

Xue Zhiming


Mobile App development

Bachelor's Thesis
Information Technology

Sept 2016



DESCRIPTION

		Date of the bachelor's thesis 2 December 2016
Author(s) Xue Zhiming	Degree programme and option Information Technology	
Name of the bachelor's thesis Mobile App development		
Abstract The aim of this study was to investigate how to use Ionic framework to build a Mobile App. Another purpose was to practice skills and learn new technologies. This study may helps IT students and developers to select the technology for Mobile App development. In the theoretical part, I researched the differences of various Mobile Apps, and stated their advantages and drawbacks. In the practical part, Node.js was used as a runtime environment and the App was developed by using the Ionic framework. The study was successfully completed. The App that was build in the study was designed as a social networks app. It contained three main pages, which were contact page, user page and the page of timeline.		
Subject headings, (keywords) Ionic, Nodejs, Mobile App, Hybrid App		
Pages 25	Language English	URN
Remarks, notes on appendices		
Tutor Timo Mynttinen		Employer of the bachelor's thesis Mikkeli University of Applied Sciences

CONTENTS

1 Introduction.....	1
2 Overview of Mobile Apps	2
2.1 Native App.....	2
2.2 Web App.....	3
2.3 Hybrid App	4
2.4 Future of Mobile Apps.....	5
3 Technical specification	7
3.1 HTML5	7
3.2 JavaScript.....	9
3.3 Node.js	9
3.4 Ionic	10
4 Preparation of Development	12
4.1 IDE.....	12
4.2 Node.js	12
4.3 Ionic and Cordova.....	13
4.4 Android Emulator	15
5 Implement	17
6 Result and Analyses.....	24
7 Conclusions.....	25
Bibliography	26

1 INTRODUCTION

Nowadays almost every one has smart devices with them. It could be a smart phone or a pad. The popularity of mobile devices is getting higher and higher. As the mobile devices are increasing, the number of mobile applications is also growing rapidly. IT companies put more effort on mobile applications, and people enjoy the convenience that mobile applications bring.

Mobile applications addict people so that they rely on them more than ever. With these applications in the mobiles, people can watch movies, listen to music, do shopping, get education, make appointments for various services without waiting in the line, money transfers or currency exchange. These conveniences greatly improve the quality of our life.

Therefore, the aim of my thesis is to develop a mobile app. The technologies I am using for this study include HTML5, CSS3 and JavaScript. After this thesis, I expect to have more knowledge of responsive web development and mobile application development.

In this thesis, I'll first introduce the types of mobile applications and make comparisons with them. Then there is an introduction for the technologies that I am using or that relate to the topic. In the practical part, I'll set up my development environment first, and explain the software and environment that are installed. Then I'll start to build the application. I'll try to show and explain every step I did in this work.

2 OVERVIEW OF MOBILE APPS

Mobile apps are applications which were designed to run on mobile devices, such as smartphones and pads. As mobile devices became more and more powerful and intelligent, the use of mobile applications becomes a lot popular than before.

The number of mobile app has grown rapidly in the last few years. According to the report of Mobile Marketing Statistics 2015 from Smart Insights Ltd (2016). In US, the media time on mobile devices has exceeded the time on the desktop and other media. The use of mobile devices is 9% higher than that of desktop devices. Meanwhile the number of mobile devices is still increasing, and the market of desktop devices is shrinking.

Because of the mobile devices occupy the most of the market share of Internet access, there are some services that do not even provide the full functionality service on desktop. For example, Instagram, the most popular photo sharing society online works like this. The web version of Instagram is not able to publish and edit photos.

As same as desktop applications, there are various kinds of mobile apps. Mobile apps are classified by the programming language they used. There are three types of mobile applications, native app, web app and hybrid app. These three types of mobile apps have their own advantages and targets, and these will be discussed in this chapter.

2.1 Native app

A native app is an application developed by using a native programming language.(Raluca 2013) For example, developing a native app for IOS requires the developer to use the native programming language, Swift. For Android is Java and for Windows Phone is C++. Therefore, if an application wants to be in the app store of Android and IOS, it has to be developed twice by using two different languages.

Native apps were designed to run on the system level, and therefore native apps take more advantages of the device's features, such as camera, GPS, fingerprint and

heartbeat detectors. Also, it can access the device's notification system. Most of the apps that can be downloaded from App Store or Google Play are native apps.

The cost of development and maintenance for native apps is higher than for the others because of the fragmentation of devices and the systems. The diversity of hardware causes the difficulty of development and maintenance.

The pros of native apps include that they provide the best user experience and interaction. Native apps offer different experiences for different platforms. And, native apps can be run offline for saving the bandwidth. Also, native apps are able to access local resources. And, for business, native apps have a clear revenue model.

The cons of native apps include that they are complicated to transplant to other platforms. Native apps have higher costs for development and maintenance. Also, native apps need to be submitted to App Store. And, native apps need to share profits with App Store.

Native apps are the best choice for applications requiring high performance, such as games and photo editing applications. And because of native apps have ability to invoke notification system, apps which require push notifications, for example chat apps, should be developed as native apps.

2.2 Web app

Web app commonly refers to an application that can run on web browsers on various smart devices, such as laptops, tablets or smart phones.(Wikipedia 2016c) Web app is friendly to almost every platform. Thus, it provides similar experience for users on different platforms.

Most Google applications are web apps. The advantage is saving time and resources to develop applications for all the platforms. Gmail is a nice example for a web app. Users can visit gmail.com in the browser by using whatever system, IOS or Android, Windows or Linux. And, it provides the same experience in all modern devices.

The pros of web apps include that they do not need installation. They are cross platforms. They have lower cost of development and maintenance. And, they can be updated rapidly. And compare with native apps, web apps have lower requirement for specifications.

The cons of web apps include that they need the Internet. They have slower response and depend on the speed of bandwidth. Also web apps have compatibility issues with different browsers. Web apps are not available on App Store. And, web apps have security risks.

The biggest advantages of web apps are that they are cross platform and do not need installation. Once the project is deployed, it can be visited by any devices with a Internet browser. Compared to native apps, web apps are easier and take less time to develop. It requires less technologies than developing a native app. Therefore, web apps are usually lightweight. Web apps are the best choice for products that do not require devices' features and do not require access to local resource from mobile devices.

2.3 Hybrid app

Hybrid app is the application mixed from a native app and web app. Hybrid apps are like web apps embedded in a shell of native apps. They were designed to combine the advantages between native apps and web apps. By combining these two technologies, hybrid apps can take advantage of the features of native apps, such as running offline, accessing local resources, and that they can be download from App store.(Raluca 2013)

Hybrid apps can be easily transplant to other platforms. Because some contents of app are web-based, only the native app frame needs to be build. This reduces the workload of development. And, there are some frameworks that could help developers to build the the native app frame quickly. The only thing to develop and maintain is the web-view content. The topic of the hybrid app development frameworks will be discussed in the next chapter.

The pros of hybrid apps include that they have similar development process with Web Apps. They take advantage of Native Apps' features. Same as web apps, they can update the content without installing a new version. And, they can use a framework to help the development.

The cons of hybrid apps include that they use Web-based content which has slower response, which mostly depends on the bandwidth. They may not provide very high performance. They do not support all system features. And, they will be difficult to debug, if knowledge of native framework is lacking.

Hybrid Apps have better performance than Web Apps, and look more native, thanks to hybrid app frameworks, like Phonegap and Ionic. It only requires web front end technologies for building a hybrid app. This is a big convenience for front end developers.

2.4 Future of mobile app

For the last few years, developers have discussed whether to choose to build a native mobile application or an HTML5-based Web App. As mentioned before, each kind of mobile app has its own pros and cons. But choosing the proper one to develop is very important. However, the final decision depends on the needs of the company.

Table 1 is the comparison of each kind of Mobile App. It compares the performance, costs, and other things that should considered before starting development.

Table 1. Comparison of web app, hybrid app, native app

	native app	web app	hybrid app
Distribution	App Store	Web	App Store
Development Speed	Slow	Fast	Moderate
Development Costs	High	Low	Moderate
Maintenance	Hard	Easy	Moderate
Performance	Fast	Slow	Moderate
Offline Mode	Yes	No	Yes
Installation	Needed	No Need	Needed
Cross Platforms	No	Yes	Yes

The development of mobile apps involves many processes. First, there should be a beginning idea, then developers start to plan, and design the prototype. And, then they turn to development, testing and deployment. Afterwards maintenance and debugging are needed.

After having the very beginning idea, the next thing to consider is which kind of an app is more suitable for the use. Native Apps have the best performance and the device access. Therefore, it is more suitable for developing games or mobile applications requiring high performance. Web Apps are cross-platforms and the development is easier and quicker, but the performance is poor and functionalities does not completely work. On the other hand, hybrid apps take advantage of native app and web app. Hybrid apps offer native performance and web-based content, and they are cross-platform, the same HTML code could be reused on different platforms. Usually, companies that want to get a presence in the App store and do not want to spend a lot of money. They will build a hybrid app as a wrappers for their website.(Raluca 2013)

However, the future is bright for mobile apps. As the number of people who has smart phones continues to grow, so does the number of mobile applications. According to the report of Smart Insights(2016), 90% of time on smart phones is spend on apps. And as it says in the report “It is clear that mobile is the future, and within 3 years it will come to dominate digital ad spending”.

3 TECHNICAL SPECIFICATION

3.1 HTML5

HTML refers to Hyper Text Markup Language. It is a primary markup language used for structuring the web pages. HTML5 is the fifth version of HTML which was released as a new standard markup language on 28 October 2014 by the World Wide Web Consortium (W3C) (Wikipedia 2016a). The goal of HTML5 is to improve the native web and to optimize the performance of web apps. HTML5 can do more with less code. It simplifies the development by creating new elements and APIs to accomplish them in only HTML.

HTML5 introduces new elements such as `<nav>`, `<header>` and `<footer>`, which are more semantic and easier for search engines to understand. (Before HTML5 released, the `<div>` assigned with specific ID were used for the page structure such as `<div id="header">`, `<div id="content">`. This kind of use of `<div>` is more complicated and not semantic. Whereas the new elements solved this problem, they were designed to simplify the structure and make it more semantic. Figure 1 shows the differences of page structure by using HTML4 elements and new HTML5 elements.

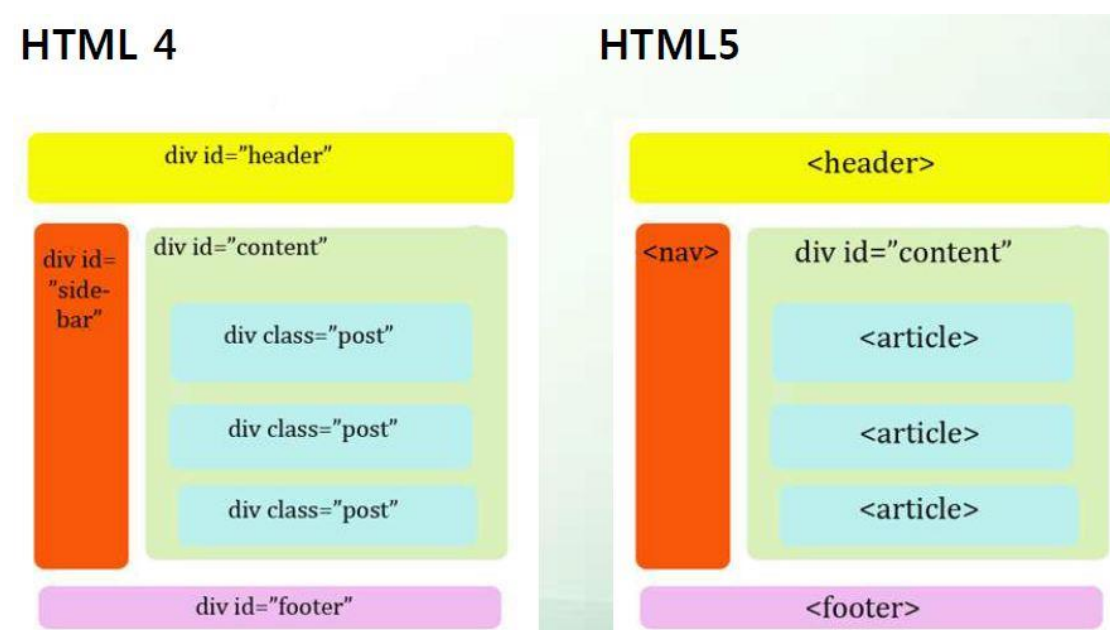


Figure 1. Structure difference between HTML4 and HTML5 (WordPress 2011)

Although HTML5 was released for a long time, but old tags of previous version of HTML are still around. Because they are still essential elements of the HTML structure. Using the old tags doesn't make a difference on the appearance. But, using HTML5 semantic tags will help for the SEO (search engine optimization).

In order to provide better performance for web apps, HTML5 added many useful features such as local storage, new types for inputs, and browser-supported validation for forms. These new features make development goes simpler for developers and make the function faster. These function used to be done with JavaScript, but now they are natively supported by HTML5.(Pieters 2014)

Also new APIs give Web Apps more ability to access and interact with the device. With the getUserMedia API web apps are allowed to access device camera and able to pass rich data back. Figure 2 shows more HTML5 related new APIs.

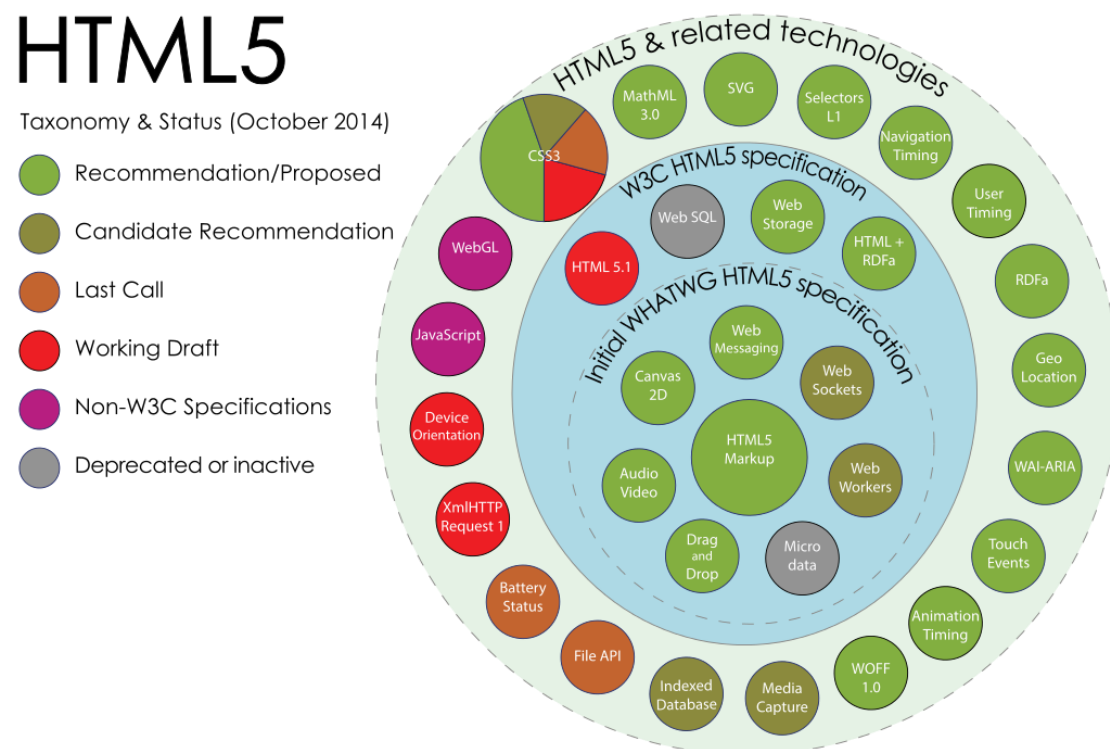


Figure 2. HTML5 related APIs (Mavrody 2012)

3.2 JavaScript

JavaScript is one of three main technologies of web page development. JavaScript is a programming language that makes web pages able to interact with visitors. HTML creates the basic structure of the page. CSS is responsible for the layout and decorations. JavaScript program behaves on to the page so that the web page is able to response to user's actions.

JavaScript is the only client-side programming language to handle all DOM manipulation. It runs on a user's browser and only needs to be downloaded once. Without JavaScript, interactive interface on web page would not response so rapidly. That's why JavaScript plays a very important role in front-end developing.

Traditionally, JavaScript only runs on the web browser. But, due to the development of JavaScript, the usage of JavaScript has changed. At first, as a client-side programming language, JavaScript was used to build a highly responsive web page which can greatly improve user experience and shorten the waiting time from the server. But, now JavaScript has been brought to the server-side as well.

Actually server-side JavaScript was introduced in December 1995 by Netscape Communications Corporation(1998). But, it wasn't popular at that time. Since Node.js was released, server-side JavaScript implementations became popular than ever. JavaScript is now one of the most popular programming languages in the world and the most popular language on Github. (Github 2016.)

3.3 Node.js

As Wikipedia(2016b) states "Node.js is an open-source, cross-platform JavaScript runtime environment for developing a diverse variety of tools and applications. In simple words, Node.js is server-side JavaScript. Node.js is made for creating high performance, lightweight real-time web apps quickly and easily. It allows JavaScript to be used on the server-side and client-side simultaneously.

Node.js was released in 2009. In the first two years Node.js could only run on Linux. Then developers started to combine V8 JavaScript Engine, a low-level I/O API and event loop into Node.js. After that Node.js became popular and got many contributions from other developers.

NPM is one of the most important features of Node.js. It is the package manager for JavaScript. NPM is short for Node.js Package Manager. Users can download JavaScript modules by using the command “npm install package-name”. Modules are really helpful for building apps. Because they are reusable, developers do not need to rewrite the utility method or copy paste the code to new projects. It saves a lot of time in repeated works.

3.4 Ionic

Ionic is a popular mobile app framework that helps users to develop Hybrid Mobile Apps quickly. Ionic is chiefly focused on the native-look UI and interaction of Hybrid Apps. Ionic comes with very native mobile application layout. It provides developers with the ability to build Hybrid Apps by using web technologies such as HTML, CSS, and Javascript. Currently, Ionic requires the technical support of AngularJS in order to work on its full potential. Otherwise, the capable UI interactions, gestures, transition animations, and other features will be missing.

Ionic is more like a stack that has bundled several technologies together. It combines the technology of Cordova, Angular with itself. With these three technologies, it provides a extremely easy way to build a Hybrid Apps.(Wilken 2016)

As the foundation, Ionic uses the Cordova hybrid app framework. Basically, Cordova allows a web app to be wrapped by a native app. As a result, these web apps can be installed on devices. Cordova provides an isolated browser instance on hybrid apps as the web view, which is the place for demonstrating contents.

Cordova also provides APIs that can be use to interact with devices. For example, access to the camera or read/write files to device's storage. In the Ionic

documents(2016), there are more than 70 APIs. These APIs make it easy to build and deploy a Hybrid App.

Secondly, it could use AngularJS in page contents. AngularJS is a front-end framework now maintained by Google. It lets users to use HTML as a template language and contains more methods to build a rich and robust Mobile App.

4 PREPARATION OF DEVELOPMENT

This chapter will introduce how to install the necessary applications, and how make sure applications are installed properly. Also, it introduces configurations and basic commands for these applications. The laptop system that used for this study is Windows 10. Therefore, most of the installation packages could be downloaded directly from the software's official website. There are also documents for developers on its website.

4.1 IDE

IDE is short for integrated development environment. An IDE is software that developers can write code and test on it. Typically, an IDE includes a code editor, a compiler and a debugger. It could be standalone or getting more utility by install extensions.

There are many choices for web programming IDE. Sublime Text and Atom are highly recommend for front end development. Sublime Text and Atom are both cross platforms. They both have installation packages for Windows, Mac and Linux. For sure there are more IDEs for developers, users can choose the one they are more familiar with.

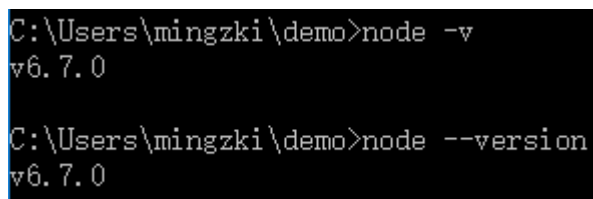
Sublime Text is a very light-weight code editor with a lot of useful features. But a license is required for commercial uses. Atom is an open source code editor that developed by Github. Atom is as hackable as Sublime Text. There are thousands of packages in the packages store that can add new features and functionality to Atom. But the important thing is that Atom is open source application, it's free to use in any case. Therefore, Atom is installed for this study. Atom can be downloaded from their official website, <https://atom.io/>.

4.2 Node.js

Node.js has two versions for users to download. One is v4.6.0, which is the LTS version. LTS refers to long term support. Another version is the current version which

is v6.7.0. Current version has the latest features, but it is not as stable as the LTS version. Therefore, the LTS version of Node.js is recommended for most users. But the beta version of Ionic requires Node.js v6 or a greater version to be installed. As a result, Node.js v6.7.0 will be installed.

First, download the installation package from the Node.js official website and install it. After installation, the version number of Node.js could be checked by the command line. Open CMD, and type the command “node --version” or “node -v”. Then it shows the version number in CMD. As we can see in figure 3.

A screenshot of a Windows Command Prompt window. The prompt shows the current directory as C:\Users\mingzki\demo. Two commands are entered: 'node -v' and 'node --version', both of which return the output 'v6.7.0'.

```
C:\Users\mingzki\demo>node -v
v6.7.0

C:\Users\mingzki\demo>node --version
v6.7.0
```

Figure 3. Show version command

If it reports an error in CMD, the solution is add the folder path of Node.js into the environment variables. In this case, Node.js is installed in the default folder which is C:\Program Files\nodejs. The environment variable can be found through the path: System info → Advanced system settings → Environment Variables.

4.3 Ionic and Cordova

To create an Ionic project, the Ionic CLI and Cordova have to be installed first. NPM is required for installing the Ionic CLI and Cordova, and because NPM comes with Node.js, which was installed earlier. Therefore, it takes only one command to install them.

First, we open CMD and type the command “npm install -g ionic” and “npm install -g cordova”. It takes few minutes to install. Then we take a look at the versions to make sure they are installed properly. Figure 4 shows the version of Ionic and Cordova.


```
C:\Users\mingzki>ionic -v
2.1.0

C:\Users\mingzki>cordova -v
6.3.1
```

Figure 4. Version of Ionic and Cordova

Once the installations are done, it is always good to create a demo project first. We use the command “ionic start project-name --v2” to create the first Ionic project. However, Ionic has two version that are available for use. By Omitting “--v2” in command, it will create an Ionic 1 project.

After the first project is created, we enter the project folder in CMD. First, we type the command “ionic serve” to test the project right in the browser. Figure 5 shows that Ionic server is running, and the project address, and some basic commands. Figure 6 shows how the app looks on the browser.

```
Running live reload server: http://localhost:35729
Watching: www/**/*, !www/lib/**/*
✓ Running dev server: http://localhost:8100
Ionic server commands, enter:
  restart or r to restart the client app from the root
  goto or g and a url to have the app navigate to the given url
  consolelogs or c to enable/disable console log output
  serverlogs or s to enable/disable server log output
  quit or q to shutdown the server and exit
```

Figure 5. Ionic serve

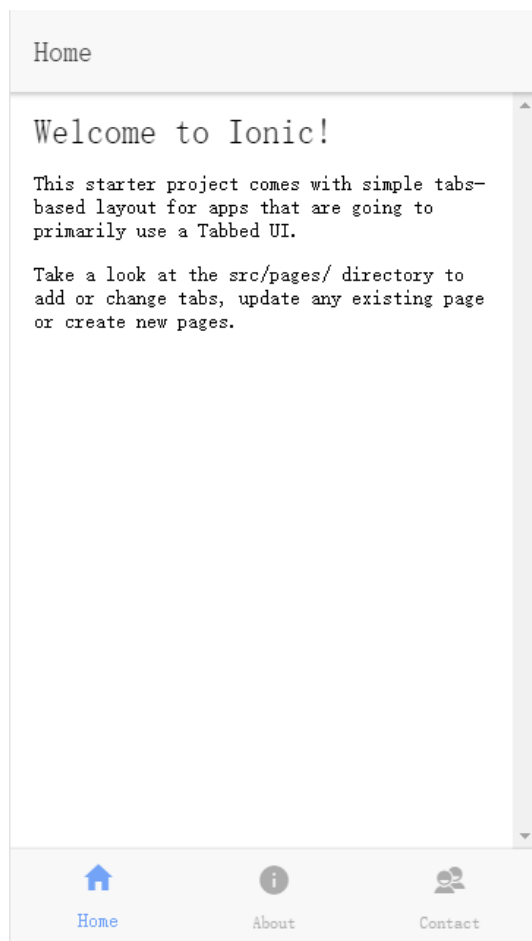


Figure 6. Ionic default UI

4.4 Android Emulator

For the reason of testing and debugging, it is necessary to install an Android emulator. The same as in other applications, there are many alternative Android emulators for users to choose from.

AndY is one of the best Android emulators. Unlike other Android emulators focused on gaming experience, AndY is mainly focused on overall user experience and offers a lot features. It has Full Android UI, camera integration, sensor integration, and it's free for personal uses.

The official website of AndY is <http://www.andyroid.net/>. It doesn't offer standalone installation package. The installer will automatically download the package and install it to C drive.

5 IMPLEMENTATION

In this chapter, the idea is to build a social networks app by using the Ionic framework. It's basically an app focused on media sharing. There will be 3 main pages in this App. One is the contact page, which is the page to find friends. Another one is a user page to change users' profiles. And the third one is the moment, the page where the user can see friend's sharing.

First, We start a project by Ionic and open it with Atom. Here is the structure of the Ionic project. Figure 7 shows the structure of the Ionic project. The introduction of the full project structure can be found in the official website of Ionic.

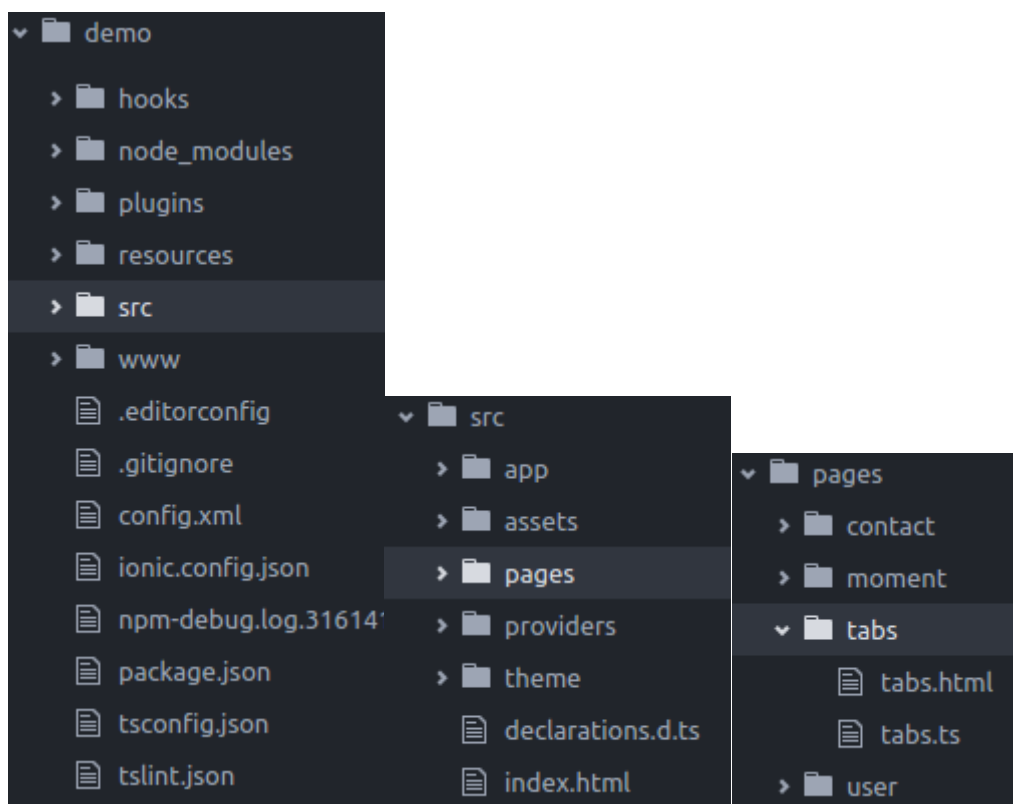


Figure 7: Project structure

As mentioned before, three pages are needed in the App. Therefore, navigation for these three pages should be created first. In this case, tabs are used for switching pages, as shown in Figure 8.

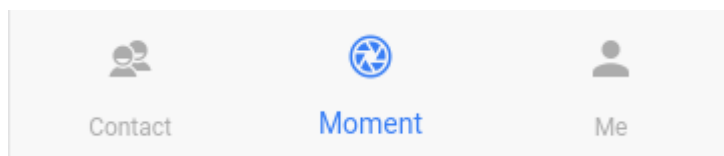


Figure 8. Tabs

The user page should include three pages. The first one is the login page, the second is the signup page, and the last one is the user profile page. Figure 9 shows the login page.

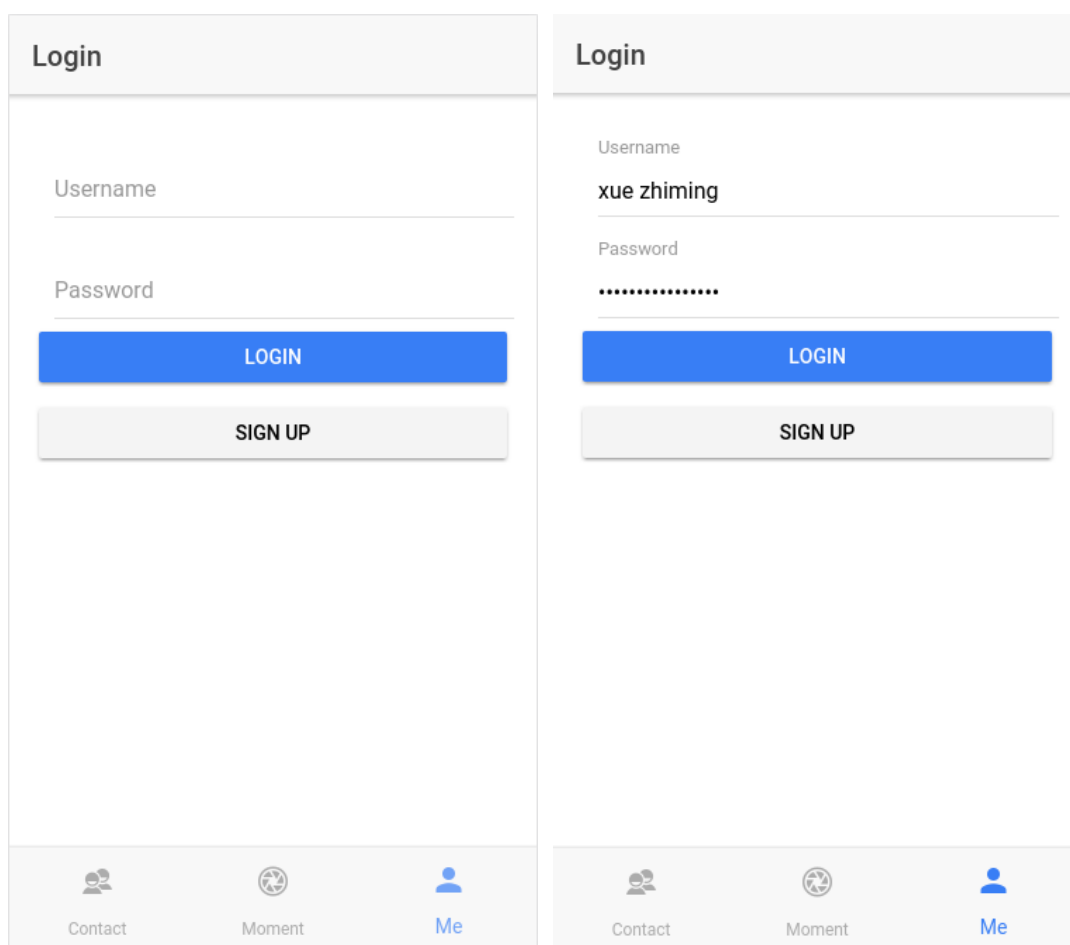


Figure 9. Login page

The signup page looks almost the same with the login page. When clicking on the signup button, the signup page should pop out. There are two main method to pop out page in Ionic. One is `nav.push()`, another one is `modal`. In this app, we use both

methods. We use modal to present the signup page. Figure 10 shows the signup page. And we use the `nav.push()` method for the user page. Figure 11 shows the user page.

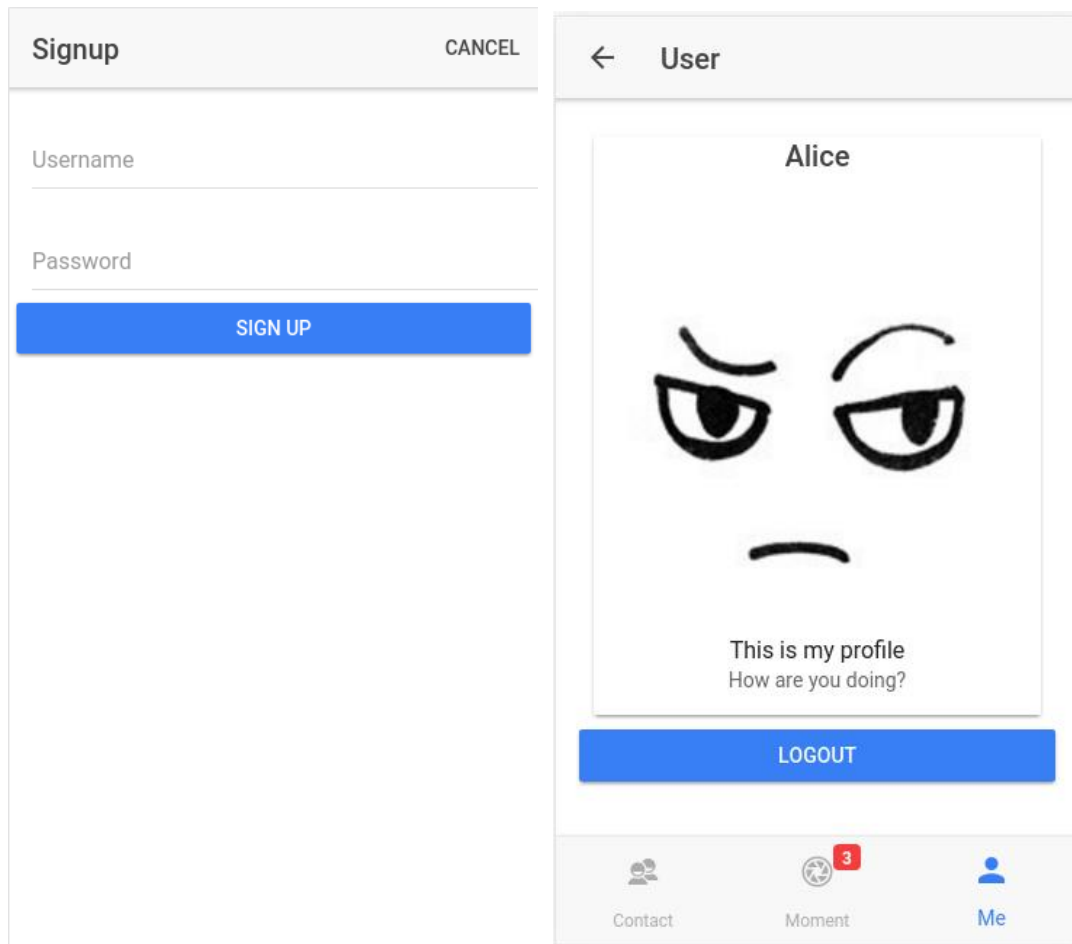


Figure 10. Signup page

Figure 11. User page

The basic usage of modal is to import a modal controller and the page that is going to be presented. Then a function must be created for displaying the modal view. Code 1 shows the basic usage of modal.

```

import { ModalController } from 'ionic-angular';
import { ModalPage } from './modal-page';

export class MyPage {
  constructor(public modalCtrl: ModalController) {
  }

  presentModal() {
    let modal = this.modalCtrl.create(ModalPage);
    modal.present();
  }
}

```

Code 1. Basic usage of Modal

In the contact page, contacts are slidable from both sides. Figure 12 shows the contact page. It can easily be set up by using <ion-item-sliding> component. This component needs to be a child of the <ion-list> component. Next we add options by adding <ion-item-options> component after <ion-item-sliding>. Code 2 shows the basic usage of sliding list.

```

<ion-list>
  <ion-item-sliding>
    <ion-item> </ion-item>
    <ion-item-options>
      <button>
        <ion-icon></ion-icon>
      </button>
    </ion-item-options>
  </ion-item-sliding>
</ion-list>

```

Code 2. Basic usage of Ionic sliding list

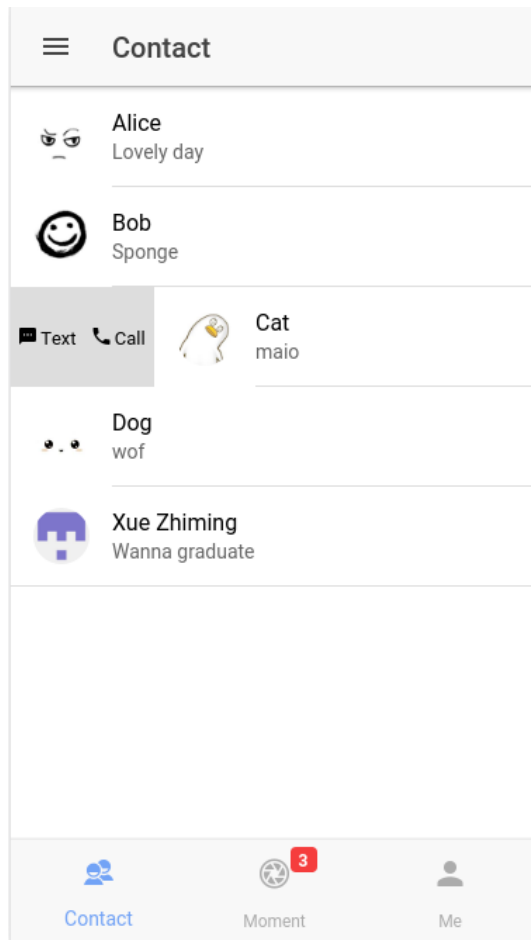


Figure 12. Contact page

The content of the moment page is basically based on the `<ion-card>` component. The code for avatar and user name in this page are almost the same with the contact page. And the picture is embedded in `<ion-card-content>` component. Then another `<ion-item>` is added in the bottom.

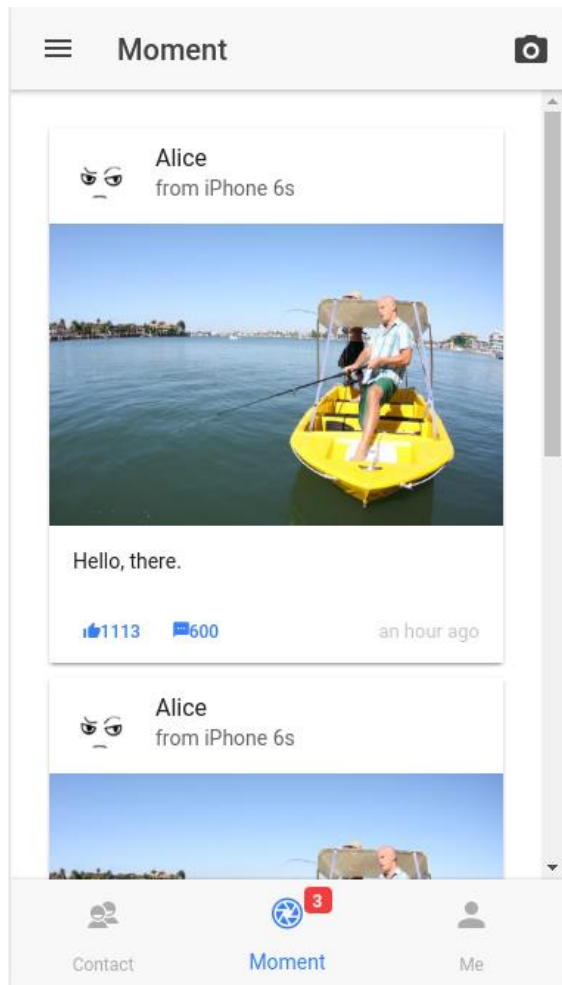


Figure 13. Moment Page

For users to post their pictures, there is a action sheet button on the right top corner of the moment page. It slides up from the bottom of the screen and shows a set of options.

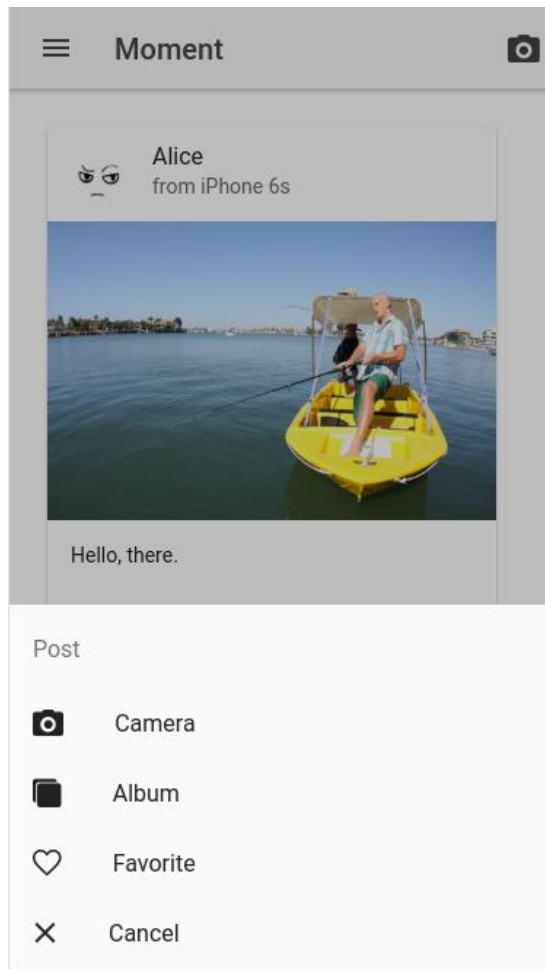


Figure 14. Action sheets

The implementation is done until this point. That's almost everything in this App. Next step is to bring it to a real device.

6 RESULT AND ANALYSES

In this chapter, the main point is to test the app and see the result. There is no IOS device that could be tested. Therefore, it is only test on the browser and an Android phone.

In order to install it on real devices, the application needs to be built as an installation package for Android. Android studio and JDK(Java SE Development Kit) are required for building packages. Then we use the command “ionic build android” to build the package. The process is really quick, it takes only a few seconds in this case.

Firstly, the app was tested and debugged by the browser, therefore this App works totally fine on the browser. It responses quickly, and doesn't delay actions. But when it switches to Android emulator, it seems quite laggy for some reason. There is a delay in touching and also in scrolling.

On the real device, which is an Android phone, it doesn't show delay anymore. It's smooth on scrolling, touching and responses quickly. Therefore, the reason causing the delay on the Android emulator may be the laptop.

However, there are some functionalities that still need to be implemented. Like side menu and native-API related function. That is why, the app looks really simple at the moment.

7 CONCLUSIONS

The main purpose of this study was to practice my skills and learn some new technologies. And I think it was really helpful. Although the app still has so many points that need to be improved, I am quite satisfied with it so far.

At first in the theoretical part, I thought it would be easy to build a hybrid app by using Web front end technologies. But the fact was not like that. The technologies used in the practical part were completely new to me. And, the information that can be found online was kind of out of date, due to too fast Ionic updates. And, the official documents were not detailed enough for me. I met a lot of problems and was unable to find solutions in the implement part. Therefore, it will be good to learn some JavaScript framework before starting a project.

However, the study is successfully completed, and the result is not bad. I've finished the project and achieved the goal, although it was hard for me to build this App and make it run on the real device. But also, I've learned a lot from this study. I have become familiar with hybrid app development and responsive web design. And, In the end, I hope this study could be useful for all readers.

BIBLIOGRAPHY

Github 2016. Web page. <http://github.info/>

Referred 23.09.2016

Ionic Team 2016. Ionic Docs. Web page. <http://ionicframework.com/docs/v2/native/>

Referred 26.09.2016

Wilken, Jeremy 2016. Web page. Getting Started With Ionic: Introduction.

<https://code.tutsplus.com/tutorials/getting-started-with-ionic-introduction--cms-25919>

Referred 26.09.2016

Netscape Communications Corporation. 11 December 1998. Retrieved 16 July 2016.

Server-Side JavaScript Guide. [http://docs.oracle.com/cd/E19957-01/816-6411-](http://docs.oracle.com/cd/E19957-01/816-6411-10/contents.htm)

[10/contents.htm](http://docs.oracle.com/cd/E19957-01/816-6411-10/contents.htm)

Referred 24.09.2016

Mavrody, Sergey 2012. HTML5 & CSS3 Quick Reference.

Referred 22.09.2016

Pieters, Simon 2014. HTML5 Differences from HTML4. Web page.

<https://www.w3.org/TR/html5-diff/>

Referred 22.09.2016

Raluca, Budiu 2013. Web page. Mobile: Native Apps, Web Apps, and Hybrid Apps.

<https://www.nngroup.com/articles/mobile-native-apps/>

Referred 18.09.2016

Smart Insights (Marketing Intelligence) Ltd 2016. Mobile Marketing Statistics compilation.

<http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>

Referred 24.09.2016

Wikipedia 2016a. Web page.

<https://en.wikipedia.org/wiki/HTML>

Referred 18.09.2016

Wikipedia 2016b. Web page.

<https://en.wikipedia.org/wiki/Node.js>

Referred 26.09.2016

Wikipedia 2016c. Web page. https://en.wikipedia.org/wiki/Web_application

Referred 30.09.2016

Wordpress, 2011. Web page. <https://html5studynote2.wordpress.com/>

Referred 22.09.2016