

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.

1 DATABASE OPTIONS FOR DATA STORAGE

1.1 Overview

In this section I'm going to discuss the different database options I have found for our project. I'm going to look at the pros and cons of each, then pick the one I find best for this project.

1.2 Criteria

The database option should be relatively straight forward and allow all members of my group to be able to access and work on it. For the project we're going to need a secure and efficient system for data storage.

1.3 Potential Choices

The database options I'll be comparing are Adminer, MyWebSQL, and phpMyAdmin.

1.3.1 Adminer

The first database option we'll be discussing is Adminer. Adminer is a PHP designed MySQL database management tool. Adminer does its best to stand out by defining, in a list format, ways in which it is 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.

1.3.2 *MyWebSQL*

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. 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.

1.3.3 *phpMyAdmin*

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.

1.4 Discussion

Each of the three options has a fair amount of documentation, giving us options when it comes to troubleshooting. One of the main downfalls for both Adminer and MyWebSQL is the required file download to operate, whereas with phpMyAdmin we just have to use a specific login to access the database.

1.5 Conclusion

Overall, we have decided to go with phpMyAdmin due to all of us being able to access it anywhere. This will allow all group members to be able to work on it when needed.

2 MOBILE PLATFORM LANGUAGES

2.1 Overview

In this section I will be looking over different languages for mobile application development. The primary objective of this is to look at the pros and cons of good mobile platform languages and find the one that best suits our application.

2.2 Criteria

These languages should be well documented, suitable for mobile application development, and the Kite application.

2.3 Potential Choices

The potential choices for the mobile platform language are: Java, Objective-C, and React Native.

2.3.1 Java

Java 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.

2.3.2 Objective-C

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.

2.3.3 React Native

The final language I researched was React Native. React Native, the language used 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 I have found is that it will allow us to develop for both iOS and Android.

2.4 Discussion

All of the language are used in for mobile platform development, are fairly well documented, and would work well for the Kite application. However, Java only works for Android based applications and Objective-C for iOS applications. This would limit our ability to test on devices as well as limit what users are involved. Whereas React Native supports both iOS and Android mobile platforms.

2.5 Conclusion

In conclusion, because of React Native's ability to work for both mobile operating systems, being fairly well documented, and being an excellent fit for our project, we have decided to use it for our project.

3 USER TESTING

3.1 Overview

The user testing section is going to go over three different methods to test users and get feedback on the Kite application. The feedback we are trying to get will revolve around UI, overall functionality, and usability.

3.2 Criteria

We want the User Testing method to give the test users a fair understanding of the application as well as opportunities to work though the concepts of the application. This will allow them to find problem areas as well as test how usable the application interface is.

3.3 Potential Choices

The user testing choice I will be looking at are Paper Prototyping, Surveys, and Focus Groups.

3.3.1 Paper Prototyping

Paper Prototyping is a method of usability testing in which we, as developers, will create paper mock ups of the application, then have test users practice moving through it as if it were a fully developed application. As described in the book Handbook of Usability Testing:

"In this technique users are shown an aspect of a product on paper and asked questions about it, or asked to respond in other ways. To learn whether the flow of screens or pages that you have planned supports users' expectations, you may mock up pages with paper and pencil on graph paper, or create line drawings or wireframe drawings of screens, pages, or panels, with a version of the page for each state[7]."

This will give us an idea of what users like and don't like as well as find problem areas were moving from one page to another is not as fluid or intuitive as it should be. Another bonus to this method is that it allows us to work out a conceptual model and have some sort of physical representation of what the application will look like.

3.3.2 Surveys

With surveys our group would be able to create a large set of questions based on the application and its interface, then send it out to a large group of people to respond it. We could get a vast amount of questions answered by a large group of people, giving us feedback on how we should move forward with the application and its UI. "While the survey cannot match the focus group in its ability to plumb for in-depth responses and rationale, it can use larger samples to generalize to an entire population[7]." These generalizations could give us a good idea on how to best design the UI moving forward through application development.

3.3.3 Focus Groups

Last, but not least, Focus Groups, as described in Handbook of Usability Testing, "employs the simultaneous involvement of more than one participant, a key factor in differentiating this approach from many other techniques." This option would give us more in depth information on the application concept and how we should move forward in development. In a sense, this method is similar to Paper Prototyping, but the models are less in-depth, and offers the test users pictures or even storyboards of the application interface. The users could then all work together talking about things they like or do not like. Depending on the group(s) of people used in the testing, this method could strengthen our conceptual model of the application giving us a strong foundation for development.

3.4 Discussion

The survey method offers us a lot of generalized information, however, it makes it harder for test users to get a feel of the application and how it will look or work. This will limit their ability to offer stronger information on design and the application's usability. As for Focus groups, this option will allow for more in depth discussion on the application, but due to its group nature it would make it more difficult trying to practice running through the application and finding problem areas. Finally, Paper Prototyping will allow us to give a lot more detail to only one user, as well as give the user a chance 'run through' the prototype of the application.

3.5 Conclusion

In conclusion, I have found that Paper Prototyping is the best option for user testing. It will allow us to show more detailed representations of our application as well as give the test user the freedom to 'move' through the application. This would then give us more feedback on problem areas or things they found confusing. Although Paper Prototyping is the best user testing method for this project, we will also be using Focus Groups for testing in other areas of our application.

REFERENCES

- [1] Adminer. Adminer - database management in a single php file. [Online]. Available: <https://www.adminer.org/en/phpmyadmin/>
- [2] MyWebSQL. Mywebsql web based database administration simplified. [Online]. Available: <http://mywebsql.net/>
- [3] phpMyAdmin. phpmyadmin bringing mysql to the web. [Online]. Available: <https://www.phpmyadmin.net/>
- [4] Shalini. Java app development: Pros and cons? [Online]. Available: <http://www.raygain.com/java-app-development-pros-cons/>
- [5] A. Inc. Introduction. [Online]. Available: <https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>
- [6] Facebook. React native. [Online]. Available: <https://facebook.github.io/react-native/>
- [7] J. Rubin and D. Chisnell. Handbook of usability testing. [Online]. Available: <http://ccftp.scu.edu.cn:8090/Download/efa2417b-08ba-438a-b814-92db3dde0eb6.pdf>