

BOOSTER - ARCHERY PERFORMANCE TRACKING APPLICATION

RESEARCH REPORT



Supervisor: Dr. Joseph Kehoe

Author: Brendan Browne

Submission Date: 01 November 2019

Abstract

The purpose of this project is to create a cross platform application for archery coaches and archers alike. It should help perfect form, track performance, scores, grouping and give suggestions on breathing, strength and conditioning techniques. The application should be used as a tool to help enhance the archery performance and be an easy to use, user friendly experience. It should be able to keep a record of scores, grouping, and heart rate of the archery to set goals that, based on previous performance, are attainable with some extra training.

Table of Contents

Abstract	1
Table of Contents	2
Table of Figures	3
1. Introduction	4
2. Overview	5
2.1 Archery	5
2.1.1 What is Archery?	5
2.1.2 Measurable Key Performance Indicators in Archery.	5
2.2 Similar Applications	7
2.2.1 Video Analysis Applications	7
2.2.2 Score Tracking and analysis	7
2.3 Android vs iOS	8
2.4 Cross Platform	10
2.5 Conclusion	11
3. Technology	12
3.1. Wearables	12
3.2 Movesense HR+ €39.50	13
3.3 Polar €149.90 - €599.90	13
3.4 Fitbit €69.95 - €299.95	14
3.5 Conclusion	14
4. Software	15
4.1 Front End Technologies	15
4.1.1 Xamarin	15
4.1.2 IONIC	15
4.1.3 Kivy	16
4.1.4 Conclusion	17
4.2 Back End Technologies	18
4.2.1 Cloud	18
4.2.2 Local	18
4.2.3 Conclusion	18
5. Conclusion	Error! Bookmark not defined.
Appendix	19
i. Bibliography	19
ii. Plagiarism Declaration	21

Table of Figures

Figure 1. Power line No.1 [Copa19]

Figure 2. Power line No.2 [Copa19]

Figure 3. Worldwide Smartphone Shipment OS Market Share Forecast [IDC19]

Figure 4. Worldwide Gross App Revenue - Q3 2019 [IDC19]

Figure 5. Adoption rate of Android OS Q3 2019 [Mixp19]

Figure 6. Adoption rate of iOS OS Q3 2019 [Mixp19]

Figure 7. Smartwatch sales in the US 2013 - 2018 [Liu19]

Figure 8. Xamarin architecture [Docs19]

Figure 9. IONIC Architecture [Qian16]

1. Introduction

The following Research Report is intended to document all research partaken for the development of the BOOSTER application. The overall understanding of the project, requirements and general idea behind the project should be expanded in conjunction with this report.

The following document will delve into the subject of archery and describe the various performance metrics that are quantifiable and of use to this project. Some performance metrics are not quantifiable, like mental comfort, but should still be considered of some importance to the application as they do assist in the mental side of the sport. Similar applications available on the market will be researched to gain an understanding of what features are most sought after and what pitfalls to avoid.

The subject of Android vs iOS vs cross platform applications will be approached to determine which platform would be best suited for the application. The benefits and dangers of each will be investigated to gain a better understanding and give the ability to make an informed decision. Various wearable technologies will be investigated to determine which could give the required data according to the metrics required.

Frontend and Backend software will be researched to allow the efficient and swift development of a functional application. Storage and Deployment methods will be a part of this and the choice of local or cloud storage, single or multiuser as well as various other decisions must be made to determine the ultimate success of the project.

2. Overview

2.1 Archery

2.1.1 What is Archery?

Archery is an ancient sport that has been around for thousands of years in one form or another. It is the sport, skill or practice of shooting a bow and arrow. Historically archery has been used for combat or hunting, “*The earliest people known to have regularly used bows and arrows were the Ancient Egyptians, who adopted archery around 3,000BC for hunting and warfare*” [Wor19].

In modern times, archery is mainly practised as a sport around the world, with some countries still allowing bow hunting to be practised. Although many variations of the sport and variations of bow style exist, e.g. Olympic style recurve, Japanese Yumi bow, Turkish horse bow, English longbow, compound bow, etc..., the underlying form that an archer practises is similar enough across all styles that one could become proficient in a single style and pick up another style with relative ease.

2.1.2 Measurable Key Performance Indicators in Archery.

Key Performance Indicators (**KPI**) are commonly used to set targets or some numerical value to achieve. In archery however, KPI's may also be used to measure consistency in form, e.g. angle of the draw arm elbow, power line alignment, stance consistency, grouping on the target, heart rate, breathing technique, to name a few.

There are two main power lines associated with archery, the first is a horizontal line drawn from the elbow of the draw arm through the nock of the arrow to the arrow tip. An example of this can be seen in Figure 1. The second is a vertical line, as viewed from above, drawn through the same points, as can be seen in Figure 2. Once an archer has perfected consistently aligning these power lines, they should be able to achieve a smaller group with their arrows. This, of course, is also dependent on consistent basic form, e.g. consistent anchor point, consistent release and follow through, consistent stance.



Figure 1. Power line No.1 [Copa19]



Figure 2. Power line No.2 [Copa19]

Heart rate also factors into the archer's ability to control their bow. If an archer has a high heart rate, this may indicate one of two issues.

1. The archer is unfit, which may lead to fatigue, instability and, in the worst case, injury.
2. The archer has lost their mental focus, this may be attributed to nervousness in a competition scenario, self-doubt or simply an inability to focus on a given day.

Breathing techniques are taught to archers once they reach intermediate / advanced levels to mitigate the effect of a high heart rate on the archer. Strength and conditioning exercises are also taught to assist in building the stamina of an archer and to develop the specific muscles used in archery. This in turn, in theory, will reduce the resting heart rate of the archer and allow them to develop a more stable shooting platform.

2.2 Similar Applications

2.2.1 Video Analysis Applications

2.2.1.1 Coaches Eye - €120 Individual / €240 Teams package

Coaches eye allows the user to capture video on any mobile device, be it phone, tablet, go pro style camera or DSLR style camera, and upload the video via the web to review on another device which can support the application. It is cross platform and available on the Google Play Store, Windows Store and iOS Apps Store. Playback video in slow motion or frame by frame to allow detailed analysis and precise feedback. Side-by-side video analysis is possible to compare and contrast movements. It enables the user to draw on the video using lines, shapes and arrows to clearly illustrate movements. Coaches can record audio notes for the athlete and the application provides a database for uploading video analysis to share with athletes.

2.2.1.2 Hudl Technique - Free

Hudl Technique is another video analysis application similar to Coaches Eye. It too allows the user to record video, playback in slow motion / frame by frame, side by side video comparison and also allows the user to draw directly on the screen. Comments, angles and voiceovers are also available on Hudl. The main difference with Hudl is that the video must be taken on either a mobile phone or tablet, cannot import from other devices and is not supported by Windows. It does have the advantage of being a free open source application which is community driven and ever changing. This does provide the opportunity of customising the application, with the relevant knowledge and skills.

2.2.2 Score Tracking and analysis

2.2.2.1 Archery Score Pad - Android €3.39, Not available on iOS

Archery Score Pad allows the user to track their score over an array of multiple predefined archery rounds. The user can set distance, target face size, record custom round details, score up to 4 users concurrently, and a host of other features.

The user can input score in one of two ways, either using a standard score sheet to record the value of the arrow or with an interactive target face selecting where the arrow hit the target. The latter also ties in to showing grouping of arrows on the target. Running total score, best round and personal best scores can all be stored. Export all scores via email in .csv or .xls formats. Export full configuration to additional media storage, SD Card or cloud storage, when changing devices. Whilst this may sound ideal, the application has experienced considerable issues with exporting files, saving scores and application resets resulting in data loss, over the last year resulting in bad reviews and a general weariness of using the application

2.2.2.2 My Targets - Android Free, Not available on iOS

MyTargets is a free open source application for tracking and analysing score-based performance in archery. It is by far the easiest to use archery application of the countless applications the author has used. With the ability to record as much or as little details about the equipment being used, customise rounds, customise number of arrows per round, choose from a selection of over 40 predefined rounds and easily understood analysis options. It also has the option to record score in the ways described previously with the Archery Score Pad application. Where MyTargets differs from its rivals is in its ability to track weather conditions like wind speed and direction on a given shooting session. Its simplistic view easily navigated layout and plethora of features make it one of the best scores and performance tracking applications for archery on the market at the minute.

2.3 Android vs iOS

With a 100% market share in 2018, according to [IDC19], of all smartphone operating systems (OS), using either Google's Android or Apple's iOS, the question "whether to develop in Android or iOS" has been at the forefront of developers minds for years now. This being said, the likelihood of many, if any, of your applications used on a day-to-day basis being cross platform is quite low. Both platforms have their advantages and disadvantages over the other, as such, attempting a direct comparison can become quite convoluted.

Android, being open source and community/developer driven allows for the selection of applications to fulfil your needs to be quite a bit larger than iOS. This can also lead to issues such as security, trustworthiness, performance as android applications do not have to pass such a stringent vetting process as iOS applications.

iOS's strict vetting process does allow for more secure applications to be distributed, yet also acts as a barrier to developers wanting to develop in iOS as they do not want to take the time to refine their work pre-release to adhere to said vetting.

Year	2017	2018	2019	2020	2021	2022	2023
Android	85.1%	85.1%	86.7%	86.6%	86.9%	87.0%	87.1%
iOS	14.7%	14.9%	13.3%	13.4%	13.1%	13.0%	12.9%
Others	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
TOTAL	100.0%						

Figure 3. Worldwide Smartphone Shipment OS Market Share Forecast [IDC19]

Even dominating the market split as Android is, as displayed in Figure 3 above, the gross profit fell to iOS in terms of Revenue income from application downloads.

There may be many contributing factors in this outcome. As can be seen in Figure 4, the app store received nearly twice the revenue of android in application downloads in Q3 of 2019, with the average growth of both platforms being ~22.9% for that quarter. Many android applications are free due to the open source nature of the platform. Generally android applications come at a reduced cost to the user compared to iOS.

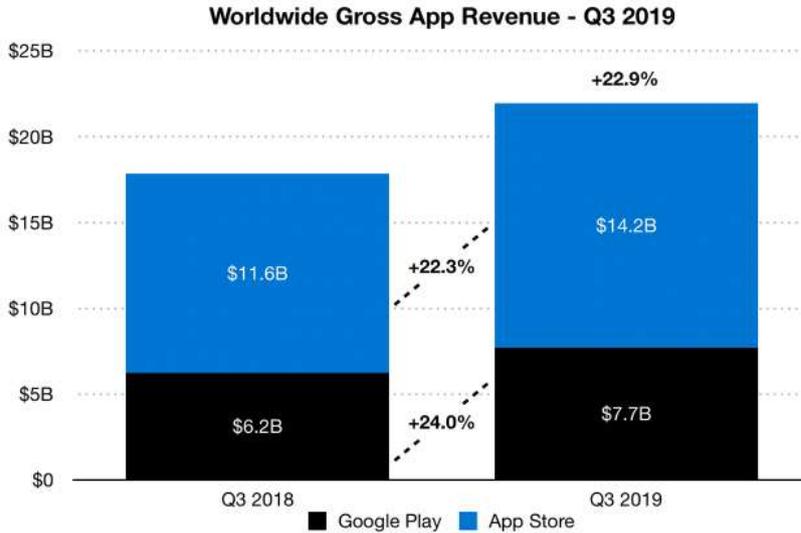


Figure 4. Worldwide Gross App Revenue - Q3 2019 [IDC19]

Code complexity and variation in available devices also contribute to the domination of revenue that iOS displays. With Android written in the widely used JAVA language, the inherent complexity of this language makes development more difficult, yet, the open source nature of Android allows the use of various third-party development tools, libraries and features. This in turn, somewhat, allows for the simplification of development via an extensive toolset. However, the wide range of Android devices combined with the various Android OS makes consistent development more difficult.

To successfully develop an application in Android, one must take into consideration not only varying OS but also hardware concerns such as screen size and resolution, available main-memory, processor specifications, etc... In figure 5 the spread of various OS available for android and their adoption rate for Q3 of 2018 can be seen.



Figure 5. Adoption rate of Android OS Q3 2019 [Mixp19]

iOS on the other hand, written in Swift, Apple’s version of Objective C, requires the use of an Apple Mac computer and has a limited toolset from which to call from. This, along with the lack of third-party development tools, somewhat restricts development in iOS. The limited variation in iOS devices, however, combined with the regularity of the platforms update schedule, results in a development environment with less unknowns. Figure 6 shows the adoption rate of iOS in Q3 of 2019.

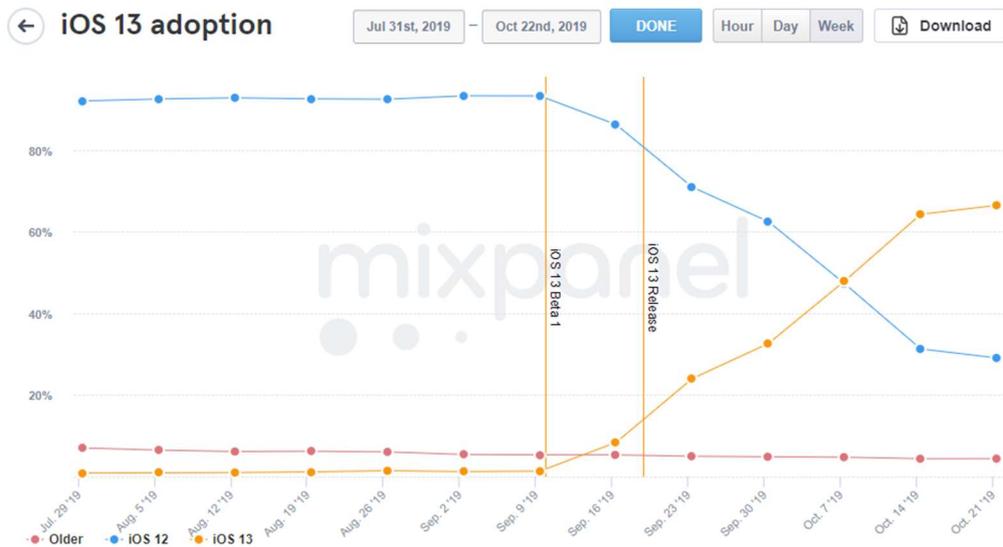


Figure 6. Adoption rate of Apple iOS Q3 2019 [Mixp19]

Application publication on these platforms differs in multiple ways. With a one-time payment for Android developers to publish on the Google Play Store and no validation of the application this process is much faster than that of iOS.

On the Apple Store, extensive validation, a relatively long waiting time and a yearly publication fee makes the process longer and harder which can ultimately deter developers from utilising this platform. “For Android apps, developer fees can range from free up to matching the Apple App Store fee of \$99/year. Google Play has a one-time fee of \$25” [Mack19].

2.4 Cross Platform

Another option for development, rather than restricting the application to a single platform, is to develop a cross-platform application that may be utilised by multiple platforms. The concept of “code once, deploy twice” is a recurring theme in cross platform development. “Code once, Deploy twice” refers to the idea of developing an application, in a single language that will work across multiple platforms, with potentially some minor platform dependant adjustment and customisation required, then to deploy this application across multiple platforms, thus maximising the potential target market. While this will mean additional development steps are undertaken, the potential gains outweigh the additional work required.

2.5 Conclusion

In conclusion with the range of applications available on the market, none seem to address the form of the archery themselves without having to record video multiple times and then require the coach and archery to stop to analyse. A real time video stream with overlay of power lines would assist both coaches and archers to correctly and quickly address form issues. The score system as seen in the MyTargets application is by far the best on the market and whilst the level of customisation can be quite useful, a more simplistic version of this would encourage people new to the sport to use the application while still providing the range of analysis needed to improve performance. Most applications available also do not offer strength and conditioning or breathing technique suggestions to assist the archer in the physical aspect of the sport. Potentially a mental comfort aspect to the application would assist in the mental side of the sport, allowing the archer to also track mood and effort put in on a given training session. With the majority being single platform applications a cross platform application to cater for the majority of archer's devices should be made available. Whilst this may prove to add a small amount of effort on the part of the developer, in the short term, a multi-platform application developed for ease of use and accessibility to future developers will prove beneficial.

3. Technology

3.1. Wearables

With the proliferation of wearable technologies over the last decade, in healthcare, sport and consumer electronics, it is easy to predict that the majority of mobile or electronic devices that we carry with us on a daily basis will be evolved into a wearable form. There are three main wearable categories that can be identified, health related, textile based and consumer electronic wearables [Cice15].

Smartwatches have been available on the market for the last 30 years in one form or another but with the release of the apple watch in 2015 the market exploded in popularity. As can be seen in Figure 7, the sales of smartwatches in the United States (US) alone has increased 2450% over a mere 6-year period. This alone shows the importance of wearable technology on today's society, not including the advancements made to the healthcare and sports industries.

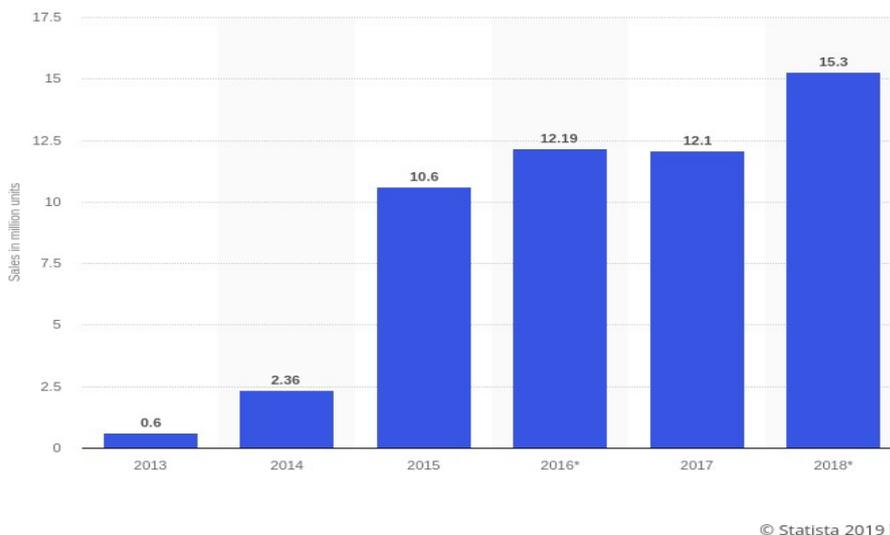


Figure 7. Smartwatch sales in the US 2013 - 2018 [Liu19]

“Self-monitoring has been identified as an effective behaviour change technique that has been used in behavioural interventions targeting increases in physical activity levels” [Mich13].

With the above in mind, the ability to monitor one's vitals, track and record this data and visualise the progress being made will enable the wearer to motivate themselves to push harder, promoting not only physical wellness but also reinforce mental health.

3.2 Movesense HR+ €39.50

The Movesense HR+ is a device that allows the user to track heartbeat, Body temperature and motion. It incorporates 3 axis accelerometer, 3 axis gyroscope and a 3-axis magnetometer to detect shifts in motion. Running on an open Movesense API and using the C++ language it allows developers to create custom in-device applications for the measurement and analysis of all motion types. Swim and shock proof with in-built memory of 3 Mb, it avails of Bluetooth Low Energy (BLE) connectivity to grant access to sensor telemetry and stored data. BLE also allows for wireless firmware updates. The devices in-built memory allows for standalone data logging which can be easily retrieved once connected through BLE. Using a CR2025 cell battery, depending on frequency and duration of use as well as sensor usage, this could potentially provide for months of usage [Lofb19]. Initial pairing of the device through Bluetooth can, at times, be a frustrating experience. Data retrieval may be accomplished via several methods, however the most widely used method is to save the data in a .csv file and transfer that through Bluetooth connection to another device the Movesense is paired with. This negates the need for code compatibility between the application and Movesense, enabling the application to be developed for any platform and in any language the developer would deem suitable.

3.3 Polar €149.90 - €599.90

The POLAR Fitness tracker is a range of devices that closely resemble smart watches / Fitbit wearables. Depending on the model chosen they can provide, but not limited to, a combination of the following features.

- Fitness tracking
- Heart rate monitoring
- Global Positioning System (GPS)
- Smart Calories
- Guided breathing exercises
- Swimming Metrics
- Distance
- Step counter
- Phone notifications

POLAR devices utilise the same BLE as Movesense for connectivity to other devices. With a battery life of up to 40 hours on the top of the range model provided by an in-built 320 mAh Lithium-polymer (Li-pol) battery, the range of use for this device is varied. The devices are compatible with Mac OS X 10.9+, Windows 7+ [Pola19], POLARs own web service and mobile application via Bluetooth Smart. Data can be manually exported to third party applications as .tcx, .gpx or .csv file formats [Pola19]. As POLAR devices are under license there is little in the way of documentation available to the general public on how it works. Accessibility to the device is limited to unidirectional data flow with third party applications.

This, in one respect, limits the options available to the developer, that being said, the range of data available to the developer is potentially so vast due to the sheer amount of functionality provided by the POLAR devices that this could potentially negate the issue.

3.4 Fitbit €69.95 - €299.95

Fitbit has been around for over a decade now and is still regarded as one of the best fitness / health trackers available. With various models in the range they offer a wide range of functionality. Some of the features are listed below.

- Preloaded workouts
- Step counter
- GPS
- Heart rate tracking
- Multi-sport & SmartTrack Wireless syncing
- Swim proof
- Altimeter
- Gyroscope
- Accelerometer

With an in-built Li-pol battery the user can expect up to five days usage and 10 hours when using GPS [Fitb19]. Compatible with MAC OS X, Windows 10 and Android devices it used BLE for data synchronisation and can store up to 7 days of motion data minute-by-minute [Fitb19]. Much like the POLAR devices the Fitbit, while compatible with up to 200 devices [Fitb19], contains proprietary technology and code within and as such are locked to a unidirectional data flow unless used in conjunction with Fitbit authorised applications.

3.5 Conclusion

In conclusion the most suitable hardware to use for this project, of the technology researched, would seem to be the Movesense HR+. With its open API, cheap price range, availability of various accessories and easily accessible extensive documentation it makes sense to develop with this particular device. When thinking of the end user and potential saleability of the end product, the relative low cost of multiple of these sensors, possibly combined with a tripod and cheap tablet or mobile device to create a package for the user, also lends to the decision to develop with the Movesense HR+.

4. Software

4.1 Front End Technologies

4.1.1 Xamarin

Xamarin is an open source extension of the .NET platform which provides tools and libraries specifically for the development of cross platform applications. It uses an extensible markup language, known as XAML, to build dynamic mobile applications in C#. It incorporates platform specific libraries which allow access to platform specific Application Programming Interfaces (API). Applications built in Xamarin leverage native hardware acceleration and perform ahead-of-time (AOT) compilation to reduce start up time and increase performance.



Figure 8. Xamarin architecture [Docs19]

As can be seen in Figure 8, Xamarin allows applications to be written on either PC or Mac and utilise the platform native UI via a shared application code written in C# / XAML. Xamarin provides the ability to directly invoke Objective-C, Java, C and C++ libraries giving the power of multi-language code and libraries and the ability to utilise a wide array of third-party code along with the extensive .NET infrastructure. With the ability to develop cross platform applications relatively easily and the use of a unified API to access common resources across multiple platforms, Xamarin provides a unique opportunity for developers to expand their potential target market.

4.1.2 IONIC

IONIC is an open source UI toolkit for building mobile and desktop applications using web technologies (HTML, CSS and JavaScript) [Ioni19]. It can be integrated with ANGULAR and REACT frameworks to create an immersive cross platform UI experience all with one code base. Functionality comes primarily from the use of JavaScript which allows IONIC to work as standalone via the use of script tags. As can be seen in Figure 9, the fact IONIC uses web-based languages, enables cross platform compatibility as all OS platforms render web applications in a similar way with the use of native API's.

IONIC uses Capacitor plugins in more recent times to gain access to host OS features such as camera, GPS, Accelerometer, etc..., having used Cordova plugins previously. It allows developers to code in a web environment then customise the applications for various platforms along with the ability to automate code deployments and builds. Ionic's additional CSS styling enables user interfaces to look like a regular native app [Majc17].

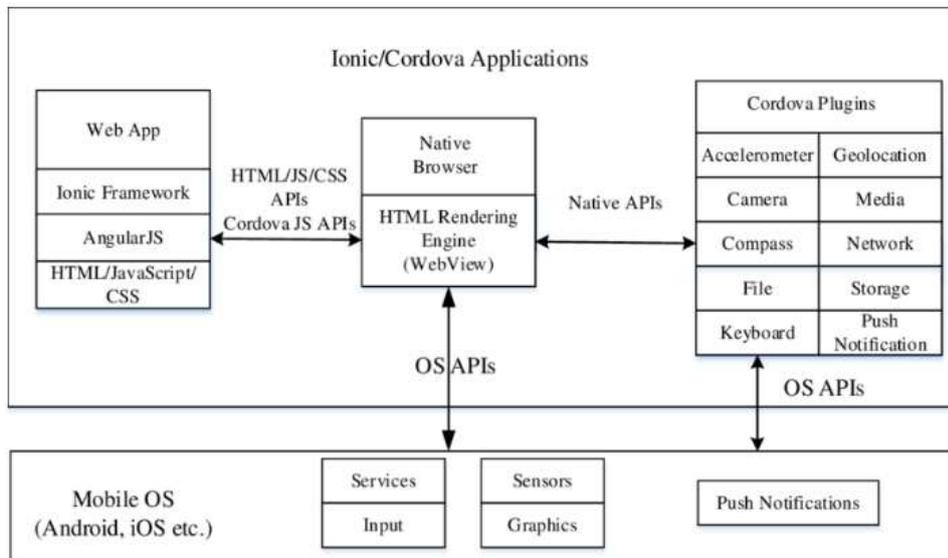


Figure 9. IONIC Architecture [Qian16]

4.1.3 Kivy

“Kivy is an open source Python library for rapid development of applications” [Kivy19]. Built on the back of the extensive python language Kivy can be utilised as an additional library with python to build cross platform applications that are not only visually appealing but also easy to use. Kivy “runs on Linux, Windows, OS X, Android, iOS and Raspberry Pi” [Kivy19] with little adaptation needed for use of native functionality. With the use of relatively simple graphics a developer can easily ensure continuity between screens of their application enabling the developer to focus on functionality rather than needing to continually refactor CSS and styling attributes. Functionality is provided through the use of widgets and predefined layouts, thus increasing the speed of development. With extensive and well written documentation on not only python but also the fact Kivy has easy to follow documentation on kv language, widgets and predefined functions, this enabled the developer to easily create a functional and well laid out application with relative ease, provided they have some experience with python.

4.1.4 Conclusion

Having used IONIC with some degree of difficulty during installation and application compilation, combined with the limitations of JavaScript to access native platform functionality, this was deemed to be unfit for use on this project. Kivy, whilst python based and relatively powerful, would not be suitable due to the amount of time investment needed to become proficient enough to execute an application of sufficient standard.

Xamarin has been chosen to develop the front end of the application. With extensive documentation, library compatibility, native API interface capabilities, AOT compiling to increase application speeds and extensive .NET support, the amount of support available far surpasses its rivals. Xamarin, being able to run through Visual Studio, gives the developer the ability to utilise the debugging and platform virtualisation capabilities of Visual Studio, the author believes this would be best suited to develop the application.

4.2 Back End Technologies

4.2.1 Cloud

4.2.1.4 Google Cloud

Google cloud offers multiple products for Cloud storage, Database hosting, Application hosting, API management, Developer tools, Networking to name a few as part of its Platform as a Service (PaaS). Cloud SQL is one such service which allows the creation, maintenance and administration of MySQL, SQL and PostgreSQL relational databases in the cloud [Goog19a]. Compatible with Java, C++, Python, Go and Ruby [Goog19a], Google cloud bridges the barriers between multiple programming languages giving them a unified platform from which to operate. Google Cloud also provides access to the Google App Engine, a serverless web framework which allows developers to run web-based applications in Google data centres [Goog19b]. Google App Engine is compatible the same languages that Google cloud is, as well as the C# and .NET languages. The Google App Engine automatically handles scaling from zero to planet scale [Goog19b]. While price plans may be chosen to only pay for resources that are used, while your application runs in a serverless environment, generally speaking, the cost to avail of these services is quite high.

4.2.2 Local

4.2.2.1 SQLite

SQLite is a library, based in the C language, that implements a small, fast, self-contained Sequential Query Language (SQL) relational database stored within the mobile device being used [SQLi19]. It uses the internal memory of the device to store data without the need of a server like MySQL, AWS or other non-native databases. A complete SQL database with multiple tables, indices, etc... is contained and stored in a single file [SQLi19]. The file format which it is stored in is cross platform compatible allowing for the use of the same database across multiple platforms if transferred from one device to another. Creating an SQLite database is relatively simple and does not require any database setup or administrative permissions. One needs to simply define SQLite statements to Create, Read, Update and Destroy (CRUD) and the platform OS will handle the rest.

4.2.3 Conclusion

In conclusion SQLite would be sufficient to test functionality in the short term allowing the developer to CRUD database tables and entries as required. As it is built into all mobile devices and is self-contained it will provide an adequate database to run queries and test the application from. However, as scalability is a factor in developing any desirable multiuser application in the future, SQLite would not be adequate as it is device based and does not have the capability or storage for multiple remote users.

While Google cloud would provide an ideal and scalable cloud service to run the application from, in general, the cost of the PaaS cloud service is unattainable and unjustifiable to the developer. The costs would invariably increase the base cost of the application to the end user which could, and likely would, make the application undesirable to the end user.

Appendix

i. Bibliography

- Cicek15** Cicek, M. (2015). *Wearable Technologies and its Future Applications*. 4th ed. [ebook] Yalova, Turkey: International Journal of Electrical, Electronics and Data Communication, p.46. Available at: https://www.academia.edu/12138425/WEARABLE_TECHNOLOGIES_AND_ITS_FUTURE_APPLICATIONS [Accessed 29 Oct. 2019].
- Copa19** Coparco.org. (2019). [online] Available at: https://www.coparco.org/Coaches/CoachManuals/Coaches_Manual_Lev1eng.pdf [Accessed 29 Oct. 2019].
- Docs19** Docs.microsoft.com. (2019). *What is Xamarin? - Xamarin*. [online] Available at: <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin> [Accessed 30 Oct. 2019].
- Fitb19** Fitbit.com. (2019). *Fitbit Ionic: adidas edition watch*. [online] Available at: <https://www.fitbit.com/ie/shop/adidas?color=blue-gray> [Accessed 1 Nov. 2019].
- Goog19a** Google Cloud. (2019). *Cloud SQL | Google Cloud*. [online] Available at: <https://cloud.google.com/sql/> [Accessed 1 Nov. 2019].
- Goog19b** Google Cloud. (2019). *App Engine | Google Cloud*. [online] Available at: <https://cloud.google.com/appengine/> [Accessed 1 Nov. 2019].
- IDC19** IDC: The premier global market intelligence company. (2019). *IDC - Smartphone Market Share - OS*. [online] Available at: <https://www.idc.com/promo/smartphone-market-share/os> [Accessed 21 Oct. 2019].
- Ioni19** Ionic Docs. (2019). *What is Ionic Framework? - Ionic Documentation*. [online] Available at: <https://ionicframework.com/docs/intro> [Accessed 30 Oct. 2019].
- Kivy19** Kivy.org. (2019). *Kivy: Cross-platform Python Framework for NUI*. [online] Available at: <https://kivy.org/#home> [Accessed 31 Oct. 2019].
- Liu19** Liu, S. (2018). *Smartwatch unit sales in the US 2013-2018 | Statista*. [online] Statista. Available at: <https://www.statista.com/statistics/381696/wearables-unit-sales-forecast-united-states-by-category/> [Accessed 29 Oct. 2019].
- Lofb19** Lofblom ,J 2019, “Prototyping with Movesense Platform – Breathing Application”, M.Sc Thesis, Metropolia University of Applied Sciences, Helsinki.
- Mack19** Mackenzie, T. (2019). *App store fees percentages and payouts what developers need to know*. [online] TechRepublic. Available at: <https://www.techrepublic.com/blog/software-engineer/app-store-fees-percentages-and-payouts-what-developers-need-to-know/> [Accessed 23 Oct. 2019].
- Majc17** Majchrzak, T., Biorn-Hansen, A. and Gronli, T. (2017). Comprehensive Analysis of Innovative Cross-Platform App Development Frameworks. In: *50th Hawaii International Conference on System Sciences*. [online] Hawaii: HICSS, p.6168. Available at: <http://hdl.handle.net/10125/41909> [Accessed 31 Oct. 2019].
- Mich13** Michie S, Richardson M, Johnston M, Abraham C, Francis J, Hardeman W, et al. The behavior change technique taxonomy (v1) of 93 hierarchically clustered techniques: building an international consensus for the reporting of behavior change interventions. *Ann Behav Med* 2013 Aug;46(1):81-95.
- Mixp19** Mixpanel.com. (2019). *Mixpanel Trends - Mixpanel | Product Analytics*. [online] Available at: https://mixpanel.com/trends/#report/android_os_adoption [Accessed 22 Oct. 2019].
-

Pola19 Polar Global. (2019). *Polar Vantage V Titan | Multisport & triathlon watch with premium materials | Polar Global*. [online] Available at: <https://www.polar.com/en/vantage/v-titan> [Accessed 1 Nov. 2019].

Qian16 Qianlong ,L 2016, “DESIGN AND IMPLEMENTATION OF A NOVEL MOBILE-OPTIMIZED REMOTE LABORATORY PLATFORM FOR MOBILE LEARNING”, M.Sc Thesis, Texas Southern University, Houston.

Sqli19 Sqlite.org. (2019). *About SQLite*. [online] Available at: <https://www.sqlite.org/about.html> [Accessed 1 Nov. 2019].

Worl19 World Archery. (2019). *History of Archery*. [online] Available at: <https://worldarchery.org/History-Archery> [Accessed 29 Oct. 2019].

ii. Plagiarism Declaration



Work submitted for assessment which does not include this declaration will not be assessed.

DECLARATION

*I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our 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 offence.

Student Name: (Printed) BRENDAN BROWNE

Student Number(s): 00223413

Signature(s): B. Browne

Date: 01-11-2019

Please note:

- * Individual declaration is required by each student for joint projects.
- Where projects are submitted electronically, students are required to type their name under signature.
- The Institute regulations on plagiarism are set out in Section 10 of Examination and Assessment Regulations published each year in the Student Handbook.