

Tech Review - CS461

Fall2017

Samuel Bonner - Group 29

I. ABSTRACT

This document is Samuel Bonner's rough draft of the Tech Review for the Social Device For Non-Technical Users, or Sharing Station. Three technology pieces are evaluated, including a native development framework, web application framework, and mobile application framework. Evaluated technologies include Cordova, native Android, native iOS for native frameworks, AngularJS, React, and Vue for web frameworks, and Ionic, OnsenUI, and jQuery Mobile for mobile frameworks. A description of the selected technology and why it was selected is provided at the end of each section.

NATIVE DEVELOPMENT FRAMEWORK

A native device framework will be used to provide access to the functionality of the underlying mobile operating system. The web and mobile application frameworks will be built on top of this device framework. In this section, the frameworks Cordova, Native Android, and iOS will be evaluated.

CORDOVA

Cordova is a development framework for mobile applications that allows development of hybrid rather than native applications by providing access to native device functionality. Hybrid applications are cross platform, and are generally targeting Android and iOS operating systems. Cordova depends upon the JDK (Java Development Kit) version 8 or later, Android SDK, and packages for the Android SDK that target the desired API levels.

Pros

Cordova applications are not specific to a single mobile OS, ie Cordova apps may require little work to port to different operating systems. This is useful to the project, as the Sharing Station application could be available for both Android and iOS devices. Cordova features several plugins and frameworks that may make development simpler by already having compatible tools. These frameworks include the web application framework Angular, and the mobile framework Ionic. Cordova applications are developed in JavaScript and HTML, rather than a language specific to the operating system. Several other elements of the project will be developed in these same languages, and integration of the components may be easier as a result. Passing updates to features may be easier in some cases, as web content is viewed dynamically. This would allow updates to be passed faster to views in the project, without having to be approved by a 3rd-party. Several features of the Sharing Station application are already at least partially implemented in a Cordova project. This may give the team more time to focus on performance optimization of the device and revision of the user interface.

Cons

Cordova applications utilize HTML5 and the Android Web View, and will most likely have a slower performance speed than native applications. This performance loss may not be significant, as the planned implementation of the Sharing Station app does not involve heavy use of video or graphics beyond two-way video calling.

NATIVE ANDROID

Native Android applications are developed specifically for devices running the Android operating system. This framework utilizes Java programming unless the NDK (Native Development Kit) is utilized, in which C and C++ libraries may be imported.

Pros

The performance of a native Android application will most likely be faster than hybrid implementations, especially for animated graphics and video streaming. Native Android applications make use of the Java programming language. The team is familiar or experienced with Java and will not need to take additional time to learn the language.

Cons

Developing a native Android application will limit the app to only Android devices. The application will need to be ported to another specific OS, and this will cost more time and money than developing a hybrid application. Passing updates to the application requires resubmission and approval from the publishing channel used. If critical or time sensitive updates are required, this may be an inconvenience to the clients or owner of the application. Choosing to implement the Sharing Station application as a native Android app would require a rewrite of existing features and functionality. This would require additional time and may limit the time partitioned to optimization or revision of performance and user interface aspects.

NATIVE IOS

TBD This section may be replaced with React Native

Pros

TBD

Cons

iOS applications utilize Objective-C and Swift, languages that are not widely used, and in the case of Swift, specific to iOS applications. Extra time may be required by the programming team to learn these languages, as well as to build a new working project. As a native application, additional time and money will have to be spent to port the application to other mobile operating systems. Unlike Cordova and native Android, developing the application for iOS would require selection of another base hardware device than the one currently selected for the project. Like development of a Native Android application, pushing updates may take additional time waiting for approval from the iOS store that published the application.

SELECTED TECHNOLOGY

Cordova is the selected technology for a Development Framework. We selected Cordova because of its portability to both Android and iOS, powerful selection of plugins and tools (Angular+Ionic), and lowest time required to develop and optimize components.

REFERENCES LIST

<https://cordova.apache.org/docs/en/latest/guide/platforms/android/>
<https://developer.android.com/ndk/guides/index.html>
<http://mubaloo.com/cordova-vs-native-apps/>
<https://developer.android.com/reference/android/webkit/WebView.html>
<https://www.smashingmagazine.com/2013/10/best-of-both-worlds-mixing-html5-native-code/>
<https://www.androidauthority.com/developing-for-android-vs-ios-697304/>

WEB APPLICATION FRAMEWORK

The web application framework will be used to build the Sharing Station web application. The web application framework should assist development by providing a library of functions, CSS, and screen elements the development team can modify and implement. The framework may extend HTML and JavaScript syntax to provide additional functionality and simplify DOM and AJAX manipulation. The web application frameworks evaluated will be AngularJS, React, and Vue.

ANGULAR

AngularJS is a JavaScript library that serves as a framework for developing web applications, with an emphasis on creating dynamic rather than static views, and simplifying DOM manipulation. Angular extends HTML syntax with abstracted DOM manipulation, form binding, and other extensions.

Pros

Easily integrates with Cordova projects, as Angular is targeted at development with Cordova. Shares a similar structure with Ionic applications, and can be used in conjunction with Ionic implementations. Can be used to simplify development of dynamic webpages and elements, by abstracting DOM and AJAX programming. Extends relatively simple HTML with additional functionality. Features that have already been implemented for the Sharing Station project make use of Angular. Choosing to continue to use Angular would keep the web app implementation consistent.

Cons

Reduces the flexibility of dynamic webpages, as implementations will be limited to the Angular model. Will be difficult to implement stretch goal features such as two-way interactive games, such as a dynamic drawing game over a network. Heavy abstraction contributes to a larger learning curve for Angular, and more time may be necessary to learn the library. May suffer from lower performance speed than React and Vue.

REACT

React is a framework for developing web applications that is used to simplify the development of interactive user interfaces. React is a JavaScript library that abstracts DOM elements to JavaScript objects, and website components to JavaScript functions. React extends JavaScript syntax by using a language called JSX.

Pros

React syntax is similar to regular JavaScript, utilizing the fewest abstractions compared to Angular and Vue. Provides the greatest flexibility of the other frameworks, as all JavaScript functionality is available rather than HTML abstractions. More diverse tools may also be available as a result.

Cons

React makes heavy use of JavaScript rather than HTML, resulting in more complexity compared to its competitors if members of the development team are not experienced developing web applications with JavaScript. Optimizing performance may be more difficult and require more intimate knowledge of the library, due to the extensive functionality.

VUE

Vue is a lightweight framework and JavaScript library for developing dynamic user interfaces. Vue uses a template model with HTML syntax to abstract DOM manipulation. Vue provides visual transition tools utilizing both CSS and JavaScript animation libraries.

Pros

Vue is the smallest web development framework, and less time may be required to learn the library and develop with it. Test builds may be very quick to develop with Vue, permitting more time to be spent evaluating the design and performance of the resulting application. May provide a more flexible application design than AngularJS.

Cons

Vue is the newest and least used mobile app framework, and will have the least extensions, examples, and tools associated with it. This may significantly limit the flexibility of the framework.

SELECTED TECHNOLOGY

We chose to use Angular for the Sharing Station web development framework. Using Angular for future elements of the project will keep the code base consistent and limit how many additional libraries are loaded and used. In addition, the development team will be able to model implementations off existing Angular models in the project, reducing the time required to develop and modify aspects of the Sharing Station web application.

REFERENCES LIST

<https://docs.angularjs.org/guide/introduction>
<https://reactjs.org/docs/react-api.html>
<https://vuejs.org/v2/guide/index.html>

MOBILE FRAMEWORK

The mobile framework will be used to build the Sharing Station mobile application and potentially the Sharing Station Social mobile application. Mobile frameworks should provide similar functionality to the web application frameworks, however they must target mobile device layouts rather than web applications. The mobile framework chosen should provide the capabilities to implement mobile device views that are simple and fluid for the target elderly, non-technical audience. The mobile frameworks evaluated include Ionic, OnsenUI, and jQuery Mobile.

IONIC

Ionic is an open source framework for developing the front end of mobile applications. Ionic applications are a version of Angular that is targeting mobile development rather than solely web development. When built on top of Cordova, Ionic is used to provide visuals and interactive functionality that models the operation of a native application.

Pros

Easily integrates with Cordova projects and Angular frameworks. Ionic development will be similar to the Angular web application development, as Ionic is a derivative of Angular. This may reduce the amount of time the development team spends learning tools and frameworks. Open source software, no cost associated with choosing Ionic.

Cons

Alike the cons for Angular, development using Ionic may limit the flexibility of the mobile app implementation. As a derivative of Angular, Ionic is highly abstracted and the development team will not be familiar with it. Ionic may take longer for the team to learn, however this may be reduced somewhat by the use of Angular, a similarly structured framework.

ONSENUI

OnsenUI is an open source JavaScript library that targets mobile app development. OnsenUI provides a variety of HTML, CSS, and JavaScript extensions that aim to make mobile app development faster.

Pros

Provides support for several web development frameworks, including Angular, React, and Vue. Support is provided in the form of framework bindings. Web components are written in native JavaScript, and will most likely be compatible with other components. Provides premade CSS and UI elements that are simple to modify. Screen elements are scalable, which may resolve current issues with the Sharing Station application. Onsen is also open source software, with no extra cost required.

Cons

Premade assets may not be appropriate to the project, as the Sharing Station application may use non-standard UI elements. OnsenUI is not as widely used as the other frameworks evaluated in this document. Less tools may exist within Onsen than older and more popular frameworks like jQuery Mobile and Ionic.

JQUERY MOBILE

jQuery Mobile is an open source mobile app development framework built on jQuery. jQuery Mobile supports animated page transitions, custom themes, and scalable screen elements.

Pros

jQuery Mobile requires knowledge of jQuery rather than AngularJS like Ionic. Members of the development team familiar with jQuery may need less time to begin development. jQuery Mobile has a large set of premade customizable themes. These themes may be quick and easy to adopt for the project.

Cons

jQuery Mobile does not make use of AngularJS, which was selected previously in the document. Using jQuery for the mobile development when the rest of the application uses JavaScript does not make sense without considerable advantages.

SELECTED TECHNOLOGY

The selected mobile framework for the Sharing Station project is Ionic, as it is structurally similar to Angular, and existing components of the mobile application already use Ionic.

REFERENCES

<https://ionicframework.com/docs/intro/concepts/>
<https://onsen.io/v2/guide/architecture.html#architecture>
<http://demos.jquerymobile.com/1.4.5/intro/>