

# Kite : A Social Media Journal Application

Project Sponsored By:

David Vasquez

## Technology Review and Implementation Plan

Written by:

Andrew Bowers

Fall 2017 - CS 461

Group 21

### **Abstract:**

This Technology Review and Implementation Plan provides an in-depth look at what technology we plan on using to build the Kite application. It aims to explain the different technology resources we have, their benefits and downfalls, and which we find best to use for the Kite Application.

## **PERSONAL ROLE**

My role in the Kite project has been to research and build the database, research the client suggested mobile platform language, and aid in project documents. Most of my work has revolved around the database: writing out tables and their attributes, drawing relationships, and verifying its integrity with our client and his goals.

## **GOAL**

The Kite project seeks to develop a social media app not unlike Facebook, Instagram, and Twitter. Our group will develop and fill the niche that exists today where social media can only capture a snippet of your day. The Social Media Journal App will allow users to tell more of a story by allowing for extended posts with embedded photos and other multimedia. To accomplish this, we need to select a mobile language platform that will allow us to develop Kite for both Android and iOS, select an IDE that supports this language and mobile platform development, and a robust database option that can support the applications data storage needs and users.

## **TECHNOLOGY 1: DATABASES**

The first database option we'll be discussing is Adminer. Adminer is a PHP designed MySQL database management tool similar to that of phpMyAdmin. Adminer does its best to stand out by defining, in a list format, of how it's a better option than phpMyAdmin. For instance, it lists that it has a better User experience, more supported MySQL features, better performance and security[1]. However, this system does require the user to download a file so they can start database creation etc. This would require everyone on the Kite project to download the same file so they can work on the database. Even though this would be a challenge to work around, Adminer also has quite a bit of documentation giving us more troubleshooting opportunities in case of error or other problems.

The next database technology I looked at was MyWebSQL, an open source MySQL database option. Similar to Adminer, this database system requires the user to download some files to set up the database management tool. However, unlike Adminer and phpMyAdmin, MyWebSQL is open source, like I stated early. The major goal of this option is to give the user more of a streamline desktop experience when working with a database while not sacrificing performance or speed. According to their website, "MyWebSQL is the ultimate desktop replacement for managing your MySQL databases over the web. With interface that works just like your favorite desktop applications...[2]" One of the downsides I've seen in my research is that there appears to be less documentation on how to use this software. Both Adminer and phpMyAdmin have a lot of different resources my group can use in case of troubleshooting.

The final database option I researched was phpMyAdmin. phpMyAdmin, like Adminer and MyWebSQL, is a free web based MySQL database management tool. One of the main differences between phpMyAdmin and the other two database management systems is that, due to Oregon State's relationship with phpMyAdmin, the only thing we have to do to utilize the system is login - no download required (but is an option). Another benefit of phpMyAdmin is its documentation. According to their official website, "phpMyAdmin comes with a wide range of

documentation and users are welcome to update our wiki pages to share ideas and howtos for various operations[3].” However, the user interface/experience offered by phpMyAdmin is lacking, it’s less intuitive and a little clunky, in my opinion. Overall, we have decided to go with phpMyAdmin due to all of us being able to access it anywhere and because of how well documented phpMyAdmin is.

## **TECHNOLOGY 2: MOBILE PLATFORM LANGUAGES**

The first language I looked at was Java. This is the main language used for developing Android applications. This is a very well documented and used mobile platform language with a significant amount of resources and troubleshooting options like the website ‘developer.android.com’. This website itself has a significant amount of resources like tips on how to build your first application or interacting with other applications and more. According to Shalini from Raygain Technology, Java offers a lot of simplicity ”to both end developer and users... Java is platform independent and its removed the usage of pointers as well. The difficulty that we faced with a programming language, Java also removed it by an implementation of traditional language C++ that called interface [4].” Other positive notes that Shalini made about Java was that its safe and secure as well as it being open source. One of the main downfalls for this language is that we would not be able to develop for iOS mobile platforms; we would be limited to only Android users for implementation, testing etc.

Objective-C is the another language I looked into. Another well documented and supported language, Objective-C is the main programming language for iOS platform development. Similar in relation to the amount of resources to Java, Objective-C has an an entire website created by Apple documenting it [5]. According to the Apple Objective-C page, ” It’s a superset of the C programming language and provides object-oriented capabilities...[5].” A main problem of using this programming language is that it would only allow us to develop for iOS, reducing our user base for implementation and testing, similar to problem we would see developing with Java.

The final language I researched was React Native. React Native, the language used to for Facebook, Instagram and more, is a mobile platform language that uses React and Javascript. Similar to the previous two languages, React Native is also well documented, less so than the other two but still noteworthy. According to the React Native home page, ”React Native lets you build mobile apps using only JavaScript. It uses the same design as React, letting you compose a rich mobile UI from declarative components[6].” Although, as I mentioned earlier, React Native does have a noteworthy amount of documentation, it is a newer language which could lead to troubleshooting issues. However, one of the main benefits we have found is that it will allow us to develop for both iOS and Android. Because of its ability to work for both mobile operating systems, being fairly well documented, and being an excellent fit for our project, we have decided to go with React Native.

## **TECHNOLOGY 3: MOBILE INTEGRATED DEVELOPMENT PLATFORMS**

The first Integrated Development Platform, or IDE, I looked at was Nuclide. I wanted to find a IDE that best supported React Native and it, among ATOM, Visual Code, and Sublime were most often noted. According tot he

Nuclide website, "Nuclide is built as a single package on top of Atom to provide hackability and the support of an active community. It provides a first-class development environment for React Native, Hack and Flow Projects[7]." Nuclide offers a built-in debugging tool, support for Javascript development through Flow and more. ATOM, which is what Nuclide is based off of, for the most part its very similar to Nuclide and would be a good alternative to it. However, a possible downside is that it does not include a debugger.

Sublime Text is a text editor that I had found, but is widely used and supports React Native. The React Native Website itself shows the use of this text editor in its hot reloading description[6]. The application has a lot of features like Go To Anything which allows the user "to open files with only a few keystrokes, and instantly jump to symbols, lines or words[8]." The editor also offers a significant amount of customization, split editing, and other useful features. The primary downside would be that it doesn't have a debugger built in, so troubleshooting errors would be more difficult. Another possible problem is that, for the editor to run React Native, we will need a rather large amount of plugins for it to work correctly, which could cause issues getting it up and running.

Visual Studio Code is another editor I took a look at. It does have more debugging features than sublime, but less than Nuclide. The editor does support React Native and works well for mobile application development. However, for the scope of this project, I feel that Nuclide will be the best option.

## **CONCLUSION**

In conclusion, we feel that for the database, phpMyAdmin will offer the most useful tools and resources. React Native will be the mobile language we use due to it's ability to support both Android and iOS. And lastly, I found that Nuclide would be the best IDE to use for this project, given its large support for React Native and Javascript.

## REFERENCES

- [1] Adminer, "Adminer - database management in a single php file."
- [2] MyWebSQL, "Mywebsql web based database administration simplified."
- [3] phpMyAdmin, "phpmyadmin bringing mysql to the web."
- [4] Shalini, "Java app development: Pros and cons?."
- [5] A. Inc, "Introduction."
- [6] Facebook, "React native."
- [7] Nuclide, "A unified developer experience for software development."
- [8] S. Text, "A sophisticated text editor for code, markup and prose."