



ISSN : 2350-0743

www.ijramr.com



International Journal of Recent Advances in Multidisciplinary Research

Vol. 04, Issue 12, pp.3121-3130, December, 2017

RESEARCH ARTICLE

HYBRID MOBILE APPLICATION BASED ON IONIC FRAMEWORK TECHNOLOGIES

¹Bakwa D. Dunka, ²Dr. Edim A. Emmanuel and ³Dantala O. Oyeyinka

¹Research Scholar/Programmer, ICT Directorate, University of Jos, Plateau State, Nigeria

²Senior lecturer, Department of Computer Sciences, University of Calabar, Cross River State, Nigeria

³Department of Computer Sciences, University of Jos, Plateau State, Nigeria

ARTICLE INFO

Article History:

Received 07th September, 2017

Received in revised form

21st October, 2017

Accepted 19th November, 2017

Published online 30th December, 2017

Keywords:

Ionic, Apache Cordova,
Phone-Gap, Ionic Framework,
Hybrid mobile application,
Leaflet, AngularJS,
Web Application.

ABSTRACT

The proliferation of mobile technologies has brought about a unified communication sphere along with verse and modern experiences. These emerging technologies have intensified competition within the mobile communication industry. This competition has further led to some problems for the users as well as application developers. Developers are not able to access and modify mobile applications codes that operate on other platforms i.e. Native platform incompatibilities. In this study, we advance the academic notion of developing a hybrid mobile application into a practical perspective. The same code base for these application promises a simple yet timely approach to multiple platforms. The result of the study addresses the problems inherent with vendor specific platform dependency and parallel developer's expertise. A hybrid mobile application was designed using emerging Ionic framework and leaflet to ensure its practicality on and roid, iOS and windows platforms. The need to adapt Ionic framework for developing hybrid mobile applications will offers simplicity and flexibility in user communication experience. It will be a potential to break the monopoly of native dominant technologies.

INTRODUCTION

The theory of Hybrid Mobile Application is the ability to take Web technologies such as HTML5, CSS and JavaScript (JS) and sum them up on a Native Mobile application platform. Mobile platforms on their own are basically developed using three different approaches namely 1) The Native platform approach 2) The Web platform approach and 3) The Hybrid platform approach. Each approach has its peculiarities as well as its advantages and disadvantages. Therefore, there is no one approach that may be absolutely perfect for mobile application development (Minh, Prashant and Donny, 2017). What is predominantly in use today is the Native mobile approach. However, Native platforms have limited interactivity with other platforms due to their parallel independence. They often have credible performance and also maintain good user interface and fluidity. This is because they are designed to have direct access to the phones own hardware and operating system's features. (James, 2017) shows that the greatest smartphone which presently takes on the mobile market are and roid, iOS and Windows phones. We describe the technical ingredients of these mobile platforms in Fig. 1. In the diagram, The Internet Operating System (iOS) was created with C, Swift, or Objective C, and roid applications are essentially built in JAVA. However, its underlying kernel is written mostly in C and C++ because it needs to communicate to the hardware and drivers.

*Corresponding author: Bakwa D. Dunka

Research Scholar/Programmer, ICT Directorate, University of Jos, Plateau State, Nigeria.

Windows phones are built with Visual basic or the .NET framework. Aside the native platforms, The web platforms are similar to web applications deployed on computers, but they are normally accessed through the mobile web browser. There are three core technologies used in developing web applications, they include 1) HTML5 "for images and text" (Gawley, Barr and Barr, 2012), 2) CSS "for its representational style" and 3) JavaScript "for interactions and logic". A Hybrid mobile application is seen as a consequential derivative from heterogeneous sources consisting of Native and Web app. This is necessary in order to have a unified operating platform which can work on a mobile web browser to increase its limits, diversity in style and the ability to access eminent Native features such as, motion sensors, accelerometer, GPS, Camera and File systems (Spachoset *al.*, 2014; Huynh, Ghimire and Truong, 2017). The hybrid mobile application is designed for push notification, data-driven capabilities, interactivity, browser previewing, Cost effectiveness and timely benefits (Tanque, 2017). Hybrid mobile applications can be configured on a Play store or Google store where they can function just like any other native application. Node.JS is a modern external development component that provides the mobile application with the backend requirement. This need to be incorporated into the application to perform functions such as listening to network request on the mobile device where the app runs, provide services, provide routing activities, centralized data storage and management of users. The backend is executed on one or more remote servers to listen to network requests from the devices that run the app, and provide them with the service

requested (Buttigieg and Jevdjenic, 2015). iOS apps require OSX IDE to function properly while Android requires SDK and Xcodes at the same time utilizing their hardware and software resources. A hybrid mobile application can also be viewed in its final form without having to compile for any specific platform such as iOS or Android. In this study, we center our design on a new hybrid mobile application frontend framework "Ionic". This framework is designed for developers to easily build more interactive hybrid mobile applications on top of the existing mobile application components developed with native apps (Weiß, 2016). The technology has a relatively low learning curve (Phan and Hartington, 2015), and can be combined with other web technologies such as HTML5, CSS3 and JavaScript. In the works of Griffith (2017), the platform is also built on an ecosystem that comprises of AngularJS serving as the web application framework and Apache Cordova for building and packaging the Native application. The mobile operating system also plays a role in loading applications, managing installation of apps that are downloaded from a market store and also provides a set of APIs for accessing features such as GPS location, contacts list, camera etc.

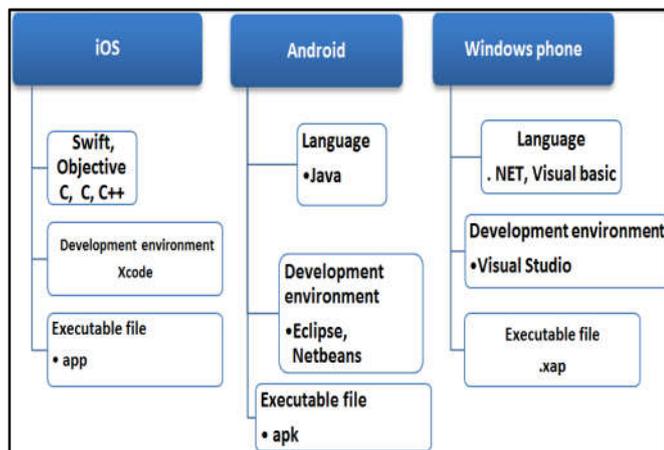


Fig. 1. Native mobile development environments

Not all mobile application built nowadays requires all the features of a typical native application. Yusuf, (2016) describe the need to have an application which can run cross-platform and also adapt the native features of the selected platform in order to support real-time web and mobile communication.

In this study, an application was designed, implemented and tested using a combination of these technologies to demonstrate the powers of Ionic framework for easy creation of hybrid mobile applications. The application is in itself not the major aim of this study. But, it was designed and developed to confirm that larger and innovative applications can practically be built and deployed using this powerful tool on mobile platforms. Christ (2011), described the fact that Native applications are been developed on the components own API. Therefore, the problem of a separate code base for each API is apparent. This, in its self-result to expensive development and time consumption when expert are hired for some particular platforms. Therefore, instead of coding for a single platform, and encountering more overhead in terms of time (developing for web or other platforms), cost (of hiring developers of different platforms) and difficulty for programmers to mention a few, A hybrid mobile application is a better choice to imposing flexibility while maintaining standards. The author breakdown how a Hybrid mobile application streamlines the

Native and Web processes of building and deploying mobile applications. According to Ramanujam, Srinivasan and Natili, (2015), the well-known Phonegap, whose name was renamed to Apache Cordova was an alternative prior to the Ionic framework. It was basically used to wrap an existing application in the native layer then give direct access to the native API layer (Fern and ez and Alber, 2015; Natili, 2013; Shotts, 2016). Apache is still in use today but it takes advantage of the improvement of recent mobile memory, power, CPU and the improvement in web standards to improve functionality and looks and feel of mobile devices. The problem prior to Hybrid mobile applications also poses a limitation of diversity in style of featured data-driven development. In the works of Fern and ez and Alber (2015), Phonegap attempts to enhance the features of a common interface and provide a uniform Operating platform. However, it has a comparable less interactivity. This study provides a platform for creating a hybrid mobile application that will increase developer's creativity and add new mobile features on the native mobile platforms as well as creating adequate user experience for common users. Some great advantages of Native SDKs are common UI, views, navigations and stack history, transitions, and gesture. But since there are no web SDKs, there is an urgent and important need to bridge the gap between these Native features and the web platform. The alternative choice is the use of a separate code base.

Objectives of the Study

- To analyze Native, Web and Hybrid mobile approaches to apps development.
- To demonstrate how Hybrid mobile apps can be built using Ionic framework technologies.

Literature Review

Various hybrid mobile application have been developed using various technologies to date. However, there are no general and perfect approaches for creating mobile applications in spite of plethora of mobile applications in the market place today. Also, one eminent problem is that of developing mobile applications that can run on different platforms using the same code base (Dave, 2017). Software Development Kits (SDK) were often used as the most familiar approaches, yet that was the major development technique which capitalizes on native programming languages such as Java, C*, .NET and their individual SDKs. These methods rely on individual platforms such as Windows, iOS, and Android, Blackberry and others for implementation. Now, with the emergence of hybrid mobile applications, The native method is gradually giving way. Hybrid mobile application uses the philosophies or frameworks for mobile development (Salehet al, 2016). Various frameworks are available for mobile development as explained in our literature but we will focus on one framework in this paper.

The approach used in this research is the adoption of the Ionic framework as a modern and emerging approach to proffer solutions to the cross platform mobile app development problems (Griffith and Chris, 2016; Yusuf, 2016). In our perspective, the solution proffered in this paper is expected to be simple yet flexible. We focus on the technology not the features of the application in view. The historical progression in development started with Web design approach, Native app approach then Hybrid mobile app approach.

Methods for web content delivery on mobile devices

Several researches have shown the importance of smartphones to the lives of individuals. This ranges from Job search, health information, educational information, politics and daily news, real time navigational directions, social media communication just to name a few. These diverse and billions of resources and activities carried out on the internet are the main reasons why people are comfortable and dependable on mobile devices (Spachoset *al.*, 2014). One of the early approaches in development of mobile applications was that developers design and develop websites for individual platforms separate from the desktop versions. That way, programmers use various programming languages and coding methods. Users can in-turn enter the URL of the website in question. The advantages of the website approach is that users will gain excellent and friendly user experience (UX) when using their respective devices. This approach is characterized with less display of images, simpler form of navigation and fewer important content on the devices. Over time of using this approach, some problems became eminent. A developer develops two versions of the website, one for mobile and the other for desktop. This approach was later discovered to 1) Increase the overall cost and effort of development. 2) Maintaining several separate websites. 3) How to identifying the various mobile devices available for implementation in the store and market place. 4) The need for constant update of these applications is necessary as the tools for development are being upgraded.

The limitation of the former approach has led to the advancement of these separate mobile sites into one that delivers the same piece of codes on any type of hardware device used. This is regarded as “responsiveness”. Therefore, responsiveness in this perspective mean the ability for the programming components such as PHP, HTML, CSS, JavaScript to finally adjust itself to the various hardware screen such as (Tablet, Mini Tablets, Laptops, Desktop, Smartphones etc.). The advantages of Responsive applications are 1) to avoid creating different versions of the same website running on various hardware 2) Easier maintenance 3) Simplicity 4) Cost saving and lower development time. Advancement in mobile technology industry is not yet exhaustive; the next in line is the native mobile applications. The approach in the development of this types places more emphasis on portable devices rather than desktop computers in terms of their operation, this is not unconnected with the importance of portable devices and smartphone on individual h and s. Native applications are very much in use today and are meant to do most critical activities. However, they are developed using native codes such as Java, C++, .NET framework etc. having separate native JavaScript APIs() for these individual platforms for Windows, Google, Blackberry or Apple which helps in accessing their native features (Lubke, Schuster and Schill, 2012). The drawbacks of the native approach are as follows 1) each platform uses its own SDK this makes it cumbersome to develop applications which can correctly run on other platforms and also have access to their sensors, data, geo-locations. These API are different for each device. The singular factor of having different API created competition in the mobile industries, which invariably mean the bigger the enterprise the less it competes among other mobile industry. An alternative to the native approach is the up-coming promising hybrid mobile approach. The hybrid mobile development approach takes an entirely different dimension. In this approach developers write a single code

with HTML, CSS and JavaScript that will reliably run on other mobile platforms Wilken and Bradley, (2016). Hybrid mobile development approach has now grown in recent years to replace the Native approach. A hybrid mobile application exhibit these characteristics 1) It can run in the mobile browser by having access to the browser shell 2) It run native by leveraging on Apache Cordova (native wrapper) to have access to the native platforms capabilities. The proportion of web and native features gives Hybrid mobile applications its name. Its codes are compiled, could be executed into binary, stored, packages, distributed and downloaded across mobile platforms using various devices. This makes it interesting to develop an app without the use of Java, C* or the .NET native technologies (Wilken and Bradley, 2016).

Hybrid mobile Application Framework

Ionic

Ionic is built with Cordova, it means, itcontains the Cordova file structure to package HTML/CSS/JavaScript into applications that can run on both mobile and desktop devices. It also provides a plugin architecture that is used in accessing Native features beyond what JS can achieve on a web browser. Ionic utilizes the CSS extensive property called Syntactically Awesome Style Sheets (SASS). This is used optionally by developers to define a value once and use it in multiple places within a style sheet document. This way, it enables faster creation of different Interfaces for rich mobile interactive applications (Drifty.co, 2016).

Mobile Angular user interface

(Jscrambler, 2017), described a composition of crucial components such as JavaScript, AngularJS directives, click.JS, overthrow.JS and Bootstrap3 to provides functions such as sidebars, scroll bars, switches, overlays, sidebars, absolute positioning for top and bottom navbars. These components work together to achieve interactive mobile user interface (UI) and responsive media queries for mobile apps.

Xamarin

Xamarinis a hybrid mobile app development framework, which solves the dilemmas that programmers are faced with when developing cross-platform mobile apps: separate coding languages and UI paradigms. Xamarin enables developers to use C# for developing iOS, and roid and Windows applications. It supports timely and simplified process of creating forms, interface design using its XAML-based framework. Walt (2016).Xamarin was founded in May, (2011) by Microsoft-owned San Francisco, a California-based software company founded which has cross-platform implementations of the Common Language Infrastructure (CLI) and Common Language Specifications (often called Microsoft .NET). Developers can use Xamarin tools to write native and roid, iOS, and Windows application with native user interfaces and share code across multiple platforms.

Intel XDK

Intel XDK is a cross platform application tool developed by Intel Company. It uses a drag and drop approach for development. It includes web services and plugins for content-rich, interactive apps, responsive apps runs on any device. Intel

SDK supports development, emulation, testing, debugging and publishing. Recently, they made some big changes by supporting all the capabilities such as developing mobile HTML5 and Apache Cordova for Android, iOS, Windows 10 UAP, including software development capabilities for Node.JS-based, on-board, IoT apps too. Intel XDK allows development via a drag and drop approach and also live preview on the connected device while development on going-on. (Walt, 2016).

Appcelerator Titanium

Appcelerator Titanium is a cross platform mobile development tool that uses open source standards. It exposes the developers to develop applications that will run on tablets, PCs, smartphones and desktops. Titanium Mobile SDK comes with its own IDE. Titanium Studio is similar to Eclipse IDE used in writing and testing applications with the aid of the Android Emulator. Appcelerator Titanium supports database, media, geo-location, contacts, notification and many other native features of a smartphone. Titanium enables web developers to create native mobile, desktop, and tablet applications using open web technologies such as JavaScript, HTML and CSS. Streaming an audio clip can take between 50- 500 lines of code of Objective-C or Java, in Appcelerator Titanium a developer can simply write one line of JavaScript to point to a URL location to start clip streaming.

Sencha touch

Sencha Touch is a HTML 5 mobile app framework for several platforms including iOS, Android, BlackBerry and Windows. It is popular and more comfortable among hybrid mobile applications developers. Sencha provides a native look and feel across all of the platforms it supports for development.

Kendo UI

Telerik's Kendo UI is a HTML 5 framework of hybrid apps, which is used for creating cross platform mobile application. It relies heavily on JQuery based widgets. Learning Kendo UI is not difficult, developers who are familiar with the JQuery will find the Kendo UI is easy to learn. Kendo UI's toolset and JavaScript framework features are open source but its commonly used widgets are still under a commercial license.

Phone Gap

This is the most popular among all platforms. It is used for packaging and releasing an app based on Cordova implementation. Once written, codes are ready for implementation, but PhoneGap takes it from there and wraps it into the mobile platform of developers' choice. Application built using PhoneGap uses a web-view to render its content. PhoneGap has a minimal set of web APIs that supports having access to the phone hardware utilities. It also accommodates writing codes in form of custom plug-ins to suit basic requirements.

System overview and Architecture

The architecture of the system is depicted in Fig. 2. When users open their mobile application using their various devices, Cordova application wrapper is loaded based on the device features. The wrapper contains a webview called index.html.

With the help of a JavaScript API, the frontend engine such as Angular.JS facilitates the bootstrapping process and determines the default view for users. Ionic components which are also part of the web app frontend help to render the UI of the application.

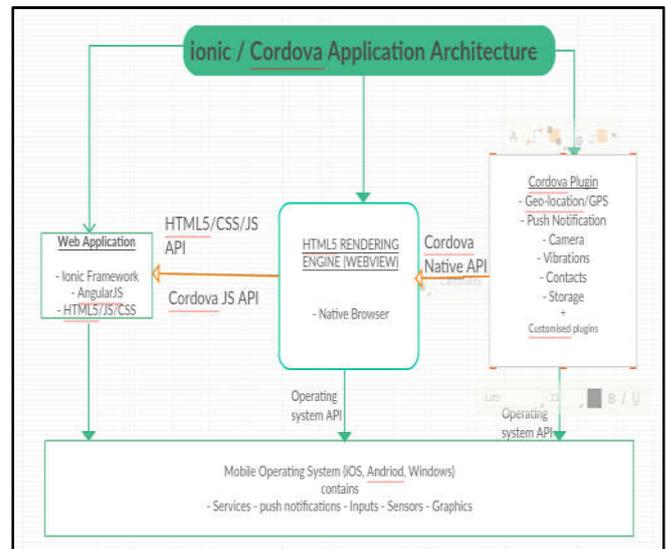


Fig. 2. Architecture of the proposed system

The diagram above shows the technical components of the Ionic architecture. Since the communications between the operating system is platform dependent and slow in its building process, the Cordova JavaScript API has become an important component standing between web applications and the devices. It has the ability to run HTML, CSS, and JavaScript as a substitute for platform specific APIs. (Oehlman and Blanc, 2011) describe Cordova as a bridge that communicates between the web app and the device, the app wrapper has access to both the web application and the native platform through the JavaScript API. This is primarily handled behind the scene and Cordova ultimately generates the native app. This is necessary in order to run native functionalities such as Geo-location, Camera, storage, contact, vibration etc. (Anibal, 2017). Web Application cannot run such functionalities except with the help of Apache Cordova. One component of Cordova that is useful is the Cordova app wrapper that helps to load the web application code.

Angular: Angular is a very popular framework for building powerful web applications. The well-known Angular web application runs inside of a container known as the WebView. Angular is primarily used to manage the web application's logic and data. Angular provides components, data binding, services, forms, directives, http, dependency injections and forms.

Ionic: Ionic is built on top of Angular to provide the user interface, and is primarily used to design an interactive and responsive user interface (Dave, 2016). This includes the visual elements such as tabs, buttons, and navigation headers. These interface controls are the heart of Ionic, and provide a near-native interface inside of a hybrid app. Ionic also includes a number of additional utilities and features that help manage your app, this ranges from creation, previewing to deployment. The combination of these technologies makes Ionic a very feature-rich platform and stands as the foundation for building reliable mobile applications. Ionic's approach is

beneficial in this design because of the evolving mobile industry; the speed of change for any one of these technologies has a lot to offer for businesses to investing in creating their own applications.

Ionic Framework: There are various hybrid mobile application frameworks, some of these frameworks are explained in our literature. However, in this paper, we adopted the use of Ionic framework due to its popularity, simplicity and maturity. Ionic provides the ability for developer to develop the frontend User Interface (UI) that caters for the general representation, interactions, animations, looks and feel and beauty of the application. This framework has a low learning curve compared to others and the ability to write once and run anywhere (Huynh & Ghimire, 2017). Ionic is more of an ecosystem than a framework because it supports programmers with all the tools needed to build mobile applications. It consists of various packages for different actions. For example,

- Ionic Creator is a platform used in building a user interface by dragging and dropping ready-made components. It was implemented in designing the final appearance of the application.
- Ionic View makes it very easy for users to run and share developed apps to multiple devices using simple comm and before they are transferred to the official market store
- Ionic Market is used in sharing the mobile applications, charging a fee to download it. In addition,
- Ionic has credible user support along with documentation
- for useful information.
- Ionic lab for testing newly developed application on and roid and IOS

Ionic resource provides a documentation to showcase how easily one can learn and grasp it use on mobile platforms.

Approaches to developing Hybrid Mobile Applications

There are three approaches to developing mobile application, the Native, Web and Hybrid mobile approaches. This is further described in the works of Anibal (2017) in Fig. 3.

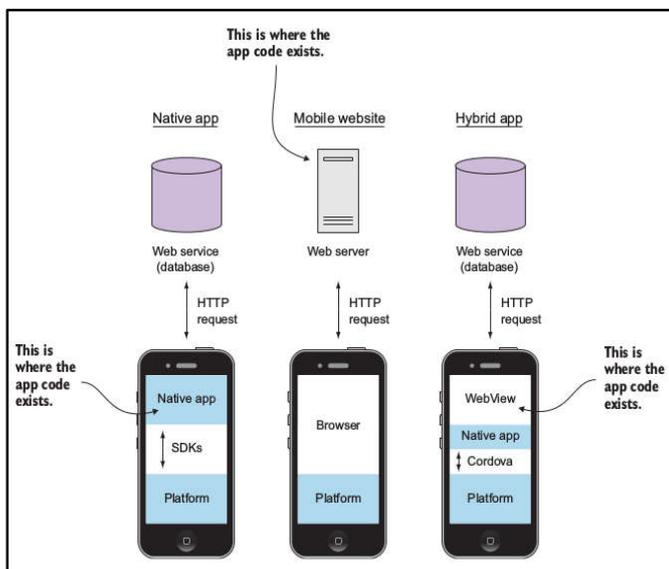


Fig. 3. Native Approach to development Adapted from (Anibal, 2017)

In the diagram, The Native approach is platform dependent. This means that development for individual platforms are essentially built based on native programming languages. Therefore, they can only be run on their various platforms. Applications developed based on Native languages usually have credible performance. The dominance of and roid, iOS and Window in the mobile market give these platforms a large market shares. After each build, they are stored and distributed via their respective market place app store where customer can have access to them. The Mobile web applications are not installed on the device rather they are stored on remote servers and delivered over the Internet and accessed on the device. They are similar to websites that run on the computer. Hence the have considerably low level of performance when dealing with certain major functions. The mobile web apps are distributed via the webs.

A combination of web and native mobile technologies give rise to Hybrid mobile app. This means that the existence of a web-view developed with JavaScript, HTML and CSS is wrapped up within any of the native container. The wrapped web application infused within the shell of any native platform prepares it for storage, portability, installation and market distribution. Xxx related the uniqueness of every platform to business needs, therefore no one platform is said to be better or superior to the other. However, the hybrid application has more benefits in terms of customers cost saving as well as a great relieve for developer in reusability of codes, time saving in production, lowered time of production, lowered cost of development and maintenance, learning and writing or modifying multiple codes for different platforms, availability of a web version of the mobile application. The benefit extends to the users. Users of the hybrid mobile application do not need to spend to acquire different mobile platform i.e. having and roid, iOS and Windows phones. Mobile versions of Facebook and LinkedIn were written using hybrid technologies.

MVVM framework

This framework is relatively new compared to the well-known MVC framework. The MVVM architecture is increasingly recognized with mobile app development. Ionic framework applications are designed to work with the AngularJS MVVM architectural pattern. Thus we cannot disconnect the function of both. This is because Ionic is widely used for developing rich SPA for server-side web apps and also gives liberty to a developer to choose its architectural pattern. AngularJS extends the MVC framework (Stanko, 2016), beyond the powers of Controller. In his assessment, Views should be bound to ViewModel to extend its properties and function for more flexibility. In his work, he demonstrated the power of the ModelView as an extention in functionality to just the View. The Model - keeps data, which will be displayed based on the implementation pattern and business logic.

The View h and les user output and input. The Controller is a manager, which maintains the various views. It also supports operation on user inputs unlike the conventional MVC Patterns, which h and les predominantly the business logic; the MVVC implements the business logic in the model. Also in this case Angular.JS is helpful in two-way data-binding interaction with APIs and back-ends services. We explain the MVVC architecture in Fig 4.

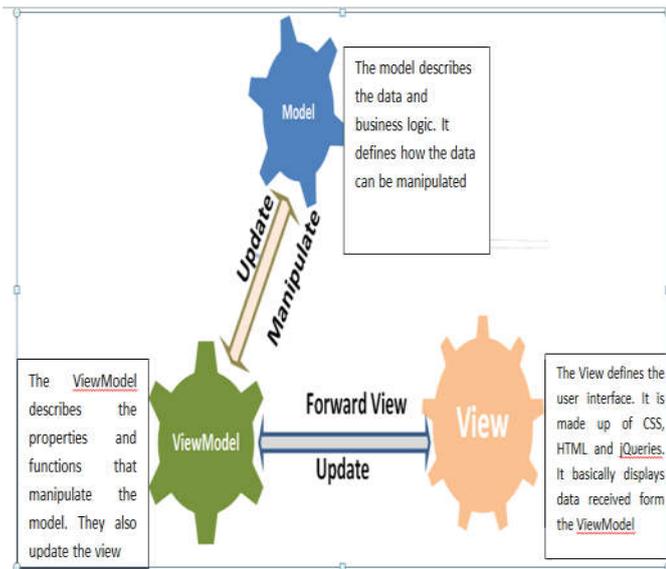


Fig. 4. The ModelView Architectural pattern

MATERIALS AND METHODS

This section aim at providing details of the research process we followed to create a basic hybrid mobile app.

Application Design flow

We start by explaining the processes and methods that were implemented in the design of the system. The application flow is described in Fig. 5. From the diagram, the structure includes the process of design, building, testing and running the system.

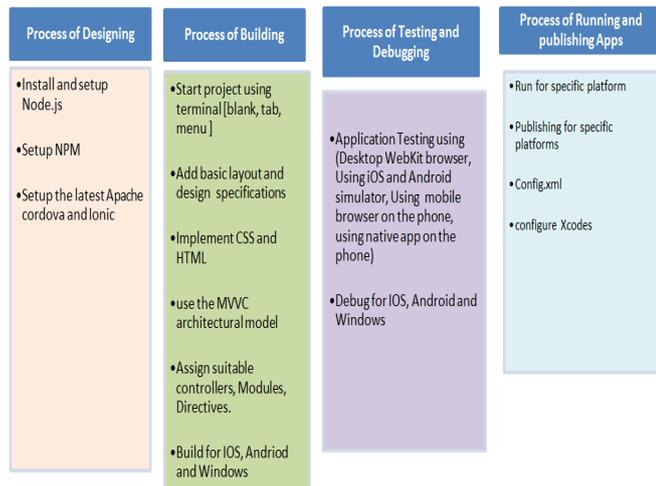


Fig. 5. Proposed system design structure

Setting up the working environment

The first step in setting up our development environment was to setup ionic and AngularJS on both windows and MacOS Operating systems. We installed node-v8.9.1-x64.msi Node.JS installer for windows and node-v8.9.1-x64.pkg Node.JS installer for macOS. Apache Cordova 7.1 for both windows and macOS, Ionic-angular 3.9.2 was also installed. This study also considers specific configuration for platform such as 1) and roid: The JDK installer. 2) Windows: The windows SDK was needed for development across the windows platform. 3) iOS: the macOS computer was used for the development. An emulator was used for testing the project. Testing for iOS on

the phone is seemingly expensive and would require a license fee per year.

Installation of components

The steps needed to carry out installation of the components are considerably easier than other mobile approaches. Node.JS was installed from their official website (<https://nodejs.org/en/download/>). The .msi and .pkgfile were installed on windows and macOS respectively. We check to see if Node.JS was installed correctly, this was done on the terminal and Comm and prompt utility of both OS by typing 'node -v'. This would print the version number. Node Package Manager (NPM) is a package manager or code library that extends the functionality of Node.JS by adding useful features. These comm and s were implemented in order to carry on the process of installation and building. We followed the general processes as follows:

Installing Ionic

```
Install node
Npm install -g cordova
Npm install -g ionic
Orrun
Sudonpm install -g ionic cordova
```

To create mobile App project

```
ionic start
ionic start "app_name" "template"
```

To run the project

```
run ionic serve
ionic serve -l (l denote ionic lab to display and roid, windows and iOS on the browser)
```

To add and remove mobile platform – iOS or and roid

```
ionic platform add "platform name"
Or Cordova platform add "platform name"
(Platform: iOS, and roid)
ionic platform rm "platform name"
Or Cordova platform rm "platform name"
```

Build platform

```
ionic build "platform name"
Or Cordova build "platform name"
```

Run Mobile platforms

```
ionic run "platform"
```

List all platforms

```
ionic platform list
```

Add and remove plugin

```
ionic plugin add "plugin package name"
Cordova plugin add "plugin package name"
ionic plugin rm "package name"
```

To run existing project

```
run: npm install
bower install, gulp install ( built-in tool that automates build processes )
```

To test applications as a native app on mobile phones

ionic platform add platform name

ionic build platform name

ionic emulate platform name

To test these applications

To run on the actual platform

For android

ionic run android

Make sure to enable the Debugging mode on the device.

For iOS

We need to sign up for an Apple Developer account to test as a native app on an iPhone or iPad, this has a cost per year. To test for successful installation of NPM, we entered the command 'npm -v'. Ionic and Cordova were also installed from the terminal using these steps: 'sudo npm install -g ionic'. The '-g' indicates a global installation on the machine. Following the basic package installations, the versions were confirmed using 'ionic -v' and 'cordova -v'. To implement a new hybrid application, the 'ionic serve -l' (l for ionic lab) command was used in the terminal after appropriate prerequisite setup for the project. Once the pending bootstrapping process is carried out, we navigate into the new directory using the command 'ionic server'. This process will automatically serve the application on the browser. We started the app for this project using '\$ ionic start Ionic-map side menu'. The new project and necessary dependencies are pre-installed with side menu rather than a blank project. The entire bootstrapping processes were compiled and run on the local server as shown in Fig. 6.

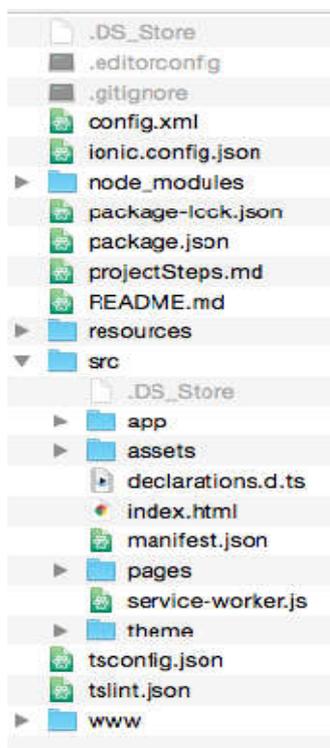


Fig. 6. Proposed system directory captured on macOS

The directory structure contains the following:

- Hooks: this directory contains customized Cordova actions that will be automatically executed at certain lifecycle events of the app.

- www: contains the current build when we run ionic preview inside the browser. The web pages index.html is contained here.
- Platforms: contain the Native project
- plugins: contains Cordova plugins installations, these are the wrapper around native functions
- Resource: contains icon and splash screens
- SCSS: contains SCSS codes that can output to the directory www/css/
- img: contains the images
- JS: contain customized JavaScript and Angular.JS files
- Lib: contain third party libraries
- Index.html: is the index default page
- Config.xml: is reserved for cordova configuration
- Ionic project: contains files for ionic configurations
- Package.json: has all the Node.JS dependencies
- Node_modules: Is automatically generated once you install the NPM dependencies with "npm install"
- tsconfig.json, tslint.json): Typescript or variant of JavaScript containing classes associated with the views. The build system will transpile this code into plain JavaScript which every browser can understand
- ionic.config.json contains some basic information about the project used in upload the app to the Ionic.io platform.

Software Specifications

The application was built and tested on both macOS and Windows platform using command line interface. NodeJS version 8.9.1 LTS was used. The Atom text editor was used to write the HTML, CSS and JavaScript enhancement codes on the macOS platforms. Google Chrome v. 49.0.2623.87 was used for testing and deploying the application.

Code design and Implementation

The code design involves understanding the basic concept related to Directives. Directives houses built-in-functions and Modules or containers for the application controllers. The concept of controllers JavaScript objects were used for controlling the data in the application. Scope basically binds the view and the controllers; It has different properties and methods which works together. Each controller is connected to the views. The frontend interface is made up of the inherent ionic components such as footer, status bar, header, content, lists and tabs etc. Ionic also permits referencing JavaScript or CSS source file for various functions. This has made life easy for the developers to invoke various logic based on the client's targeted needs or design goal. JSON data is mostly used to distribute real-time data across the design platform.

Result for Ionic interface

We followed the research design explained in our methodology in developing the hybrid mobile application. It should be noted that a humungous amount of time was saved compared to development using native apps for single platforms. The final result came to live by entering the command "ionic serve -l" on the local machine on <http://localhost:8100/ionic-lab>. The simulated result automatically shows up on the browser as in Fig. 7 and Fig. 8. The result in Fig.7 is a user interface for a mobile geographical map represented on android, iOS and Windows while Fig 8.

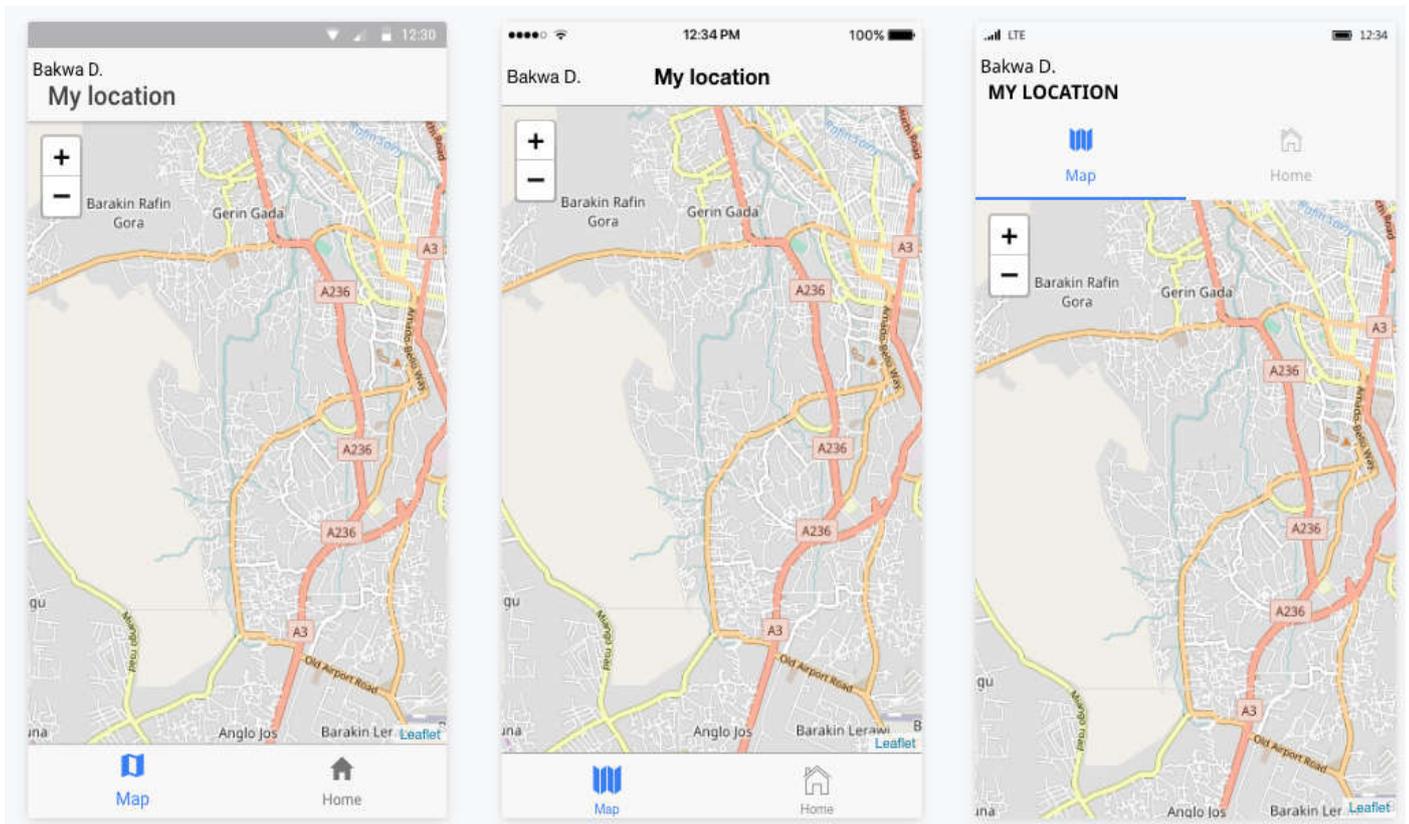


Fig. 7. Mobile version of hybrid mobile application displaying my location on iOS, Android and Windows phones

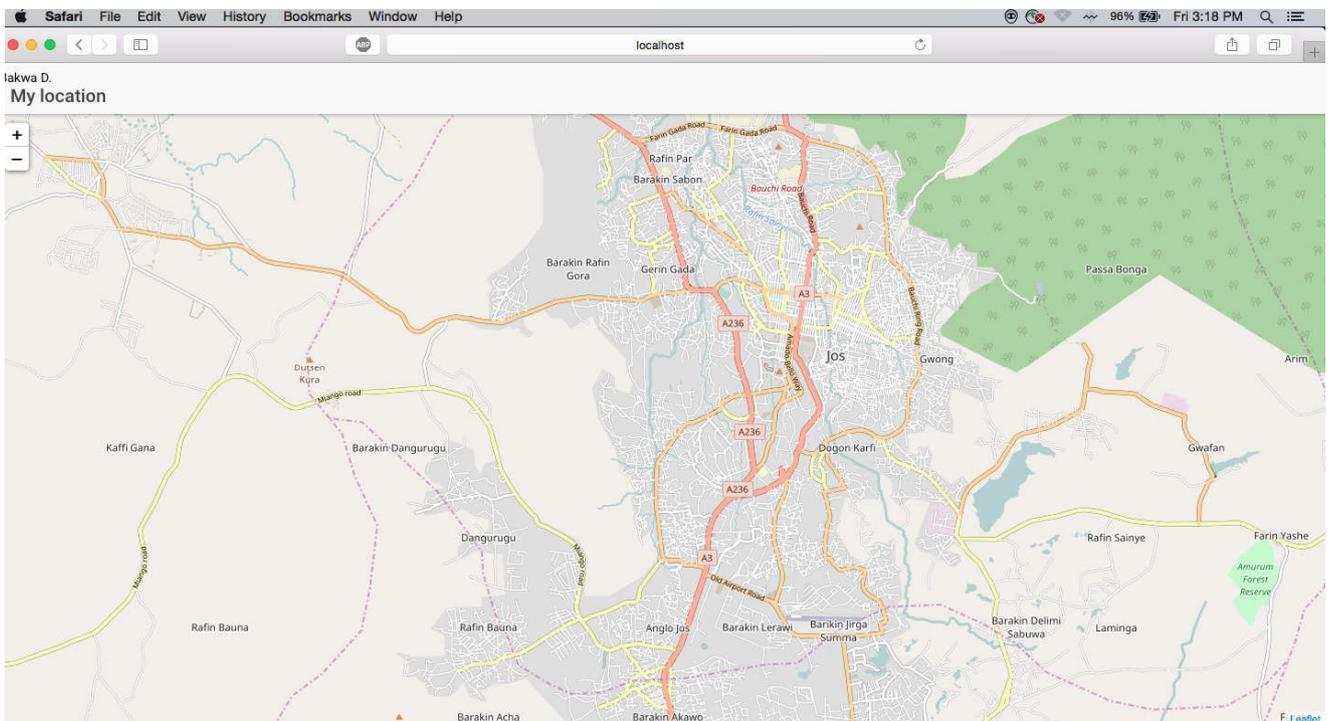


Fig. 8. Web version of hybrid mobile application displaying location around Jos

Represent the web version of the same application. The final output of the application can be adapted onto various platforms such as mobile and desktops without writing separate codes. The product of this study is a hybrid mobile application meant for identifying physical Geographic locations. The project reached its goal in building a functional cross-platform application with Ionic and leaflet. From the real-world perspective, this project did not focus on the features of the

application. It only emphasizes the benefits received from the various technologies.

Leaflet

Leaflet is a major component that was implemented in building the ionic application to create the interactive map. It is considered among the latest open source library for creating maps.

```

12  @Component({
13    selector: 'page-map',
14    templateUrl: 'map.html'
15  })
16  export class MapPage {
17    map: L.Map;
18    center: L.PointTuple;
19
20    constructor(public navCtrl: NavController, public navParams: NavParams) {}
21
22    ionViewDidLoad() {
23      console.log('ionViewDidLoad MapPage');
24
25      //set map center
26      this.center = [9.895653, 8.865071]; //University of Jos
27      //setup leaflet map
28      this.initMap();
29    }
30    initMap() {
31      this.map = L.map('map', {
32        center: this.center,
33        zoom: 13
34      });

```

Fig. 9. Code snippet for setting up geographical map in ionic framework

Its JavaScript logic is very simple and important in bringing several features into the proposed application. The features of leaflet include: 1) Interactive features such as dragging, scrolling, and zooming. 2) Visual features such as animations, fading, map controls, popups and resolution controls. Leaflet is designed with simplicity and performance in mind. It is designed to work across various mobile platforms and its API is easy to use. The process of coding was considerably easy. The map was initialized to view Jos geographical coordinates (long. 9.095653, lat. 8.0650). This is displayed in the code snippet in Fig.9. The zoom and attributes controls were set to its default.

DISCUSSION

The design of the hybrid application was less stressful with the ionic framework. The www directory contains the raw web application which includes HTML5, Angular and CSS. The entire design pattern of the application followed the MVVC pattern. MVVC architectural pattern is useful in separating codes into modules and also utilize JavaScript language across the entire application. This study targets developers, to make the right choice in order to build rich modern featured mobile application. This will indeed challenge the mobile native implementations and also impacts on operational versatility. Despite the advantages and disadvantages of native and web mobile app, hybrid mobile apps falls in the middle of the spectrum of both platforms to improve the problems of cost, timeliness, user experience, future maintainability, developer skills and mobile market competition. Hybrid apps have considerable benefits over native applications, specifically in terms of platform support, speed of development, lower learning curve for developers. Cordova serves as a bridge, which connects JavaScript with the native language. In short, if we need to connect with mobile hardware we use Cordova. Finally, Ionic framework has helped us tremendously in building a single cross-platform application which can run on Windows, iOS and android mobile app.

This study will be useful in order to build enterprise level applications, for code reusability i.e. limiting the effort of having to write code in Objective C or Swift for iOS platform, Java for android platform or .NET for windows. Ionic allows developers to code in JavaScript, HTML and CSS in order to build functional hybrid applications for users.

Conclusion

In this study, we explained the taxonomy behind developing modern apps. The native approach is credible in terms of performance and functionality; the web approach can be used across several mobile browsers. However, the hybrid approach allows a single code base work for different platforms (Windows, iOS and android). As a result, it saves cost and time for developers. Hybrid apps combine the best of both web and native app to deploy a native-like app to a wide range of platforms. The bottom line of this study is that the choice to go with hybrid approach is worthwhile considering the emerging framework and all it has to offer. A hybrid app will neutralize the competition in market competitions for the top three mobile platforms if widely accepted. We believe the design will support all platforms equally on phones and tablets.

Future Recommendation

A further improvement in performance of hybrid apps is needed. The one-size-fit-all strategy of hybrid mobile approach must live up to its promise by extending its functionalities to other platform aside major platform such as Windows, iOS, android and blackberry.

Acknowledgment

My appreciation goes to Prof. TankoIshaya, ICT Director, University of Jos, Prof. ChojiDavou, Head of Department, Computer Science, University of Jos, for their mentorship. I deeply acknowledge my supervisor Dr. Edim Emmanuel, Senior Lecturer, University of Calabar, for his guidance throughout my studies.

REFERENCES

- Adeleke, A., G. 2017. Influence of Time-on-phone on Undergraduates Academic Achievement in Nigerian Universities. *American Journal of Educational Research*. doi: 10.12691/education-5-5-14 5(5) pp. 564-567
- Anibal A., A (March 20, 2017). Hybrid Mobile Development. [Web log Post]. Work With The Best. Retrieved June 19, 2017 from <https://workwiththebest.intraway.com/blog-post/mobile-hybrid-development/>
- Buttigieg, S., & Jevdjenic, M. 2015. Learning Node.js for mobile application development: Make use of Node.js to develop of a simple yet scalable cross-platform mobile application. Birmingham: Packt Publishing.
- Chauvinc, M., & Phan, H. 2017. Learning path: Ionic: Application development with Ionic 2.
- Christ, M. A, 2011. Bridging the Mobile App Gap: *Sigma Journal: Inside the Digital Ecosystem*. 11(1), p. 29
- Dave Chaffey. Mobile marketing statistics compilation [online]. Smart insights official website; 21 January 2016 URL: <http://www.smartinsights.com/mobile-marketing/mobile-marketinganalytics/mobile-marketing-statistics>. Accessed: 20 February 2015
- Eschenbach, A. 2016. Plattformunabhängige Softwareentwicklung für mobile Endgeräte: Hybrid-Apps mit Cross-Platform Toolkits.
- Fern and ez, W., & Alber, S. (January 01, 2015). Beginning PhoneGap: Configuring your Development Environment.
- Gawley R., Barr J., Barr M. (2012) Native to HTML5: A Real-World Mobile Application Case Study. In: Zhang J.Y., Wilkiewicz J., Nahapetian A. (eds) Mobile Computing, Applications, and Services. MobiCASE 2011. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 95. Springer, Berlin, Heidelberg
- Griffith, C. 2017. Mobile App Development with Ionic 2: Cross-Platform Apps with Ionic, Angular, and Cordova. Sebastopol: O'Reilly Media.
- Griffith, and Chris. 2016. Mobile App Development With Ionic: Cross-platform Apps With Ionic 2, Angular 2, and Cordova. Oreilly and Associates Inc. Retrieved on feb 17th, 2017 from http://www.worldcat.org/title/getting-started-with-ionic/oclc/935639204&referer=brief_results Internet resource.
- Huynh, M., Ghimire, P., and Truong, D. 2017. Hybrid app approach: Could it mark the end of native app domination? Issues in Informing Science and Information Technology Education, 14, 49-65. Retrieved from <http://www.informingscience.org/Publications/3723>
- James. V. 2017. 99.6 percent of new smartphones run and roid or iOS. Retrieved on September 5 from <https://www.theverge.com/2017/2/16/14634656/> and roid-ios-market-share-blackberry-2016
- Lübke R., Schuster D., Schill A. 2012. A Framework for the Development of Mobile Social Software on and roid. In: Zhang J.Y., Wilkiewicz J., Nahapetian A. (eds) Mobile Computing, Applications, and Services.
- MobiCASE 2011. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 95. Springer, Berlin, Heidelberg
- Natili, G. 2013. PhoneGap 3 beginner's guide: A guide to building cross-platform apps using the W3C st and ards-based Cordova/PhoneGap framework. Birmingham, Engl and : Packt Publishing.
- NripinBabu&ArunBhat.Mindtect - Development of Hybrid Applications with HTML. www.mindteck.com 2013 pp. 4-12
- Oehlman, D., and Blanc, S. (January 01, 2011). Native Bridging with PhoneGap.
- Pamungkas, Z. S., TotalBoox, and TBX,. 2015. PhoneGap 4 Mobile Application Development Cookbook.Packt Publishing.
- Phan, H., and Hartington, M. 2015. Ionic cookbook: Over 20 exciting recipes to start up you application with Ionic.
- Phan, H., TotalBoox, and TBX. 2015. *Ionic Cookbook*.Packt Publishing.
- Ramanujam, P., Srinivasan, R., & Natili, G. 2015. PhoneGap: Create, develop, debug and deploy your very own mobile applications with PhoneGap.
- Rosca, S., & Patin, D. 2015. WebStorm essentials: Build efficient HTML, CSS, and JavaScript applications using the powerful WebStorm IDE. Pp. 199-240
- Saleh, H., Ethan, H., Tom, B., TotalBoox, and TBX. 2016. Mobile Application Development: JavaScript Frameworks. Packt Publishing.
- Shotts, K. 2016. Mastering PhoneGap Mobile Application Development. Birmingham: Packt Publishing, Limited.
- Spachos, D., Chifari, A., Chiazese, G., Merlo, G., Doherty, G., Bamidis, P., and 2014 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL). (November 01, 2014). WHAAM: A mobile application for ubiquitous monitoring of ADHD behaviors. 305-309.
- Tanque, M. (January 01, 2017). Security for Hybrid Mobile Development.
- Villalobos, R., and Lynda.com (Firm). 2015. *Building a Mobile App with AngularJS and Ionic*, eVideo: Clipart/images/graphics : English
- Weiß, B. 2016. AngularJS and Ionic Framework: Hybride App-Entwicklung mit JavaScript und HTML5.
- Wilken, J. 2016. *Ionic in action: Hybrid mobile apps with Ionic and AngularJS* ISBN 9781633430082 pp. 209-240 pages
- Wilken, J., and Bradley, A. 2016. Ionic in action: Hybrid mobile apps with Ionic and AngularJS.
- Yusuf, S. 2016. Ionic framework by example: Build amazing cross-platform mobile apps with Ionic, the HTML5 framework that makes modern mobile application development simple.
