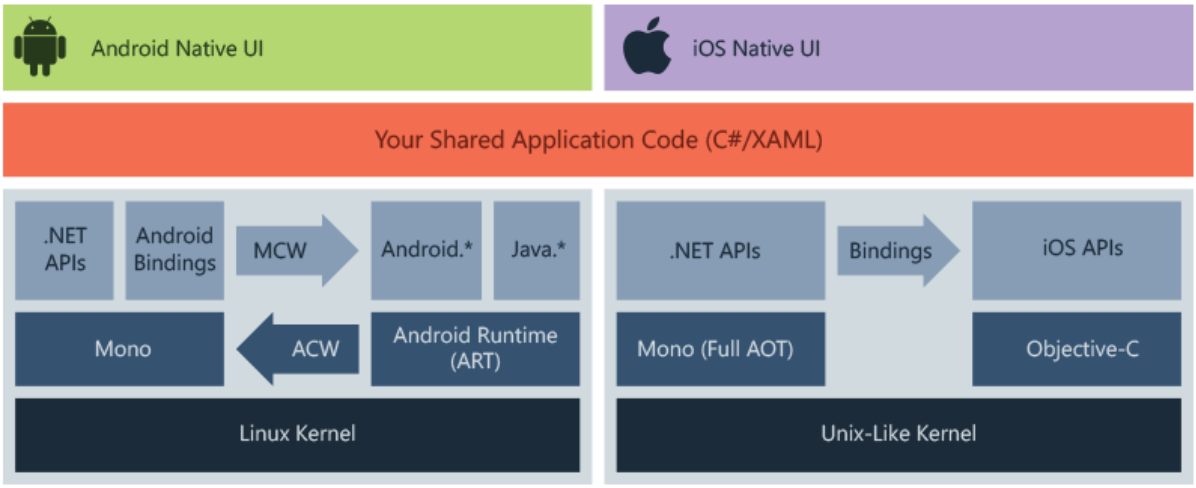

Figure 3. How Xamarin works [3]

“The diagram shows the overall architecture of a cross-platform Xamarin application. Xamarin allows you to create native UI on each platform and write business logic in C# that is shared across platforms. In most cases, 80% of application code is sharable using Xamarin.

Xamarin is built on top of .NET, which automatically handles tasks such as memory allocation, garbage collection and interoperability with underlying platforms.” [3]

Xamarin being so widely used in the industry means there is also so much documentation on how to do everything which will cut the learning time in half allowing the development process to be completed smoother and faster. As Xamarin is a platform not an environment the actual IDE I would be using will be Microsoft Visual Studio. This works amazingly with Xamarin as any addon required comes in the form of a NuGet package which can be browsed and installed instantly.

In conclusion Xamarin seems to be the perfect Environment to begin developing my Gym Analytics and Personal Training application as it includes role back abilities, the ability to code for both platforms in one environment without the use of addons. This will greatly help with version control and the release.

Conclusion

Based on my research into the above development environments I have decided to begin development with the Xamarin environment as it seems to be the best fit for my project allowing both applications being developed side by side. That combined with the amount of information publicly available I feel it is the best fit for taking on this project.

Programming Language

To develop this application and accompanying dashboard I am required to code the functionalities and aesthetics. This will be done in a two-step process as I am essentially building two things here:

1. Application for mobile device
2. Dashboard

Both tasks will require different approaches and different programming languages. Below I have given a brief overview of what languages I will use and a brief description of each.

Language Used

Now that I have decided on an environment to develop my application in, I can move onto the next step which is to decide on a programming language. This decision would usually take hours or days comparing different coding languages as all have their merits however Xamarin applications are written in C# and XAML, so the decision has been made for me in regards the application side. This is ideal for me as I have previously spent time working with C# so the only syntax learning will be for XAML.

"Modern language constructs – Xamarin applications are written in C#, a modern language that includes significant improvements over Objective-C and Java such as dynamic language features, functional constructs such as lambdas, LINQ, parallel programming, generics, and more." [3]

Next, I must look at the administrator's dashboard for the application. Initially I had plan to develop this dashboard using a mix of HTML, CSS, and JavaScript. However, during the development of my database I discovered a built-in dashboard in Firebase Analytics which allows the administrator full privileges and is built to a standard that would require a team of developers to develop so I figured why not take advantage of this.

C#

"C# is a strongly typed object-oriented programming language. C# is open source, simple, modern, flexible, and versatile." [4]

C# is a programming language that was developed and launched by Microsoft in 2001. C# is a simple, modern, and object-oriented language that provides modern day developers flexibility and features to build robust software.

Key characteristics of C# language include: [4]

1. Modern and easy
2. Fast and open source
3. Cross platform
4. Safe
5. Versatile
6. Evolving

XAML

XAML stands for Extensible Application Markup Language and is a decretive markup language developed by Microsoft to be applied to the .NET Core programming model. The purpose of XAML is to simplify the creating of a User Interface (UI) for all .NET Core apps.

XAML and C# will work together by separating the backend logical code from the front-end UI "You can create visible UI elements in the declarative XAML markup, and then separate the UI definition from the run-time logic by using code-behind files that are joined to the markup through partial class definitions." [13]

HTML

HTML stands for Hypertext Markup Language. It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications.

HTML is not a programming language, meaning it does not have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.

When working with HTML, we use simple code structures (tags and attributes) to mark up a website page. For example, we can create a paragraph by placing the enclosed text within a starting and closing tag.

CSS

"CSS stands for Cascading Style Sheets with an emphasis placed on "Style." While HTML is used to structure a web document (defining things like headlines and paragraphs, and allowing you to embed images, video, and other media), CSS comes through and specifies your document's style—page layouts, colours, and fonts are all determined with CSS. Think of HTML as the foundation (every house has one), and CSS as the aesthetic choices (there's a big difference between a Victorian mansion and a mid-century modern home)." [5]

JavaScript

“JavaScript is a scripting language used to create and control dynamic website content, i.e., anything that moves, refreshes, or otherwise changes on your screen without requiring you to manually reload a web page. Features like:

1. animated graphics.
2. photo slideshows.
3. autocomplete text suggestions.
4. interactive forms. “[6]

Firebase Analytics

The Firebase Analytics dashboard will give the administrator full access to a wide array of data, the ability to fully customise the returned data, view versions released and currently in use and it also provides a high level of security as only the database and/or app administrator can gain access. This not only cuts development time down and allows me more time to focus on the mobile application but it allows for the data to also be downloaded in the form of a csv file so in future development if the application administrator required a specialised dashboard the data from Firebase analytics could be transferred.

“Analytics surfaces data about user behaviour in your iOS and Android apps, enabling you to make better decisions about your product and marketing optimization. View crash data, notification effectiveness, deep-link performance, in-app purchase data, and more.” [12]

Conclusion

In conclusion I will be using C# and XAML for the development of the application side of this project as it is the language used by Xamarin. I will also be then using the prebuilt firebase dashboard equip with platform breakdown, user retention data, version adaption and much more.

Database

In Order to complete this project and allow it to sync data to allow user to use not only mobile application but also the dashboard and to allow the trainer to be able to see the client's data I will require a back-end database. Throughout my research I found the three best database I feel will be best suited for this project and they are as follows:

- MS SQL Server
- MySQL
- Firebase

Below I have given a brief overview of each and have come to my conclusion based on my findings.

MS SQL Server

MS SQL is a Microsoft based SQL server, its main features that lead it to be considered are:

- Ensures Security of Your Data
- Optimizes Data Storage and Accuracy
- Simple Installation and Automatic Updates
- Optimized Security Features
- Simple Management and Maintenance

Having only little previous work with database I had to give a lot of thought to which one I would use, and I wanted one that is powerful but also easy to use as a beginner with a lot of documentation. MS SQL seems to tick all these boxes.

"Between the various database management software options available, Microsoft SQL Server is one of the most powerful ones to date. However, there are many web developers and web administrators who are still clueless about just how powerful this software is. They are missing out on the great assets that it can provide in making a website's database management more transparent and efficient." [9]

In conclusion MS SQL Server is in the top running for this project as its simple, has lots of publicly available documentation and powerful enough for what I need for this project.

MySQL

MySQL is one of the most popular, powerful and reliable database servers to date.

"The latest version of MySQL is one of the world's most popular databases. It is open source, reliable, compatible with all major hosting providers, cost-effective, and easy to manage. Many organizations are leveraging the data security and strong transactional support offered by MySQL to secure online transactions and enhance customer interactions." [10]

MySQL being so massive means there is so much documentation out there and being open source means any modifications needed to tailor it to my project can be done easily. With there being so much out there on MySQL I looked for the main reasons this would suite my project and its main selling points overall. Below I have included the top five reasons why this would suit my application:

- Secure Money Transactions
    - "MySQL transactions work as a single unit, which means unless and until every individual operational stage is successfully completed, the transaction is not cleared." [10]
- On-Demand Scalability / Customization
    - Provides ultimate platform scalability.
    - Completely customizable
    - Unmatched flexibility in regards scaling
- High Availability
    - Consistent availability is the stalwart feature of MySQL – enterprises that deploy it can enjoy round-the-clock uptime.
    - MySQL comes with a wide variety of cluster servers and master-slave replication configurations that enable instant failover for uninterrupted access.
- Rock-Solid Reliability
    - Powerful data encryption prevents unauthorized viewing of data and SSH and SSL supports ensure safer connections. [10]
- Quick-Start Capability
    - You can go from downloaded to installed in under 15 minutes, simple design and vast documentation means you can get started instantly.

In conclusion MySQL would be a perfectly suitable database for this project as its the one I have the most experience with. Also with its simplistic layout and amount of information available I see the high success for this project using it.

Firebase Realtime Database

Firebase is an extremely powerful and secure BaaS (Backend as a Software).[1] Firebase's goal is to provide developers with all the tools and services needed to securely develop high-quality apps.[14] in relation to databases, Firebase has what they call its Realtime Database which is an in cloud-hosted NoSQL database that lets the users store and sync data in Realtime. [15] There are many reasons why Firebase's Realtime Database would be an ideal solution for this project, some of which are listed below: [15]

- The Realtime Database works using SDK's which removes the need of servers.
- The Realtime Database is also optimized so when a user goes offline local cache is used until connection is re-established and the local data is then synchronized.
- The Realtime Database integrates with Firebase Authentication to provide simple and intuitive authentication for developers.

In conclusion Firebase Realtime Database is a perfect fit for this applications database. Although I have had no prior experience with its Firebase Realtime Database does seem like the obvious choice and the one that will achieve the highest results.

Conclusion

Based on my findings throughout my research into databases I feel any of these database providers would serve as sufficient for this project. I have decided to go with Firebase Realtime Database as I feel it stands out from the rest, is more optimized to suit the needs of this project and will work best with the Firebase tools I already plan to implement.

Data Analytics

Data Analytics

Data analytics is the science of analysing raw data to make conclusions and form information. Many of the techniques and processes of data analytics have improved and since been automated into mechanical processes and algorithms that work with raw data and turn it into information for human consumption.[11]

Firebase Analytics

For this application as mentioned in previous sections I will be using Firebase Analytics, I have opted to use Firebase Analytics as it will take in all the raw information for this application and display it back to the administrator in the form of graphs and readable information integrated into the dashboard.

“Analytics surfaces data about user behaviour in your iOS and Android apps, enabling you to make better decisions about your product and marketing optimization. View crash data, notification effectiveness, deep-link performance, in-app purchase data, and more.” [12]

Firebase Analytics comes with several abilities some of which I have listed below:

- Dashboard which displays an overview of all data tracked within the application to the administrator.
- Realtime which can give you a sense of where in the world people are using your app, right down to the city level.
- Events which display all events programmed into the application from the app being removed to the amount of people signing up.
- Latest Release which displays the current releases and how many users are on each version.

This is only the surface level of what Firebase Analytics can do which is why I have chosen to incorporate it within my application.

Conclusion

In conclusion I feel Firebase Analytics will be a great addition to this application as it will help the administrator of this application better understand the applications users and guide them on the best way to manage the application.

Payment Systems

In Order to complete this project and for this application to work I require a way for the Trainers to receive money from Clients therefore I need a checkout system that will integrate with Xamarin. Many payment systems came to mind instantly such as

- Apple Pay
- Google Pay
- PayPal
- Stripe

However, both apple and android pay are limited to the devices that the application is installed on. To make this application fully cross platform that left my two main options to be Stripe and PayPal.

PayPal

PayPal is one of the most secure ways to transfer money as it allows you to do so by only revealing your email address "PayPal is a service that enables you to pay, send money, and accept payments without revealing your financial details." [16] This being said, PayPal's Checkout and Payout SDK's do not allow for an application to make payments from one given account to another without the money being transferred to a set business account for that application. This could be solved by allowing the money to be transferred from the clients PayPal account to the application's business account and then from there into the trainers account. However, this causes a new issue as the client must now also have a PayPal account as the Checkout and Payout SDK do not support guest card payments at this time in Xamarin, which leads us to Stripe.

Stripe

Stripe is a payment processing platform that allows a business to handle payments using its API's. These APIs include:

- Once off payments
- Subscriptions
- Credit card payments
- Account transfer payments
- Bank transfer payments

And many more. The only limitation with stripe's payment system I have found is it will require the trainers to have a Stripe business account to receive payments from the application. The way I plan on solving this is by integrating both payment systems for their individual strong points.

Conclusion

In conclusion this application requires a payment system to function properly. I have decided to incorporate both payment systems to achieve the best outcome within my application. I feel Stripe is suited best for handling the money received by clients and PayPal to send money from the Application's business account to the trainer's account once the payment has been received from the client. This would also allow for if the application were to require a commission fee in its later business plan.

Bibliography:

1. Android Developers. 2020. *Meet Android Studio | Android Developers*. [online] Available at: <https://developer.android.com/studio/intro>.

2. JournalDev. 2020. *XCode 11 Features - Journaldev*. [online] Available at: <https://www.journaldev.com/30976/xcode-11-features>.

3. Docs.microsoft.com. 2020. *What Is Xamarin? - Xamarin*. [online] Available at: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin#:~:text=Modern%20language%20constructs%20%E2%80%93%20Xamarin%20applications,programming%2C%20generics%2C%20and%20more.>.

4. Chand, M., 2020. *What Is C#*. [online] C-sharpcorner.com. Available at: <https://www.c-sharpcorner.com/article/what-is-c-sharp/>.

5. Morris, S., 2020. *What Is CSS, How Does It Work And What Is It Used For? - Skillcrush*. [online] Skillcrush. Available at: <https://skillcrush.com/blog/css/>.

6. Morris, S., 2020. *What Is Javascript? A Guide For Total Beginners - Skillcrush*. [online] Skillcrush. Available at: <https://skillcrush.com/blog/javascript/>.

7. RIMTIC, H., 2014. Architecture iOS Et Android. [online] Slideshare.net. Available at: <https://www.slideshare.net/HadinaRIMTIC/pres-1-42958371>.

8. Chauhan, S., 2018. *Understanding Xamarin iOS - Build Native iOS App*. [online] Dotnettricks.com. Available at: <https://www.dotnettricks.com/learn/xamarin/understanding-xamarin-ios-build-native-ios-app>.

9. Certification Camps. 2020. Why Choose Microsoft SQL Server For Your Website's Database Management? [online] Available at: <https://www.certificationcamps.com/why-choose-microsoft-sql-server/#:~:text=As%20a%20basic%20definition%2C%20Microsoft,analytics%20in%20corporate%20IT%20environments.>.

10. BRANSON, T., 2017. The 5 Best Reasons To Choose MySQL - And Its 5 Biggest Challenges - Dataconomy. [online] Dataconomy. Available at: <https://dataconomy.com/2017/04/5-reasons-challenges-mysql/#:~:text=It%20is%20open%20source%2C%20reliable,transactions%20and%20enhance%20customer%20interactions.>.

11. Frankenfield, J., 2020. How Data Analytics Work. [online] Investopedia. Available at: <https://www.investopedia.com/terms/d/data-analytics.asp>.

12. Firebase. n.d. *Google Analytics for Firebase | Free and unlimited app analytics*. [online] Available at: <https://firebase.google.com/products/analytics>.

13. Docs.microsoft.com. 2021. *XAML overview - WPF .NET*. [online] Available at: https://docs.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-5.0.

14. Educative: Interactive Courses for Software Developers. n.d. *What is Firebase?* [online] Available at: <https://www.educative.io/edpresso/what-is-firebase>.

15. Firebase. n.d. *Firebase Realtime Database | Store and sync data in real time*. [online] Available at: https://firebase.google.com/products/realtime-database?gclsrc=aw.ds&&gclid=CjwKCAjwj6SEBhAOEiwAvFRuKEE_B4_buFSzAKUCktpok68W4VTE888rb_jzz75eMSsdSu4iNHUbahoC2akQAvD_BwE.

16. Paypal.com. n.d. *What is PayPal? Learn more about PayPal's unique features*. [online] Available at: https://www.paypal.com/ie/webapps/mpp/paypal-popup.

Declaration

- I declare that all material in this submission, e.g., thesis/essay/project/assignment, is entirely my own work except where duly acknowledged.

- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams, or other material; including software and other electronic media in which intellectual property rights may reside.

- I have provided a complete bibliography of all works and sources used in the preparation of this submission.

- I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offense.

Student Name: Jamie Hawthorne

Student Number: C00226160

Student Signature: 

Date: 30/04/2021