

Resemblance of PhoneGap and Titanium for Mobile Application Development

Jigna Solanky
Teaching Assistant Uka
Tarsadia University,
Bardoli, Gujarat

Kajal Patil
Teaching Assistant Uka
Tarsadia University Bardoli,
Gujarat

Gayatri Patel
Teaching Assistant Uka
Tarsadia University Bardoli,
Gujarat

ABSTRACT

MAD (Mobile Application Development) is becoming more demanding because of their varied platform and software development tools. To reduce the cost and in order to reach the application to maximum users across various platforms, the developers are emigrating towards the cross platform application development tools. In this paper, we provide distinct benchmark beyond the portability for choosing appropriate cross platform tools for application development. Cross platform mobile app development is mobile based apps development. So, mobile application development for this type of application can be made platform independent. An analysis has been made in this topic while thinking about IDE is being offered which will help a software developer to develop an application in a single code. Deploy that single code to multiple operating systems, i.e. if the developer can write code in java for android, same code can be deploy to ios system. A review has been made in this paper for comparison and contrast about two frameworks PhoneGap and Titanium for cross platform application development.

Keywords

Cross platform tools, mobile application development, native apps, hybrid apps, PhoneGap, Appcelerator, and Titanium

1. INTRODUCTION

Nowadays, the mobile application development is growing towards the peak. Today each and every person is having smart phone in their pocket because it provides a large range of functions such as Media Player, Cameras, Bluetooth, WI-Fi, video calling and GPS with advanced competence. There is an increasing insistence of mobile applications to use mobile devices, but the development of mobile application for mobile devices is not so easy and it's a big challenge. These mobile devices have different operating systems and unlike PC OS, mobile OS determines the type of programming language required for applications running on particular OS that's why we need to design the mobile application congruent to its platform. There are various categories of mobile application development broadly classified into three i.e. native apps, mobile web apps and hybrid apps. According to the application classification and platforms, we have different preferences in programming language and framework for mobile application development. To run an application on different platforms, a cross-platform framework is required. The open source framework Titanium and PhoneGap gives the opportunity to the developer to develop the mobile application with their skills. Both framework works differently.

With the variation in devices (like iPhone, Android, Blackberry, Windows Mobile Phone), in order to provide above referenced kits for application development to its developer, an application suitable for the heterogeneous

platform environment is needed. We have discussed the Literature Review of cross platform mobile application

development in II, varied framework required for application development advisable for heterogeneous platform mobile applications are discussed in section III, Comparison between titanium and PhoneGap in IV.

2. LITERATURE REVIEW

Following are the reviews of the literature related to the cross platform application development:

Niyigen Jean Pierre and Mukiza Octavien [2] proposed a review of PhoneGap APIs accessing the native mobile platform APIs. They have described how PhoneGap APIs and how JavaScript is used to access mobile native device. PhoneGap APIs is an application that is mostly used by the developers because just a few lines of JavaScript is used to access mobile device. Sometimes PhoneGap does not support native feature but a native feature is needed for an application and also it supports Plug-in for particular native feature. So in that case Plug-in for particular native feature are written for that particular platform i.e. In android written in java and in ios written in objective C.

Isabelle Dalmasso, Soumya Kanti Datta, Christian Bonnet, Navid Nikaein[3] have concentrated on cross platform tools and also described the general requirement of cross platform framework and its general architecture. A detail survey of the tools is also presented which allow developer to gain insight about the tools. For Android Application the performance is measured in terms of memory, CPU usage and power consumption whereas the application written in PhoneGap is found to be minimum memory, CPU usage and power consumption. They have also reported that PhoneGap with Sench Touch 2.0 works well when memory is not an issue.

Chaitanya Kaul, Saurav Verma [4] have found that CIDER helped them to analyse the entire ios and the Android ecosystem which they could basically port some of the functionality of ios platform on the Android subsystem. Also this helped them to convert research problem to mathematical model which worked on the functionalities of IDE.

The mobile app that is needed by the business must first look at the mobile app approach, which provides the simplest and cost effective mobile app option [10]. Businesses that require full device integration or app store distribution must consider hybrid approach which gives the easy path for "native" features. And finally businesses that require game-like graphic for their mobile app must consider the native approach which provides the powerful graphical abilities.

C. S. Patil, R. R. Karhe, M. A. Aher [5] has discussed on mobile development using various technologies and the generation of telecommunication- 1G, 2G, 3G, and 4G. 1G

gave start-up for cellular- wireless concept. 2G provides different features for users. 3G provides different attractive multimedia functions. 4G provides more data rate, high scale use, and less transfer power.

CHARKAOUI Salma, BEN LAHMAR El habib, ABDELBAKI Issam and MARZAK Abdelaziz [6] have discussed on platform fragmentation, application types, and cross platform approaches. The selection of platform is depends on three different elements – the programming tricks, importance of application that appears native, model based approach. MDA- Model Driven Application reacts well with above elements.

Spyros Xanthopoulos and Stelios Xinogalos [7] have talked about various approaches and application. Native mobile apps are developed for specific platform and offer without doubt best user experience. Source code of native mobile app is quick performance, steady look and feel and full access to platform hardware. Current trend in developing cross platform mobile applications focus on four app types - hybrid, web, generated and interpreted apps. Generated app approach is best reliable. It is best solution after native apps. Development process is based on domain specific language and set of code generators. Hybrid and interpreted app promising cross platform development for generic apps. Interpreted app is best option for native look and feel. Development environment for interpreted apps is Titanium. For hybrid apps JQuery library and PhoneGap is dominant environment.

Nabil Litayem, Bhawna Dhupia, Sadia Rubab [8] have contrasted various platforms, and with the help of quizzes and assignment application example they came up with solution that hybrid application is best suitable solution for mobile application development. They suggested that instead of re-implementation of applications with same functions, first compare strength of each platform. So that compatible and stable app can be developed.

James Brocklehurst [9] has compared Titanium and PhoneGap. Titanium provides robust solutions with closer to natively authored apps but with less platform supports. Titanium is better choice for ios and Android app development. PhoneGap is more suitable for Symbian, Web OS, and VIM with very less access to hardware.

Niyigena Jean Pierre, Gakwaya Daniel, Fan Xiumei, and Gombaniro Jean claude [18] have discussed on cross-platform mobile geolocation applications based on PhoneGap. Cross-platform development framework PhoneGap is invented to solve the problems of different programming languages that developers faces challenges while building application for various mobile platforms. Geolocation API of PhoneGap is supported famous mobile operating systems like ios, android, blackberry, WebOS, Symbian and Windows Phone. This API uses HTML, CSS and JavaScript for finding location of mobile devices.

2.1 Categories of mobile application

Regardless of the type of mobile platform there are three main categories of mobile application as in which generally we develop the applications. Based on the application requirement, we determine the category adequate to our requirements [8]. Firstly, let us discuss about all of them in detail.

2.1.1 Native Application

Native app is been developed with a single code for all devices and are specific to the mobile operating system.

Single project is developed for each Operating System which leads to a raise in development team, cost and time. For the developer the new challenge continually appear in the market. If you are working in the environment where there is no internet connectivity, native app is the best way to go for it. This native app works with the device which is built-in feature and thus they are easier and faster to work on the device. The application which is developed using this category is platform independent. Thus whatever language you use, you will have the full access to Integrated Development Environments which is preferable tool to develop and helps to debug the project faster. Example: Angry Bird.

Beyond advantage of using native app, there are also few disadvantages. Maintenance is one of its which is the difficult task for both user and developers. Developers have to develop the application according to different platform whereas the users have to upgrade it regularly. Another disadvantage is the development cost when you are making application for different platform. Sometimes it is very difficult for developer to maintain and provide support as the user may use different mobile of different version.

2.1.2 Web Application

Web application works on desktop computers as well as on mobiles. In case of responsive web application, the design has been decided by the server and according to the device feature it is applied at the client side renders according to the device features. Whereas mobile web application which furnishes superior utilization as contrast to responsive web application, its content, provided solely for mobile devices due to there is a necessity to maintain variant sites for each mobile device. A user does not require going to mobile application store to upgrade or download the apps. Thus whenever the users will log in, they will acquire the upgraded version. Developers also does not required to worry about the mobile platform because there will be the single ubiquitous version which can be used by any of the mobile platform. That's why; the maintenance cost of the web application is low.

Excerpt advantage of using web application, there are also few disadvantages. Internet connectivity must be there in web application. Web applications are not coherent with smart phone features like GPS, Camera, and Phone dialling etc. Web applications even cannot be listed in play store. If user wants to search the application then have to search on web. Performance wise web applications are slow as compared to native apps. If user wants to use the web application on regular basis, then have to mark it as bookmark. Without bookmark it is difficult to use web applications. No application icon is there on user's device because it is a web link which can be open as per requirement. As a publisher or developer you not able to send notification to them.

2.1.3 Hybrid Application

Hybrid application combines variant technologies from native and mobile Web application to obtain the benefits of each technology. Application are installed from a web store and have right of entry to device specific features as available in native application but application is developed using web app tools that's why they act correctly like a native application. Hybrid application tools can modify pre-packaged HTML pages, able to change user interface coherent to device platform and allow both offline and online usage. Hybrid mobile applications can be delivered on various platforms using accurate web technologies like CSS3, HTML5 and JavaScript. It will save the elevated cost and time used to assemble software for each and every platform. Example of

hybrid applications are Netflix application, Face book, Tune in Radio, LinkedIn which runs the same code based on all platforms.

Beyond the advantages, there are few disadvantages of using hybrid application development. Hybrid applications are not executed natively because the JavaScript and HTML5 portion of the application is produced and executed by the platform's web engine, which insert another layer between the application and the user. That's why execution of the application becomes slower. Hybrid application development is new technique so there are fewer tools available for the development of hybrid application. Applications with complex native functionality or bulky transitions, such as 3D games have sometimes performance issue.

2.2 Mobile Platforms

There are various operating systems that are used in mobile market like ios, android, blackberry and window's phone [1]. Mobile application developments in a few of them are discussed in detail.

2.2.1 ios

To develop an application for iPhone, ipad and iPod touch, you will require Mac computer running OS X 10.5.8 or later is needed. iPhone SDK furnished by Apple includes IDE of Xcode, iPhone simulator and a suite of excerpt tools for the development of an application. App Store supply user facility to search and download applications which are developed by ios SDK.

2.2.2 Android

Android OS is released by open source apache license and built on linux. Android application can be built using linux, windows, or Mac and java is primary language for development of Android application. A java class recompiled in Dalvik byte code and runs on DVM. DVM stands for Dalvik Virtual Machine. Android does not support J2ME, and its most ordinarily used is Eclipse and Android Studio. Developers can create native libraries in C or C++. The Google play store is the official site and portal for searching and downloading android application.

2.2.3 Windows Mobile

Windows mobile OS gives more desktop like user experience. In addition to C# or C++, Silverlight and Microsoft XNA game studio are used for game and application development. For the development of windows mobile application required tools are windows mobile SDK, visual studio .NET framework and developer toolkit.

2.2.4 Blackberry

Blackberry platform supports various ways of developing application:

- a) Web development with HTML, CSS and JavaScript.
- b) Java application development using Mobile Information Device Profile (MIDP), Connected Limited Device Configuration (CLDC) and RIM proprietary APIs. Development of Blackberry application in java supports 32 bit windows operating system.

2.3 Development languages of application

Preferring the correct tools for development is very critical and key decision. Development tools play key role in

successful adoption of new technology. Strong tools help to make experience developer more productive and new developers can easily get started with application development. This title encapsulates the mostly application development languages available for variant platforms with their continuity to cross platform mobile applications development [19].

2.3.1 JavaScript :

JavaScript is deliberated along with CSS and HTML for application development. JavaScript is mounted with each platform. iPhone and Android have browser with JavaScript which supports full features and only some of the versions of JavaScript are supported with BlackBerry. Thus the code for platform specific browser features of one device can't deal with other browser features of HTML and CSS.

2.3.2 Objective C :

Primary programming language for Apple ios, offered with Objective C, which provides object oriented stuffs, need dynamic runtime and tide purely on MAC OS. XCode contain interface and instrument for developing ios app. Objective C class is compulsory for iPhone app development with MVC software architectural pattern.

2.3.3 HTML 5 :

HTML-5 Markup Language, advanced version of HTML with latest features. With reduced differences between hybrid and native, HTML and JavaScript executed on system, results in same structure as native apps. It gives knowledge oriented information respected to offline storage, multimedia, network, error-handling, threads and sockets, advanced form control, drawing animation. Up to 5 MB stuff can be saved user of app. Both iPhone and Android deliberate application developed in HTML5.

2.3.4 Java :

Tremendous programming language that comes with Mobile application is JAVA. Plenty of users are using JAVA as it is easy and tutorials are accessible without efforts. Development tool – Android SDK – It contains Java Libraries such that Math, Graphics, Network, and Data Structure. Library helps to design good Android Apps. Java can deal with Hybrid apps, Web apps, and Native apps. Among various languages, Java provides platform independent app with Object-Oriented Programming environment.

2.3.5 jQuery Mobile :

From the word jQuery Mobile, the first question may be arising what's that? It's a touch optimized framework that can be used to produce responsive apps and sites with use of HTML5 based user interface and that are available on tablet, smart phone and personal computer. Instead of coding one app for each mobile OS, JQuery mobile provides standard for all mobile browsers. It uses simple JavaScript, CSS and HTML.

To terminate this discussion we come up with an idea that there are different languages used for cross-platform application development. According to report of developer research 15% of mobile developers are targeting browser while 42 % are using JavaScript, HTML and CSS [8]. Whereas Java and Objective C are also very favored among developers.

Development Languages Support For Varinat Mobile Platform

Development Languages	Mobile Platform			
	ios	Android	Windows Phones	BlackBerry
JavaScript	Full browser support	Full browser support	Depend on browser and Webkit	Partial support
JAVA	Duke Scripts	Native Eclipse IDE	Java CLDC EMU	Sun JDK J2ME platform Eclipse IDE
Objective C	Native Xcode IDE	-	-	-
HTML5	Full Webkit based browser	Full Webkit based browser	Depend on hardware UI IE Browser	Browser dependent
jQuery Mobile	Full support	Full support	Full support	Full support

3. FRAMEWORK FOR CROSS PLATFORM MOBILE APPLICATION DEVELOPMENT

In the preceding section we have talk through various programming languages used for development of mobile applications. These languages usage percentage depends upon

the mobile application category and device platform backing by the specific target language. Despite the fact that an application can be built without a framework, but with the framework we can acquire a cross platform application with an easy development and deployment process. The given table abbreviated some of the frameworks used for mobile apps

Table 1. Mobile Application Development Framework

Parameters	Framework		
	PhoneGap	Titanium	jQuery Mobile + PhoneGap
APIs and Documentation	<ul style="list-style-type: none"> - HTML, JS, CSS API. - Library wrapping recourses. - Documentation for proprietary API. 	<ul style="list-style-type: none"> - Rich API. - Low level objects. - Object customizable using JavaScript API. - No HTML and CSS coding. 	<ul style="list-style-type: none"> - Build with top APIs. - Offers libraries and components.
Environment Dev/Build	<ul style="list-style-type: none"> - Eclipse IDE 	<ul style="list-style-type: none"> - IDE Titanium Studio 	<ul style="list-style-type: none"> - Uses Imported CSS.
Deployment	<ul style="list-style-type: none"> - Binary file cannot be published. 	<ul style="list-style-type: none"> -- 	<ul style="list-style-type: none"> --
Framework Stability	<ul style="list-style-type: none"> - Mature framework. 	<ul style="list-style-type: none"> - Mature framework. 	<ul style="list-style-type: none"> --

Advantages	<ul style="list-style-type: none"> – Source code available, easy to customize. – Easy to adopt. – Works with HTML, CSS and JavaScript. – Free to combine PhoneGap with other tools. 	<ul style="list-style-type: none"> – Outputs on Mobile device very quickly and easily. – Easy setup. – Supports tablet app development. 	--
Disadvantages	<ul style="list-style-type: none"> – Not supports dev tools, components of Native UI, design patterns. 	<ul style="list-style-type: none"> – Less no of phone supported. – Finite APIs. 	
	A PhoneGap app is coded in HTML5 and execute in a native container.	A Titanium app is coded in JavaScript, use native controls and compiles in to native apps.	-
	Developers can use CSS, JavaScript and HTML5 also.	It is not an HTML5 app run in a web container.	
	UI elements may be produced using jQuery mobile or some other JavaScript web mobile development framework.	This tool can furnish better performance and better user experience by design.	

With this adumbration about variant frameworks mentioned in above table, we can make a differentiation of their usage across operating systems (mobile platforms) and can give a layout of the development process according to the application category. It is essential that developers must choose software that provides an ordinary development track across variant platforms when application is developed with cross-platform mobile frameworks. In this title we talk over briefly on some of the popular cross-platform frameworks chosen from above table, focusing on application development process.

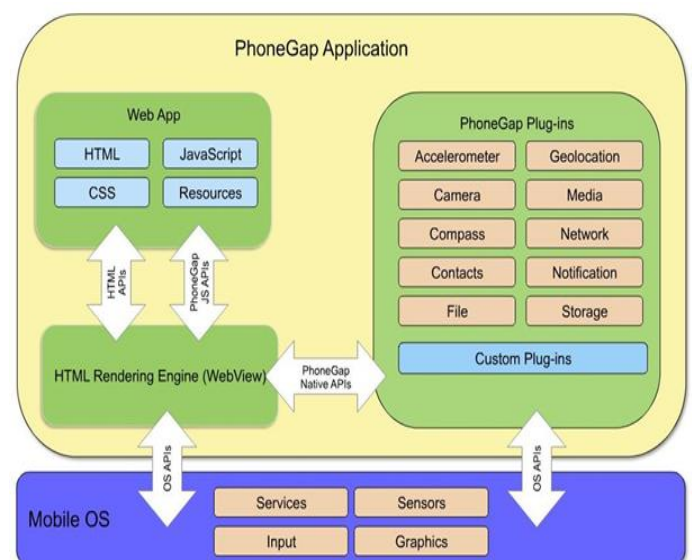
3.1 PhoneGap

PhoneGap is a free of cost and open source framework that gives you to make mobile applications with standard web APIs of the platforms. To create an app for iPhone, we need a computer with Mac OS X. PhoneGapLib is a static library that permits users to cover PhoneGap in their iPhone app projects. We can also make new PhoneGap-based iPhone app projects using an Xcode template. Xcode is environment for Apple’s development and Mac OS X that covers the iPhone SDK potentials. In Android, a developer requires to install the Android SDK and ADT plug-in for the Eclipse. ADT expands the potentials of Eclipse to build Android projects and APKs to distribute applications. Development in Blackberry device requires Eclipse 3.4 or 3.4.1. With this developer has to install BlackBerry JDE Plug-in for Eclipse, and BlackBerry JDE v4.6.1Component Pack for Eclipse Software Update.

Working of PhoneGap is easy. Application which developed in PhoneGap when launched, web application start up page first runs (which is mostly index.html but developer can change to any other page as per their conveniences) in to the web view and then web view allows the user to collaborate with the web application. As the user collaborates with the

web application’s content, JavaScript Code or Hyperlinks within the application can load content from resource file packaged within this application or can loads the content from web application server.

PhoneGap architecture is shown in below fig.



3.1.1 Usage Of PhoneGap API

This section describes most of the PhoneGap APIs and use of that APIs.

3.1.1.1 Accelerometer:

This API is used for capturing device motion in X, Y and Z direction. Accelerometer.getCurrentAcceleration() method is

used to get current acceleration along with x, y and z direction. Accelerometer.watchAcceleration() method is used to keep track of interval distances relatively to the current position. Accelerometer.clearWatch() is used for interrupting interval tracking.

3.1.1.2 Vibration

This PhoneGap API gives a way for vibrating device. In this API notification.vibrate() will vibrate the device with specified amount of time. For vibrating the device a JavaScript code is navigator.notification.vibrate(time), here time is a number which represent the amount of time for the device to vibrate. Time is defined in milliseconds.

3.1.1.3 Network Information (connection) :

Network connection is accessible through navigator.connection object which is capable to determine the device's network status that is connected or disconnected and type of connection that is Wi-Fi, 2G, 3G, 4G, and Ethernet etc.

Navigator.connection.type is used for checking the network status.

3.1.1.4 Contacts :

This API provides access to the device contact database. For creating contact contact.create() method is used and for finding contact from database contact.find() method is used. Navigator.contacts.create(properties) will returns a new contact object.

3.1.1.5 File-Transfer :

This PhoneGap API is used for reading, writing and navigating file system hierarchies. The content of file are written and read by default as UTF-8 encoding but can be read as base64 or text.

3.1.1.6 Camera :

This API provides access to the device's default camera application. Camera.getPicture() method is used to take photo using device default camera or can retrieves a photo from image gallery of device. It returns base64 encoded string or as the URI of image file for further processing. Camera.cleanup() method is used for removing intermediate photos taken by the device camera from temporary storage of image. It used as navigator.camera.cleanup(onSuccess,onFailure).

3.1.1.7 Capture :

This capture API of phoneGap provides grant to access the device's audio, image and video capture capabilities. capture.captureAudio() method is used for launching device's audio recording application for recording audio clips. Device user can capture multiple recordings in single session. It is used as navigator.device.capture.captureAudio(captureSuccess, captureFailure,{limit:3}) where limit is maximum no of audio clips can be recorded in a single capture operation. Capture.captureImage() method is used for device's camera application for taking photos and more than one photos can be captured in single session. Capture.captureVideo() method is used for launching the device's video recorder application for recording videos and allows user to record more than one video in single session. We can specify limit and duration property for above captureAudio and captureVideo methods. Limit is for maximum no of audio/image can be capture in single session. Duration is for maximum duration of audio clip in seconds.

3.1.1.8 Device:

This device object API is used for getting device's hardware and software information. Device.model – used for getting device model

Device.cordova–used for getting cordova version running on device.

Device.platform-used for getting device's operating system name.

Device.uuid-used for getting device's UUID(Universally Unique Identifier)

Device.version-used for getting operating system version of device.

3.1.1.9 Compass:

This API is used for obtaining direction that device is pointing. Compass.getCurrentHeading() method is used for getting the current compass heading. Compass is a sensor which detects heading or direction that device is pointing. It used as navigator.compass.getCurrentHeading(success, failure, compassoption). Compass.WatchHeading() method is used for getting compass heading in degrees at a regular interval. Compass.clearWatch() is used for stop watching compass referenced by the WatchID parameter.

3.1.1.10 Events:

PhoneGap platform provides a complicated event based integration with native and JavaScript. Native Code loading anticipate JavaScript event. Eventually native code fires deviceready event for loading DOM and make available to JavaScript. Event control of application is achieved by adding Event listeners with document.addEventListener() method. Most commonly used events are: pause, deviceready, resume, backButton, menubutton, searchbutton, startcallbutton, endcallbutton, volumeupbutton, volumedownbutton, offline and online.

3.1.1.11 GeoLocation :

The GeoLocation object API provides access to the location based on the device's GPS sensor or inferred from network signals such as IP address, RFID, Wi-Fi, Bluetooth MAC address and GSM/CDMA cell Ids. It provides information about the device's location, such as latitude and longitude.

Geolocation.getCurrentPosition() returns device's current position as a Position object.

Geolocation.WatchPosition() is used for watching the changes of the device's current position.

Geolocation.clearWatch() is used for stop watching modifications to the device's location referenced by the watchID parameter.

3.1.1.12 Globalization :

Globalization gives information and performs operations precise to the user's locale and timezone. Globalization object is the child of navigator object so have global scope.

Globalization.getPreferredLanguage() gives the string identifier for the client's current language.

Globalization.getLocaleName() gives the string identifier for the client's current locale string.

Globalization.dateToString() returns a date formatted as a string as per the client's locale and timezone.

Globalization.stringToDate() is parses a date formatted as a string, as per the client's calendar and user preferences using the time zone of the client, and returns the corresponding date object.

Globalization.getDatePattern() is returns a pattern string to format and parse dates according to the client's user preferences.

Globalization.getDateNames() method returns collection of names of the months or week days, according to the client's user preferences and calendar.

Globalization.isDayLightSavingsTime() announce whether daylight savings time is in effect for a particular date using the client's calendar and time zone.

Globalization.getFirstDayOfWeek() gives first day of week according to the client's user preferences and calendar.

Globalization.numberToString() gives a number formatted as a string as per the client's user preferences.

Globalization.stringToNumber() determines a number formatted as a string as per the client's user preferences and returns a corresponding number.

Globalization.getNumberPattern() gives a pattern string to format and parse numbers according to the client's user preferences.

Globalization.getCurrencyPattern() gives a pattern string to format and parse currency values as per the client's user preferences and ISO 4217 currency code.

3.1.1.13 InApp Browser:

The InAppBrowser API is web browser view that will display when we call window.open() method or when opening any link which contains

addEventListener() method is used for adding a listener for an event from the InAppBrowser API.

removeEventListener() method is used for removing a listener for an event from the InAppBrowser.

Close() method is used for closing the InAppBrowser window.

insertCSS() method is used for injecting CSS to InAppBrowser window.

executeScript() method is used for injecting JavaScript code to InAppBrowser window.

3.1.1.14 Media :

Media API provides the ability to record and play audio files from device.

Media.getCurrentPosition() method will return current position within an audio file.

Media.pause() method will pause playing audio file.

Media.getDuration() method will returns the duration of an audio file.

Media.play() method will starts or resume playing an audio file.

Media.release() method will release the operating system's audio resources.

Media.seekTo() method will sets the current position within an audio file.

Media.setVolume() method is used for setting volume for an audio file.

Media.startRecord() method will start recording audio file.

Media.stop() method is used for stop playing an audio file.

Media.stopRecord() method is used for stop recording an audio file.

3.1.1.15 Notification :

Notification API will provides visual, audible and tactile device notifications.

Notification.alert() method shows custom alert box.

Notification.confirm() method shows customizable confirmation dialog box.

Notification.prompt() method shows customizable prompt dialog box.

Notification.beep() method is used for playing beep sound in device.

Notification.vibrate() method is used for vibrating the device for specified amount of time.

3.1.1.16 SplashScreen :

This API is used for showing and hiding application start up screen when system is busy doing something.

Splashscreen.show() method is used for showing splash screen.

Splashscreen.hide() method is used for dismissing splash screen.

3.1 Titanium

Development in native application for mobile environment is done using Titanium. It composed of an SDK with compiler, tools and APIs for building the target platform and for managing development visual environment is used. It makes use of web technologies that are trendy and powerful both, involving HTML5, AJAX, jQuery and CSS3. Titanium is convenient with Windows, Mac and Linux. Titanium framework can make applications feature-rich with a platform-independent API because it can access advanced features like touch screens, cameras, storage, navigation, GPS, contacts and many more. Titanium also keep enlarge features like Record Video, Shake and Screenshot.

Titanium is cross platform JavaScript runtime and API framework which is used to do mobile application development in ios, android and in future it may be possible that they provide support for Windows and Blackberry. It is similar to PhoneGap because they both provide cross-platform approach using web technologies.

When writing application in titanium, actually you are writing application in native application using JavaScript. Titanium framework adopting a unified JavaScript API, with platform specific functionalities and native performance combined.

While using titanium as framework two major points needs to be understand that are as follows:

1. Core features in mobile application development can have same experience across all platforms and you can utilize code reuse.

2. With more platform specific APIs, UI and features, developers should use specific platform code. So it's not "one time write code and runs that code everywhere" approach.

Because of above two points, titanium offers great user experience by providing access to native features and native UI to achieve higher level of performance than other most hybrid approaches.

Titanium takes code which is written in JavaScript, combines this code with the Titanium API which is written in the device's native language and assess your code at runtime with a JavaScript interpreter that runs on the device's operating system. When application is launched, a JavaScript execution is created in the native code and JavaScript is inserted as an inline object paired with native objects within the runtime environment. This pairing is closely one to one in a way like running inline JavaScript within HTML. This pairing allows the Titanium API to step as a link to expose direct access to the native experience, through actual native features and APIs. Titanium doesn't use web view component approach which is used by other JavaScript toolkit and PhoneGap. JavaScript which is written in titanium framework is not cross-compiled into the corresponding native language. JavaScript code is evaluated at runtime. Instead of replacing this through CSS and JavaScript toolkits, titanium framework provides animations and native UI controls.

Advantages of using titanium are access to a wider and higher level of APIs, native UI and device specific functionalities. Titanium offers same User interface and components that are available to both android and ios developers, which is not provided by any other hybrid frameworks. Visual emulation via CSS, Flash and OpenGL, Titanium offers platform's backend UI controllers, behaviors and animations which provides native experience that are mostly expected by user. You also able to extend these visual aspects and other core features by extending titanium API as per the need of your application. Main benefit of using titanium is to develop native application with JavaScript, instead of having to learn platform specific languages. Once application is completed, titanium provides automated way to package and

Distribute your applications to the platform's respective app store through an IDE called Titanium Studio.

One of the major disadvantages of using titanium as framework is that it still doesn't provide as wide support as most of people hope for today. The main reason behind that is the fact that titanium API connects directly to the native platform and taking up a new platform has lot of work. And because of this reason titanium only supports the larger platforms in android, ios, blackberry and some level of mobile web browser support. Also some of the UI components don't perform as well as their native counterparts. Titanium can never support all native APIs and device functionality, which continue to gain and develop on the native platforms per each new release and update to the native operating systems. Titanium can tries to offer at least 90 percent of these APIs and native functionalities on its supported platforms.

Titanium offers:

1. It provides solution to develop mobile applications currently on ios, android and blackberry without learning the program code individually for each and every individual platform.
2. A titanium platform that is able to access native controls

and device features directly and provides a better user experience by offering behavior and animations which user expects. With titanium framework application is written in JavaScript. Instead of manipulating an HTML DOM, as would be done in PhoneGap, JavaScript is interacting with Titanium API. Titanium provides UI objects for things like text fields, buttons, and lists and so on. In many cases, code you write for one platform can run unmodified on other platform as well. As such, with Appcelerator, at least some parts of your code will have to be written specifically for each platform.

Appcelerator consists of suite of products, including titanium SDK for developing mobile applications, and for the back end of the application some cloud services that can be used. Eclipse based IDE named Titanium studio is provided by Appcelerator. This Integrated Development Environment contains a JavaScript editing environment, along with a source-level debugger.

Alloy framework of titanium allows screen layout to be written either in JavaScript or in an XML. Alloy application framework grant you to specify the User Interface of an application in XML and employ styles to the controls in a project via a CSS-like syntax.

3.1 PhoneGap + jQuery Mobile

jQuery Mobile is a light-weight API and development of any application carrying web services is quite easy.

jQuery Mobile is built on top of jQuery API and provides a JavaScript library of CSS and other components. Excellent documentation of the mobile API allows developers to familiarize with the tools quickly. The environment uses exotic CSS and JavaScript file for a project and builds as any regular android application. jQuery Mobile + PhoneGap are mature framework.

4. COMPARISON AND CONTRAST OF TITANIUM AND PHONEGAP

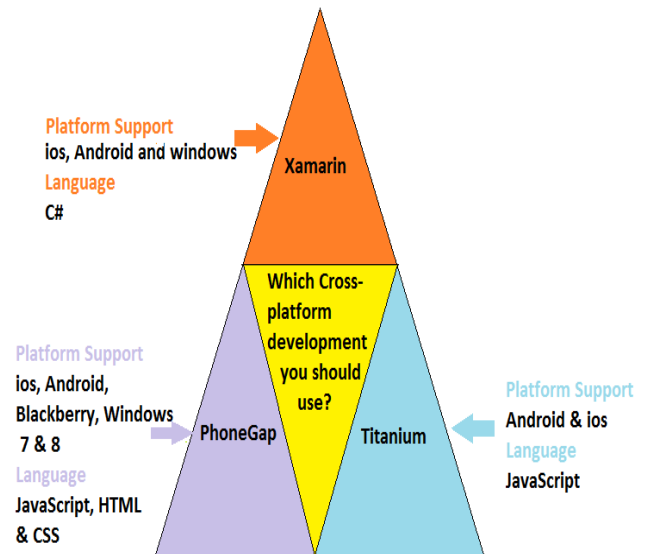
PhoneGap	Titanium
PhoneGap is open source and simplest open source framework compared to Titanium.	Titanium is open source and complex framework compared to phoneGap.
PhoneGap application is simply a set of HTML Pages which is rendered as a Web view. For developing application you need to use CSS, HTML and JavaScript.	Titanium is bit more complicated because it desires to determine the UI API for each platform on top of JavaScript which again is composite for building large application.
Supports all platforms and operating systems which include IOS, Android, Windows Phone 8, BlackBerry, Firefox OS and Ubuntu.	Currently it supports Android and IOS.
It supports third-party libraries.	It does not support third-party libraries.
Lower performance of application as the original codes of the application remains that of a web application and launches via a web browser. That means the performance of PhoneGap application does not come close to native apps.	Performance is better because of native API usage, which also provides grants to access the features and elements of IOS and Android.
User interface of application varies depending on the quality	The look and feel of titanium application is

of web view rendered.	better rather than the applications which are built on other platforms as the User Interface is essentially native.
It uses HTML5 and CSS, the animations and DOM elements are more responsive.	It does not use HTML5 and CSS, the animations and DOM elements are less responsive.

PhoneGap and Titanium	
PhoneGap and Titanium both provide tools for cross-platform mobile application development.	
They both need to use Web technologies and JavaScript in some capacity for mobile application development.	
Titanium and PhoneGap both are Open Source Software	

5. CONCLUSION

Titanium seems to offer a more powerful solution because results are much closer to a native application, but titanium has limited platform support. If you have a specific requirement for WebOS, VIM or Symbian devices, then PhoneGap is only choice, despite the fact that it may not give you access to all of the device hardware that you might need. If you only interested in ios and Android devices, then Titanium appears to be better choice than PhoneGap. From above study, we have learned that there are some frameworks which are suitable for hybrid cross- platform mobile application with each of them having some advantages and disadvantages. With respect to avoid re-implementation of the same applications for various platforms individually, it is necessary to compare and make use of the vitality of individual framework or the combination of frameworks, so that; more adaptable and more reliable cross-platform application can be developed. Studies also showing that there are many current systems that are not being scrutinized fully and will possibly have a golden future if some more functionality is studied and proper implementation is carried out. Therefore, in our future work we will continue the studying of the latest cross-platform framework, device features available by these frameworks, and implement a new system incorporating new features. Hence, this study is important for a cross-platform mobile learning application development.



6. REFERENCES

- [1] <http://www.shoutmeloud.com/top-mobile-os-overview.html> [30-1-2016]
- [2] Niyigen Jean Pierre and Mukiza Octavien, "Review of PhoneGap APIs Accessing the Native Mobile Platform APIs", Lecture Notes on Software Engineering, Vol. 4, No. 1, February 2016.
- [3] Isabelle Dalmasso, Soumya Kanti Datta, Christian Bonnet and Navid Nikaein, "Survey, Comparison and Evaluation of Cross Platform Mobile Application Development Tools".
- [4] Chaitanya Kaul and Saurav Verma, "A Review paper on cross platform mobile application development IDE", IOSR Journal of computer Engineering (IOSR-JCE).e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 17, Issue 1.
- [5] C.S.Patil, R.R.Karhe and M.A.Aher,"Review on Generations in Mobile Cellular Technology", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, volume 2, Issue 10, October 2012
- [6] CHARKAOUI Salma, BEN LAHMAR EL habib, ABDELBAKI Issam and MARZAK Abdelaziz,"Towards a Multi-Platform Development Based on MDA approach", International Journal of Computer networks and communications security, ISSN 2410-0595,VOL. NO 3, March 2015.
- [7] Spyros Xanthopoulos and Stelios Xinogalos, "A comparative Analysis of cross-platform development approaches for Mobile applications".
- [8] Nabil Litayem, Bhawna Dhupia and Sadia Rubab, "Review of cross-platforms for mobile learning application development", International Journal of Advanced Computer Science and Applications, Vol.6, No.1, 2015
- [9] James Brocklehurst, "A Brief comparison of titanium and phoneGap for cross platform Mobile Development".

- [10] Native mobile apps: The Wrong choice for business?
- [11] http://docs.phonegap.com/en/edge/cordova_plugins_plugins_inapis.md.html#Plugin%20APIs [01-02-2016]
- [12] <http://docs.appcelerator.com/platform/latest/#!/api/Titanium.Android.ActionBar> [25-03-2016]
- [13] <http://docs.appcelerator.com/titanium/latest/#!/api> [28-03-2016]
- [14] <http://www.cygnet-infotech.com/blog/phonegap-or-titanium-or-xamarin-which-cross-platform-should-you-choose> [30-3-2016]
- [15] <http://forumone.com/blog/what-titanium-appcelerator-really-and-how-it-works/> [01-04-2016]
- [16] <https://www.smashingmagazine.com/2014/03/4-ways-build-mobile-application-part4-appcelerator-titanium/> [05-04-2016]
- [17] <http://www.appcelerator.com/blog/2012/05/comparing-titanium-and-phonegap/> [07-04-2016]
- [18] Niyigena Jean Pierre, Fan Xiumei, Gakwaya Daniel and Gombaniro Jean Claude, “Cross-Platform Mobile Geolocation Applications Based on PhoneGap”, Lecture Notes on Software Engineering, Vol. 3, No. 2, May 2015.
- [19] <http://www.androidauthority.com/want-develop-android-apps-languages-learn-391008/> [06-03-2016]