



PetSpace

RESEARCH MANUAL

Student: Emma O'Connor
Student Number: C00237292
Supervisor: Paul Barry
Submission Date: 17/04/2023



Contents

Table of Figures	2
1. Introduction	3
2. Description Domain.....	4
3. Inspiration.....	5
3.1 Existing Systems	5
3.1.1 VetSCOPE	11
3.1.2 TeleOSS	12
4. Technologies.....	13
4.1 IOS, Swift.....	13
4.1.1 React Native.....	15
4.1.2 Xamarin.....	16
4.1.3 Flutter.....	17
4.2 Android OS, Java	18
4.3 Visual Studio Code.....	20
4.4 Data Storage	21
4.4.1 MariaDB	21
4.4.2 SQLite.....	22
4.4.3 Firebase.....	23
5. Conclusion	24
6. Bibliography	25

Table of Figures

Figure 1 Veterinary Software Pro	5
Figure 2 Veterinarian Section of Veterinary Software Pro.....	6
Figure 3 Veterinary Software Pro Main Menu	6
Figure 4 Clients Page	7
Figure 5 Reports page	7
Figure 6 Schedulers Page.....	8
Figure 7 Petsapp on Google Play.....	9
Figure 8 Petsapp chat to clinic	10
Figure 9 Petsapp appointments	10
Figure 10 VetSCOPE logo.....	11
Figure 11 TeleOSS logo.....	12
Figure 12 IOS Swift programming language.....	14
Figure 13 React Native logo	15
Figure 14 Xamarin logo	16
Figure 15 Flutter	17
Figure 16 Android OS Java code example	19
Figure 17 Visual Studio Code Logo	20
Figure 18 MariaDB logo	21
Figure 19 SQLite logo	22
Figure 20 Firebase Logo	23

1. Introduction

The purpose of this document is to outline the research carried out in order to create **“PetSpace – the Pet Management Application”**.

This document will contain a background of the proposed application including the main topics of interest; Description Domain *e.g., pet information, owner(s) information and vet information etc.*, the inspiration for the application including examples of *existing systems* holding similarities to the application, insight into the type of system vets use within their practice from field (in person) research *e.g. TeleOSS*. Technologies to consider *e.g., Android OS, IOS*, platforms such as *Flutter, React Native and Xamarin*, the programming languages used, features included in the application and lastly where the data can be stored.

The aim of this application is to provide users with a way to keep track of vet appointments, vaccinations and the pet itself.

2. Description Domain

PetSpace will be an application for owners to keep track of their pet history and vet appointments *e.g.*, *vaccination appointments*. The owner can store their pet details on the application and create appointments with their local vet for their specified pet, the type of appointment being booked, the date and the time. Confirmation emails will be sent to the owner upon booking to remind them of their upcoming appointment with their vet which will be useful for the pet owner. A copy of the appointment will also be sent to the application's system. Alongside booking an appointment, pet owners will have the freedom to modify their appointment details if they need to and if circumstances arise; remove their appointment before the allocated time. Pet owners can also contact the vet via email for queries related to an animals health without needing to book an appointment.

Owner and vet information can also be stored. Vets will be able to login securely to the application and view their upcoming appointments for the next two weeks. They will also have access to past appointments. Vets can analyse information given about the specified pet in order to provide feedback and visualisation about the pet's health and wellbeing *e.g.* reports. These analysis reports can then be viewed and printed for the vet's records. Possible diagnosis will be sent to the pet owner via the application about their pet after gathering the necessary information needed. This will be achieved using data analytics.

3. Inspiration

The inspiration for this application came from my pet. When she became ill, we brought her to my local vet to which the receptionist asked when she was last vaccinated. Having an application that would track vaccination history of your pet, I found would have been ideal during this time, along with other means of keeping track of my pet history.

3.1 Existing Systems

Upon research, there are similar applications across the Internet. The first application discovered is a paid application from the Microsoft online store (see figure 1).

Veterinary Software Pro is a mobile pet management application with premium features such as client information, accounting, records and schedules managed around the pet in question. It is available for both mobile, Microsoft Surface devices and desktops. The necessary Operating System required to run this application is Windows 10 or higher. The recommended Architecture is x64 for the *Veterinary Software Pro* application.



Figure 1 *Veterinary Software Pro*

Its main purpose is to manage and record information regarding appointments. Its interface (see figure 3) is simple with four main buttons: *Veterinarians*, *Clients*, *Reports* and *Schedulers*. The *Veterinarians* section (see figure 2) includes all important information relating to local vet's e.g., phone number, address etc. The *Client's* section (see figure 4) includes all necessary information relating to the client (animal in question). The *Reports* section (see figure 5) details information on previous appointments and/or vaccinations. The *Schedulers* button (see figure 6) relates to creating new date appointments for the client.

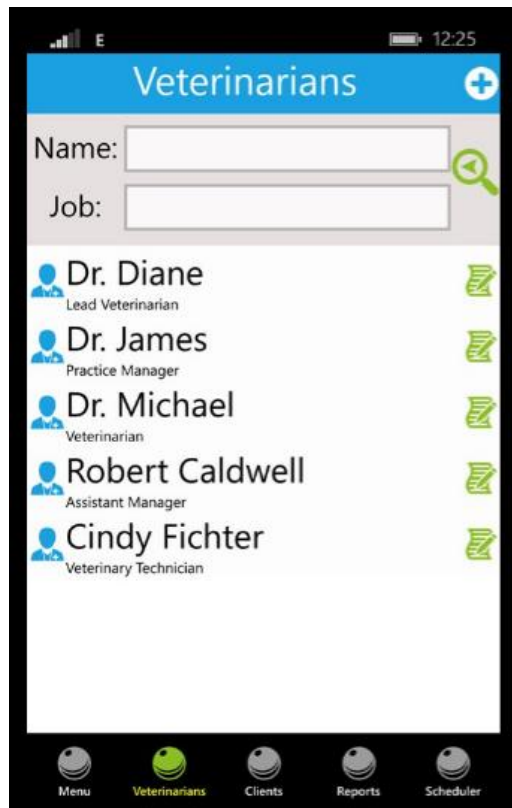


Figure 2 Veterinarian Section of Veterinary Software Pro



Figure 3 Veterinary Software Pro Main Menu

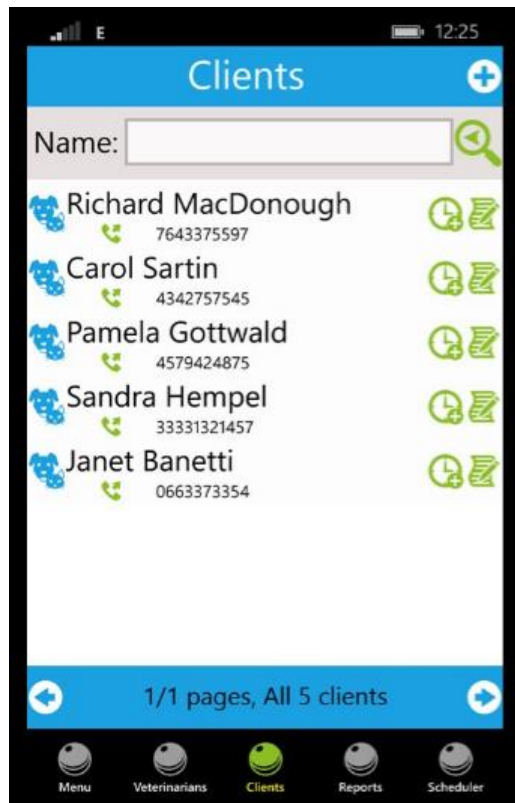


Figure 4 Clients Page

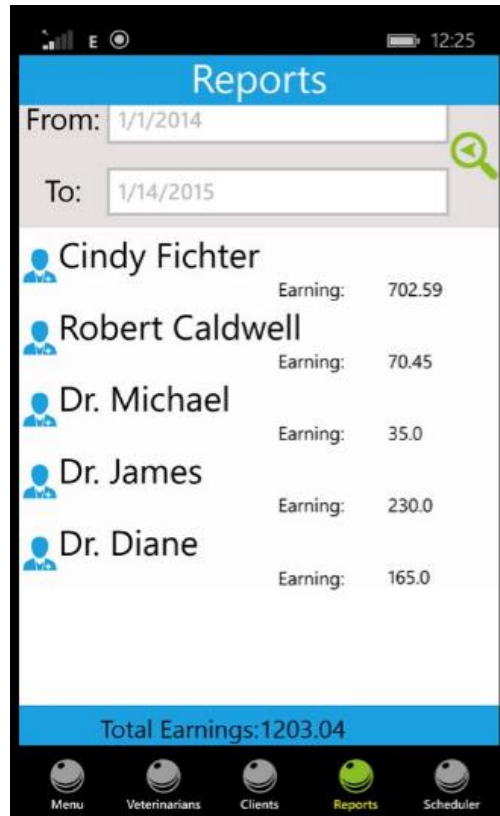


Figure 5 Reports page



Figure 6 Schedulers Page

Another existing Pet Management Systems available to users is a free mobile application that does not require additional or upfront expenditures.

Petsapp (see figure 7) is a mobile application that is free on Google Play store that allows the user to chat to their vet and have video consultations about their pet (see figure 9). There is also the option to book appointments for your pet within the application and retrieve answers about your pet's health and better understand their overall wellbeing (see figure 8).



Figure 7 *Petsapp* on Google Play.

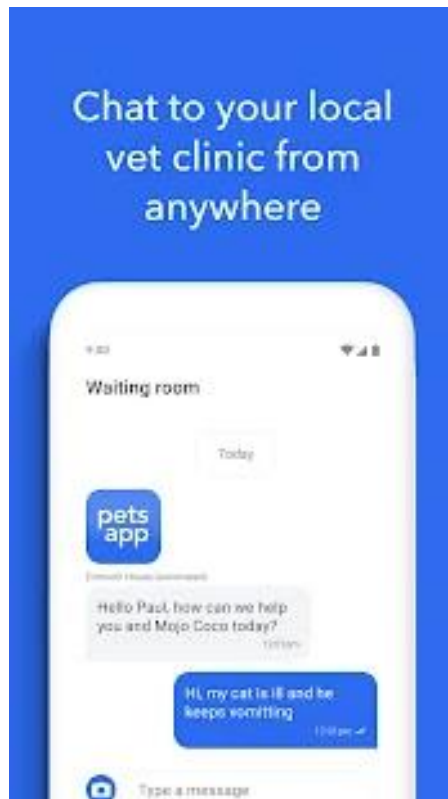


Figure 8 Petsapp chat to clinic

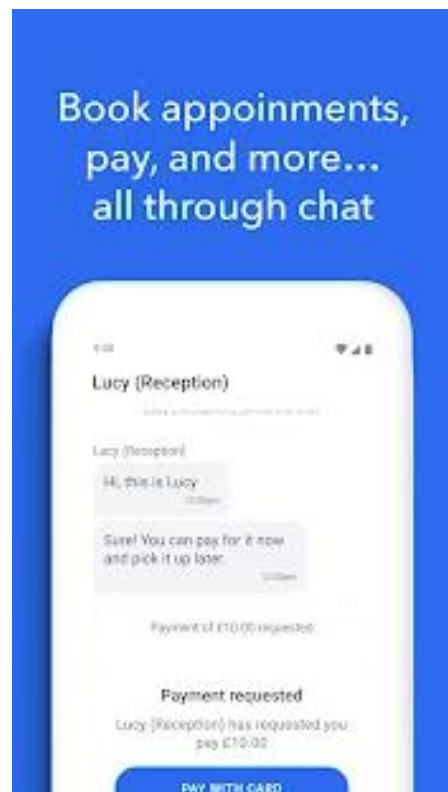


Figure 9 Petsapp appointments

The last form of research into existing systems was from a local vet practice in Portarlinton Co. Laois as part of the field research for the application. Important information was gathered about the type of system and approach the vets take in keeping track of their clients and scheduled appointments along with vaccinations e.g. worm dose. By doing this, there is a clear understanding of the main features vets require to undertake daily requirements and what are the basic requirements for the Pet Management Application.

The vet practice use two online softwares in order to achieve this: VetSCOPE and TeleOSS.

3.1.1 VetSCOPE



Figure 10 VetSCOPE logo

VetSCOPE (see figure 10) is a veterinary management system and also accounting management system created for veterinary practice. It is owned by an Irish company known as Lawler Developments Limited and as of today, there are more than 1,000 users of the software.

The purpose of VetSCOPE is to allow the user to control all aspects of their business. It manages animal/equine, Customer Relationship Management (CRM), prescription printing and accounts for large animals. It is flexible and very easy to use.

VetSCOPE also come with their own application as well as a number of features including:

- Purchase Invoice Processing (PIP).
- Electronic Signature Processing (ESP).
- SMS Service
- Barcode Scanner

The application is cross-platform meaning it can run on a Mac machine and a Windows machine. There is also a new VetSCOPE app for iPhone, iPad and iPod Touch available for the vet practice. This will allow for the user to create prescriptions and bills simultaneously, email/print wirelessly with no extra expenditures for hardware

3.1.2 TeleOSS



Figure 11 TeleOSS logo

TeleOSS (see figure 11) is an SMS gateway software solution provider. They offer a solution platform focusing on the SMS industry. They have received the award as the best software in 2014. Their solutions are created and designed in order to support large scale SMS and OTT messaging as well as innovative features in providing the user with the best service at a reasonably cheaper cost.

TeleOSS Messaging suite platform include:

4. SMS Gateway Software Solution
5. SMS router software
6. SMS controller software
7. SMS Hub platform
8. OTT Messaging gateway
9. Smart City SMS Solution
10. Business Solutions

Due to its open architecture, customers can actively design and customise the solution by their requirements. They offer business features which support requirements, enabling customers to get a flavour for the competition.

TeleOSS is trusted by over 150 clients globally for large scale SMS business.

4. Technologies

The technologies being considered for this project are *Apple IOS*, *Android OS* for mobile devices along with *Visual Studio*. The programming language most commonly used for Apple IOS is a compiled programming language called *Swift* whilst for Android OS and Visual Studio it is *Java*.

Xamarin and *React* will also be explored in order to decide on the best platform for the application.

Data storage will also be discussed to store critical data from the application in the form of *MariaDB*, *SQLite* and *Firebase*.

4.1 IOS, Swift

Swift runs on multiple Apple devices such as *MacOS*, *IOS*, *watchOS*, *tvOS* and *Linux* applications whilst also including support for Windows platform. IDEs used with Swift include *Xcode*, *AppCode* and *Atom*. It is an open-source technology with a wide variety of third-party tools. It is also the fastest growing language known for being 2.6 times faster than Objective-C along with being 8.4 times faster than Python. In March 2019 it became backwards compatible with previous versions.

However, the language does have its limitations such as having limited talent pool. This can be difficult to find experienced developers in the language due to it being a relatively new programming language (2014). There is a poor interoperability with IDE's and third-party tools. It also lacks cross-platform support mainly being developed for native IOS development. This leaves the market open to the likes of React Native, Xamarin and Flutter.

An example of the programming language used is shown below (see figure 12).

```
1  let vegetable = "red pepper"
2  switch vegetable {
3  case "celery":
4      let vegetableComment = "Add some raisins and
      make ants on a log."
5  case "cucumber", "watercress":
6      let vegetableComment = "That would make a good
      tea sandwich."
7  case let x where x.hasSuffix("pepper"):
8      let vegetableComment = "Is it a spicy \(x)?"
9  default:
10     let vegetableComment = "Everything tastes good
      in soup."
11 }
```

Figure 12 IOS Swift programming language

4.1.1 React Native

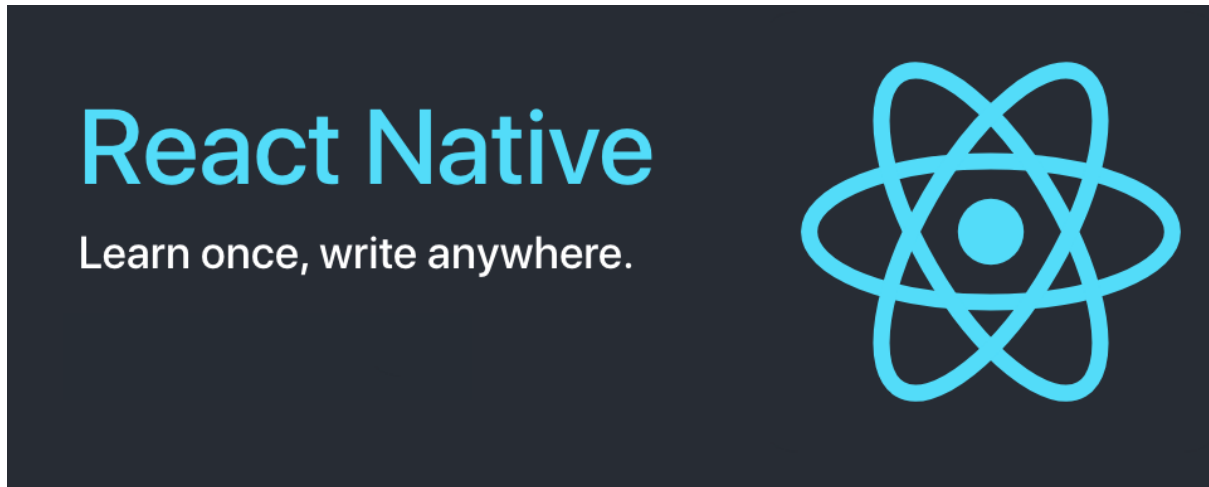


Figure 13 React Native logo

React Native (see figure 13) is one of the most actively developed on platforms available. It is a framework used to create mobile applications for IOS and Android. Examples of large companies that integrate React Native into their system would be *AirBnB*, *Salesforce* and *Walmart*.

There are constant updates being made bi-weekly. React Native-based applications are extremely similar in regard to performance, look and feel when comparing to Native apps. Libraries allow the developer to deploy their code as a web-application. React Native also use *JavaScript*, in which finding developers will not be a challenge as JavaScript has been available for many years. It is especially useful for applications that are visually appealing and require user input.

A downside of using React Native would be that it is not reliant on performance.

4.1.2 Xamarin

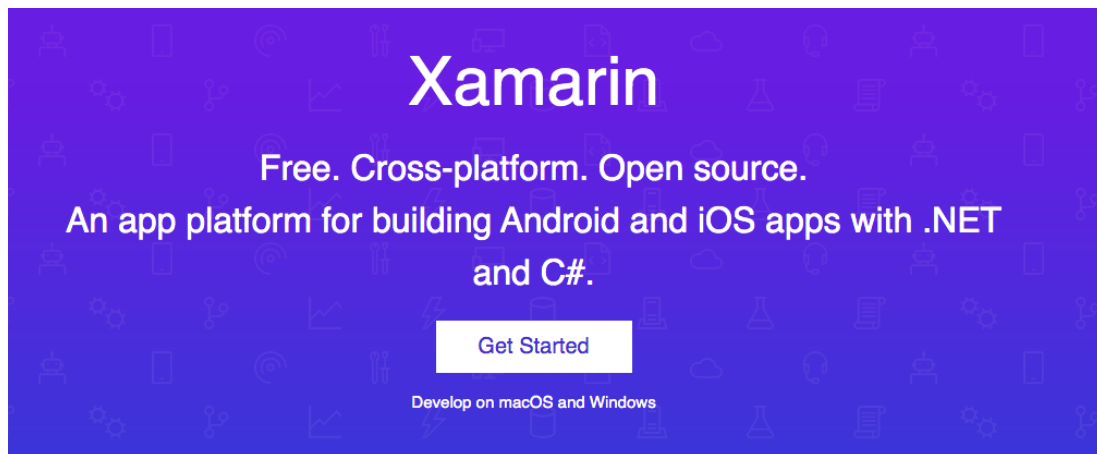


Figure 14 Xamarin logo

Xamarin (see figure 14) is owned by Microsoft, was released in 2011, marking it the oldest of the three platforms. Though the performance was high, it lacked severely for graphically heavy or complex applications. It is a free, cross-platform and open-source platform for creating Android and IOS apps using C# and .NET. Examples of companies that integrate Xamarin into their business are *UPS*, *BBC*, *Good Food* and many others.

Amongst the three, Xamarin executes code the fastest. The user interface (UI) is also very smooth while having the ability to compile IOS code on Windows machines.

In order to use Xamarin however, previous experience is required with the platform as there is a miniscule community with the ability to support any challenges that may occur. It is also not as popular as Flutter or React Native therefore, finding experienced developers may be difficult. Updates to the platform are slow and utilizing the features of newly Android/IOS updates may be immeasurable.

4.1.3 Flutter

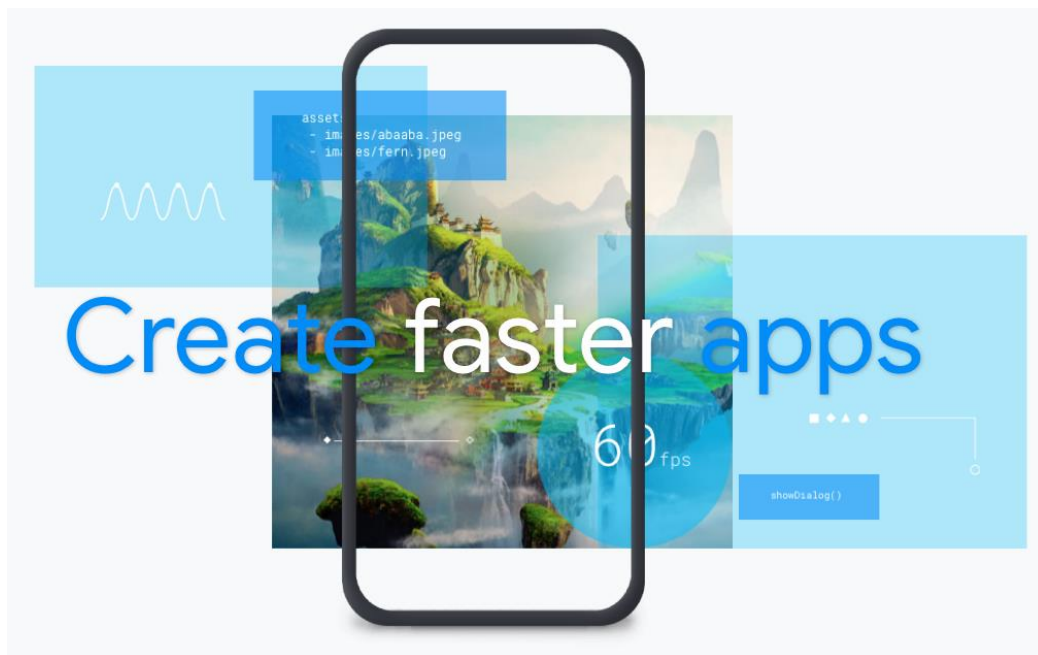


Figure 15 Flutter

Flutter (see figure 15) is the youngest platform to enter the market. It is popular in its approach to the UI development, making it easy to both create and update the user interface. It uses *Skia* graphic engine which allows the application created to look identical no matter the platform. Plugins and platform channels utilize features of the OS easily. There is also Flutter for Web, making deployment to the web less complex. For businesses, Flutter is appealing for their development speed, cost and mimicked native performance.

As it is the youngest, stability and documentation issues arise constantly. There are less features in Dart and the interfaces Skia helps to create only imitate native elements.

4.2 Android OS, Java

Android is an open-source platform. The Java programming language is the most used language when developing for Android due to its versatility and smooth development process. *Android Studios* is a leading popular IDE for Android development which includes Gradle-Based support. Along with *PhoneGap* which is a cross-platform IDE, *Visual Studio* – Xamarin which was built by Microsoft and a popular IDE for both Android and IOS developers and *IntelliJ IDEA* – unfortunately this IDE only serves development for Android app projects.

There are a number of benefits when it comes to developing on an Android system. It is less restrictive when compared to IOS in terms of its flexibility. It allows the application to run on any Android device with little issues from hardware compatibility.

Unlike IOS, Android are not limited to what platform their devices can run on. They allow for the option of adding extra functionality to extend to VR headsets, Wear OS devices and many others. Their ability to integrate an application into a wide variety of platforms is an ideal example of their development being of multipurpose use.

Unfortunately, due to Android being an open-source platform, this leaves the application open to a multitude of cyber-attacks. I will ensure that the security of the Vet Management application is not less favoured and just as much of a key importance as the *Functionality, Usability, Reliability, Performance, Scalability* and *Security* (FURPS+).

An example of Java code can be seen below (see figure 16).

```
Android1.java* [X]
package com.Android1;

import android.app.Activity;
import android.widget.TextView;
import android.os.Bundle;

public class Android1 extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        /* Create a TextView and set its text to "Hello world" */
        TextView tv = new TextView(this);
        tv.setText("Hello World!");
        setContentView(tv);
    }
}
```

Figure 16 Android OS Java code example

4.3 Visual Studio Code



Figure 17 Visual Studio Code Logo

Microsoft offers an integrated development environment called *Visual Studio*. Computer programs, including websites, web applications, online services and mobile applications are developed using it.

On Windows, macOS and Linux, *Visual Studio Code* is a standalone source code editor. The preferred choice for web developers and JavaScript, with extensions to support almost any programming language. Git is also built-in allowing the user to make commits from the editor itself and push/pull into the editor.

With Microsoft *Azure*, you can easily store and query relational and document-based data, build and host your React, Angular, Node, Python etc. websites, and scale with serverless computing all from within Visual Studio Code.

4.4 Data Storage

Data storage can exist locally on the device or remotely. Storing the data will be considered on the cloud e.g., *Firebase* or in a database such as *MariaDB* or *SQLite*. The type of data to store will be meta data from the main features of the Vet Management application e.g., *petName*, *petAge*, *lastVaccinated* etc. The data will be stored when the user using the application saves their pet/owner information or contacts their vet through the app. Their vaccination and vet appointment history will also be stored for the pet.

4.4.1 MariaDB



Figure 18 MariaDB logo

MariaDB (see figure 17) is an open-source relational database. It is one of the most popular and widely used databases, created by developers of MySQL. It is accompanied by majority cloud services and Linux distributions. MariaDB is created with the importance of performance, stability and openness. Functionalities include compatibility features alongside Oracle Database and Temporal Data Tables. It's latest version as of September 2021 is 10.7.0. Large sponsors include Microsoft, Alibaba Cloud and Tencent Cloud.

MariaDB plugins (software components) can be added to the core software without needing to rebuild the entire server. They can be storage engines, security requirements and others. They can be loaded upon start-up, during initialization or dynamically loaded when the server is running.

4.4.2 SQLite



Figure 19 SQLite logo

SQLite (see figure 18) utilises the C-language library to implement a small, fast, high-reliability and fully featured database engine. It is the most commonly used database engine built into majority of all mobile devices, computers and other devices. The file format is cross-platform, stable and backwards compatible. Database files are interpreted as containers which transfer content between systems. The source code for SQLite is free to use and members include Bentley, Bloomberg, Expensify and Navigation Data Standard.

A range of features include:

- zero-configuration meaning setup or administration requirements are not needed.
- It has a fully-featured SQL implementation with capabilities like JSON, common table expression and Windows functions.
- The support for terabyte sized and gigabyte sized databases.
- It has an easy to use API.
- Well commented source code.
- Cross-platform.

4.4.3 Firebase



Figure 20 Firebase Logo

Firestore (see figure 19) founded in 2011, is a cloud-based platform for building and growing popular apps and games. It requires NoSQL (non-Structured Query Language) and offers real-time hosting of databases *i.e.* data can be stored by users in real-time.

It is supported by Google and relied upon by millions of companies worldwide. One can install open-source, pre-packaged code bundles to automate typical development chores.

Extensions include:

- Trigger email – which will be used to handle email confirmations
- Stripe payments
- Hosting/Deployment
- Crash reports
- Analytics and
- Authentication

Firestore can be integrated with many tools such as; *Google Ads, Google Play, Slack, Jira* and much more. Data is stored as JSON (Javascript Object Notation) and projects through Firestore are also backed by Google Cloud for both online and offline use resulting in little loss of data. Firestore is trusted by some of the largest games and applications in the world including *Venmo, Duolingo, Alibaba, Trivago, Gameloft* and *Wattpad*.

Cross-platform SDK's and detailed documentations are provided by Firestore in order to aid shape and build apps across multiple platforms such as *IOS, Android, the Web, C++* and *Unity*.

5. Conclusion

Upon completion of the Research Document entailing the technologies and software discovered, the following decisions have been made in order to ensure the completion of this project is successful along with the testing of environments/platforms.

Visual Studio Code: This IDE I found to be easier to grasp the concept of when compared to IntelliJ or Visual Studio. It also had a seamless integration with Git and the array of useful dependencies.

React Native: Having downloaded React Native alongside Visual Studio code, tutorials and documentation have been looked at in extensive detail in order to fully grasp the concept of the React Native environment. It was less time consuming to pick up the development structure because it was similar to the works of Angular.

Xamarin: Difficulty was proven when testing out the Xamarin framework. Visual Studio Code doesn't support the Xamarin framework so the development for the project was constrained to Visual Studio. Multiple problems arose when creating an emulator alongside many dependencies for Xamarin having been slowly deprecated as of 2021. Taking in these factors, it proved difficult to test and develop.

6. Bibliography

Altexsoft Software r&d Engineering. (2021). *The Good and the Bad of Swift Programming Language* [online]. Available from: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-swift-programming-language/> [Accessed 15/10/2021].

Rami Anwar. (2020). *Android vs IOS Development: Which Platform Should I Develop for First?* [online]. Eastern Peak. Available from: <https://easternpeak.com/blog/android-vs-ios-development-which-platform-first/> [Accessed 15/10/2021].

Selected Firms. (2021). *16 Best Android IDEs for App Development in 2021* [online]. SelectedFirms. Available from: [16 Best Android IDEs for App Development in 2021 \(selectedfirms.co\)](https://selectedfirms.co/16-best-android-ides-for-app-development-in-2021/) [Accessed 22/10/2021].

Mike James. (2015). *Visual Studio V Android Studio* [online]. I Programmer. Available from: <https://www.i-programmer.info/news/193-android/9163-visual-studio-v-android-studio.html> [Accessed 31/10/2021].

Ego Creative Innovations Limited. (2021). *Flutter VS Xamarin VS React Native: What's Best in 2021?* [online]. Ego. Available from: <https://www.ego-cms.com/post/flutter-vs-xamarin-vs-react-native-whats-best-in-2021> [Accessed 31/10/2021].

MariaDB Foundation. (2021). *MariaDB Server: The open source relational database* [online]. MariaDB Foundation. Available from: <https://mariadb.org/> [Accessed 31/10/2021].

SQLite Homepage. (2021). *What is SQLite?* [online]. SQLite. Available from: <https://www.sqlite.org/index.html> [Accessed 31/10/2021].

SQLite Features. (2021). *Features of SQLite* [online]. SQLite. Available from: <https://www.sqlite.org/features.html> [Accessed 31/10/2021].

VetSCOPE. (2021). *VetSCOPE – Veterinary Practice Management Solutions* [online]. VetSCOPE. Available from: <http://www.vetscope.ie/> [Accessed 01/11/2021].

Firebase. (2022). *Firebase* [online]. Firebase. Available from: <https://firebase.google.com/> [Accessed 07/11/2022].

Visual Studio Code. (2022). *Visual Studio Code – Code Editing Redefined* [online]. Visual Studio. Available from: https://code.visualstudio.com/?wt.mc_id=DX_841432#built-in-git [Accessed 07/11/2022]