CS Capstone Technology Review

November 18, 2017

# Remote Seed Identification

Prepared for

Crop and Soil Science Department, OSU

Dan Curry

Prepared by

Group 12

Remote Seed

Alex Ruef

**Abstract**

When working on a project there are many tools for the job. Each tool has a different purpose and a different focus to get the job done the way you want it done. In this paper I have outlined three major compoments of our project. For each component I have chosen three tools which I compare and contrast before choosing one to be used in our project.

# CONTENTS

1 UI Framework 2

1.1 Overview 2

1.2 Options 2

1.2.1 Xamarin 2

1.2.2 Corona SDK 2

1.2.3 Appcelerator 2

1.3 Comparison 2

1.4 Conclusion 3

2 Computing Device 3

2.1 Overview 3

2.2 Options 3

2.2.1 Jetson TX2 3

2.2.2 Raspberry Pi 3 Model B 3

2.2.3 Arduino Tian 3

2.3 Comparison 4

2.4 Conclusion 4

3 Database Management System 4

3.1 Overview 4

3.2 Options 4

3.2.1 Oracle 4

3.2.2 MySQL 4

3.2.3 Microsoft SQL Server 5

3.3 Comparison 5

3.4 Conclusion 5

References 5

# 1 UI Framework

## 1.1 Overview

When writting an app or software that will be used by users of differing technical backgrounds much care must be placed into the UI. UI frameworks are tools that aid in the production of graphical user interfaces by bringing in functionalities and APIs from a wide range of sources. A goal of many of these frameworks is to abstract further away from the device platform and make cross-platform development easier. There exists many free UI frameworks all with differing backgrounds and utilizing different programming languages. I have chosen three of the most popular UI frameworks to examine for use in our project.

## 1.2 Options

### *1.2.1 Xamarin*

Xamarin is a UI framework built by Microsoft and integrated straight into Visual Studio. It is built on the C# language commonly used in app development. Within Xamarin you are not limited to C#, however, as you have full access to the Android API and you can call on existing Java code [1]. Xamarin also comes with an advanced test framework. This framework allows you to create full automated test suites that tests your app as if a user was using it. You can create tests based on actual user gestures such as fingure taps and send the tests to be run on actual devices in the cloud [2].

### *1.2.2 Corona SDK*

Corona SDK is a UI framework that calls itself ”The 2D Game Engine”. It was created to be a lightweight and fast performing framework. Corona utilizes the Lua language which also strives to be very lightweight as well as being well-rounded. However, you are not limited to Lua, Corona allows you to call on any native library such as Java and C++ or Swift for iOS development. There is no single Corona IDE but it can integrate with many popular IDEs and it has a simulator to let you test your code as you write it. Another useful testing tool is live testing which allows you to instantly send a new build of your app to devices on your local network [3].

### *1.2.3 Appcelerator*

Appcelerator combination of many platforms and APIs all put together into one UI framework. Cross-platform development is a key feature of all UI frameworks but Appcelerator makes it one of their main goals. To do this the framework is built on the Javascript language. Javascript is hated by many but its cross-platform capabilities make it nothing to scoff at and a good choice for Appcelerators goals. Appcelerators API Builder is a unique way to build APIs through a graphical user interface. The API Builder also comes with support for many different kinds of databases to easily include that data in your app [4].

## 1.3 Comparison

Right off the bat Xamarin’s testing capabilities is a huge advantage over the other frameworks. The ability to easily create automated test and have them run on a variety of actual hardware is invaluable. As far as programming language choice goes each language has its benefits and downsides depending on how it will be used. All of the frameworks support other languages anyways so there is a lot of flexibility here. When comparing cross-platform capabilities Appcelerator is a clear winner. That is not to say that the other two frameworks don’t provide significant cross-platform capabilities. Corona is a winner in the performance department with Lua being so lightweight. All frameworks support a performance monitor to fine tune your apps. Appcelerator with its graphical UI seems to cater to a less technical audience and thus should be easier to use. On the other hand Javascript can be frustrating to use and one thing Corona and Lua like to boast about is how easy Lua is to learn. However, easy can sometimes be a burden to more experienced programmers who want more power or don't need a graphical UI to get stuff done.

### 1.4 Conclusion

Our app will be a relatively simple one. We don't need flashy graphics or to perform heavy calculations. With this in mind Corona's performance advantage is not as useful to us. We are only planning to develop the app for Android so cross-platform is not a huge concern to us. That being said the simplicity of our app means that a lot of what we write will be cross-platform anyways. When comparing languages Javascript leaves much to be desired when up agaisnt C# and Lua. Since cross-platform capability is not a concern to use C# or Lua stand out. Lastly, the test framework in Xamarin is far to good to ignore. Corona and Appcelerator both have simulation capabilities but being able to generate tests that behave as a user would save us a lot of time and resources. Because of the test framework and other comparisons we have decided to use Xamarin as our UI framework.

## 2 Computing Device

### 2.1 Overview

When picking a device to run your program on you have many things to consider. The system needs to be powerful enough to run your program. You need to consider what operating system works on the hardware or if you even need an operating system. Each piece of hardware is designed with different functions in mind. Since we will be doing imaging and machine learning in our project we need a device with lots of power.

### 2.2 Options

#### *2.2.1 Jetson TX2*

NVIDIA calls its Jetson TX2 ″an AI supercomputer on a module″ [5]. The Jetson TX2 is designed to be a power house in a small box. The JetPack operating system designed for the Jetson and built from Linux comes installed on the Jetson [6]. It comes with a dedicated NVIDIA Pascal graphics card. The Jetson comes with a Quad ARM and a Dual Denver CPU. For memory the Jetson has 8 gigabytes of DDR4 RAM [5].

#### *2.2.2 Raspberry Pi 3 Model B*

The newest Raspberry Pi is designed to be accessible and easy to use. The NOOBS operating system was created to run on Pis but it does not come with the system. You must supply your own microSD card with either NOOBS or a different supported operating system installed. The Pi's processing power is a single quad core 1.2GHz CPU. For memory the Pi has 1 gigabyte of RAM [7].

#### *2.2.3 Arduino Tian*

Arduino's are typically designed to be microcontrollers as apposed to the microprocessors listed above. The Arduino Tian, however, comes with both. The microprocessor supports a Linux distribution named Linino. It comes with an Atheros AR9342 CPU. The Tian also has 64 MB DDR2 [8].

### 2.3 Comparison

Comparing these three devices is difficult to do because they all fit different niches. When it comes to power the Jetson TX2 is the clear winner. It has more processing power than the Pi and the Arduino combined as well as more memory. The dedicated graphics card is another huge resource it has over its competitors. Now if you do not need something with that much power, the Raspberry Pi may be more up your alley. It is weaker than the Jetson but it still has a significant amount of power for its size and its still more powerful than the Arduino. What the Pi has over the Jetson is its usability. Raspberry Pis are designed to be easy to use and to work with and is marketed to people of a wide range of age groups. It was mentioned earlier that sometimes you don't need an operating system; this is where the Arduino shines. The Arduino Tian is lacking in power but it is not designed to do things that require lots of power. Programming for the microcontroller happens at a very low level so it is not easy to use for the inexperienced. The low level operations, however, give the programmer a lot of flexibilty with how the controller runs. The microprocessor is always there with an operating system to provide more functionalities.

### 2.4 Conclusion

To quickly sum up each device, the Jetson TX2 is made for power, the Raspberry Pi is made to be easy, and the Arduino is made to be an embedded system. For our project we are required to use a Jetson TX2. However, the Jetson is the best of the three options for our project. The Jetson was designed to do AI and machine learning which is exactly what we need it for. We will be analyzing images taken from phones and tablets that won't be the highest quality. To do this we need as much processing power we can get and that is what the Jetson TX2 was made for.

## 3 Database Management System

### 3.1 Overview

Database management systems (dbms) are tools which aid you in handling your database or other data storage method. There are many different dbms options each focusing on a different database model. I have chosen three dmbs options based around relational databases.

### 3.2 Options

#### *3.2.1 Oracle*

Oracle describes the newest version of its dbms to be "the Worlds First Self-Driving Database" [9]. Machine learning is used to automate routine maintenance. Oracle dbms allows you to deploy a private cloud in house and behind your own firewall. They also have strong support for cloud based solutions in general [10]. Oracle hosts many different security options including encyption at rest and in transit [11].

#### *3.2.2 MySQL*

MySQL is an open source relational data base manager. It is also owned by Oracle like the previous dbms. MySQL has a free community edition as well as payed options which additional support options [12]. The community edition support many common storage formats such as CSV and InnoDB. The free edition also supports stored procedures and triggers [13].

### *3.2.3 Microsoft SQL Server*

The newest version of Microsofts SQL server has added support for running on Linux machines. Microsoft offers many security tools such as encryption when the data is stationary and when it is being manipulated and moved [14]. This dbms comes with many different payment options including some free options. The free versions, however, suffer significantly in the amount of data you can store. JSON, XML, SQL are all development tools supported by Microsoft SQL [15].

## 3.3 Comparison

Comparing dbms options is difficult but they all provide the same core features. Each dbms tries to stand out from the rest by having additional features. Oracle Database stands out with its automated tools and cloud integrations. Microsoft SQL has tried to grab attention by branching out onto other operating systems and offering strong database security. The free versions of each option differ significantly. Microsoft's free option has significant drawbacks on storage size and support. Oracle is much the same way, if not worse. MySQL shines in this department. With it being open source their free version does not suffer significant drawbacks.

## 3.4 Conclusion

For our project we don't need an advanced system or anything flashy. With this is mind we will be going with MySQL's free community edition. It will provide us the tools we need without incurring added costs.

## References

[1] "Xamarin overview," 2017.
[2] "Xamarin test cloud," 2017.
[3] "Corona product," 2017.
[4] "mobile app development products," 2017.
[5] "Modules and developer kits," 2017.
[6] "Jetson tx2 developer kit," 2017.
[7] "Raspberry pi 3 model b," 2017.
[8] "Arduino tian," 2017.
[9] "Oracle database," 2017.
[10] "Oracle database," 2017.
[11] "Oracle database," 2017.
[12] "Mysql editions," 2017.
[13] "Mysql community edition," 2017.
[14] "Compare microsoft sql server versions," 2017.
[15] "Compare sql server 2017 editions," 2017.