

**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**

---

**SINGAPORE**

Wee Kim Wee School of Communication and Information

CI6222 - MOBILE & UBIQUITOUS APPLICATIONS

# *Fotosquare*

A location based photo sharing app

Submitted By

NAH ZHENG XIANG, PHILSON (G1701513D)

# FOTOSQUARE

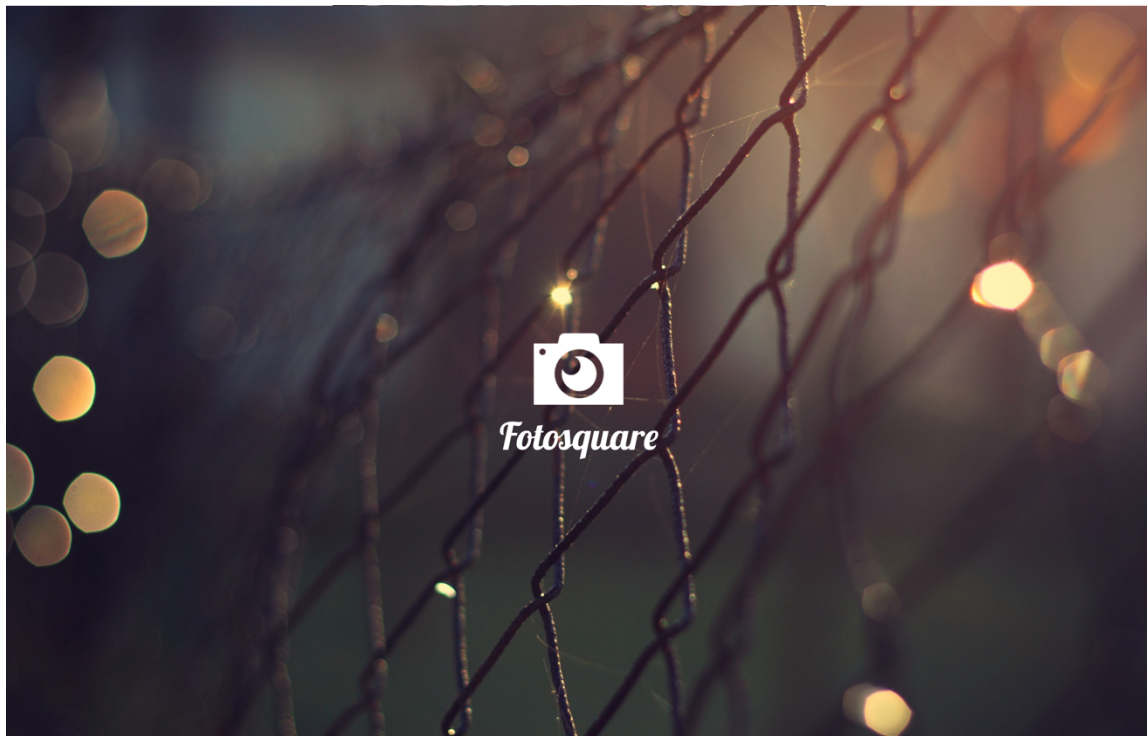
*location based photo sharing*

18 Nov 2017

## INTRODUCTION

---

Fotosquare is a location based social photo sharing app. The aim of this app is to let users discover the places around them, through pictures. Leveraging on social media technologies, we also make taking pictures and sharing with friends a more fun and enjoyable experience.



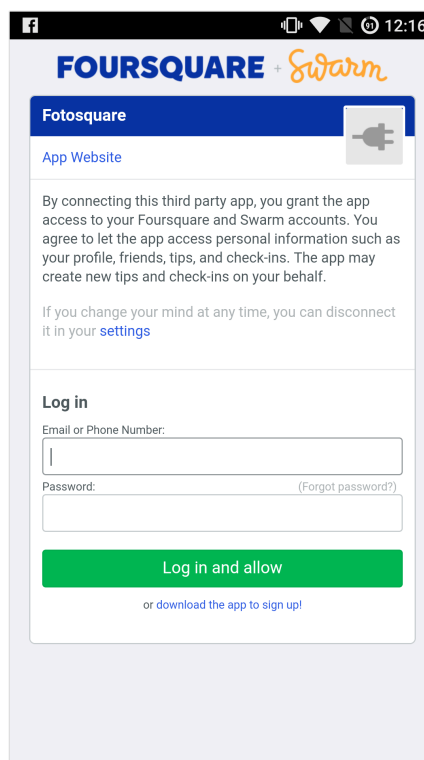
### API Usage

Fotosquare is tightly integrated with the Foursquare API. Meaning, the photos you post will actually also appear on your Foursquare account. Adding to the social element of the App, you can also see the friends you have followed on Foursquare, and click to view the photos they have taken, making for a more social experience. Documentation and information on the API is available on: <https://developer.foursquare.com/>.

# FEATURES

## FOURSQUARE LOGIN

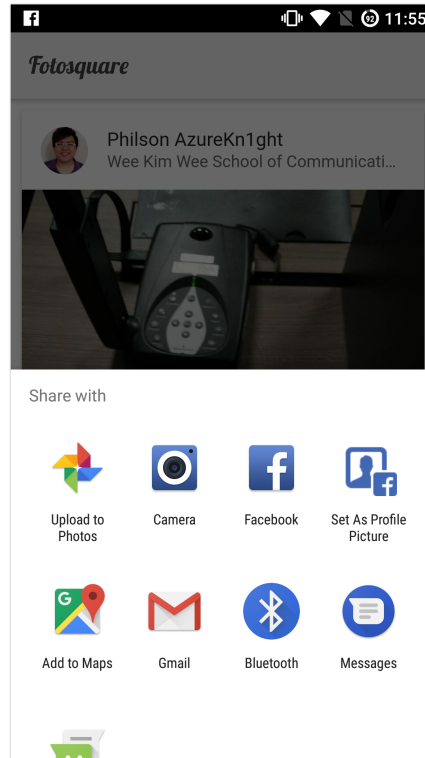
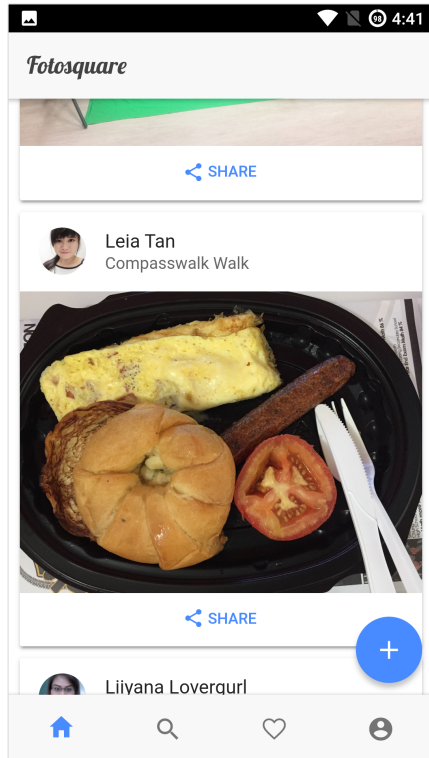
---



Fotosquare leverages on the Foursquare API to provide users with a social platform for location based photo sharing. As such, they are required to login with their Foursquare account to access the many features provided by the app. By clicking on the “Connect with Foursquare” button, an integrated in-app browser opens to allow users to login with Foursquare’s website. The user is kept within the app rather than launching a separate browser window. As an added security measure, the app will never handle any user password data directly, everything is mediated by Foursquare. Login is done using OAuth 2.0 to provide an access token for the API calls.

# HOME FEED

---

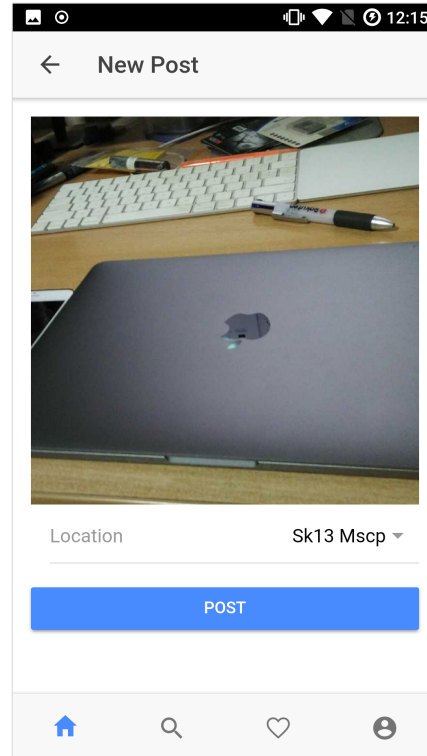


The Home Feed provides a list of pictures taken nearby the user's location. Based on the GPS coordinates of the phone, a new and different list of photos will be shown wherever you go. This adds heavily to the discovery element, and is one of the key selling points of Fotosquare. Go to new places, and discover new things!

Additionally, photos from the Home Feed can be re-shared, and passed on to other apps and shared on other social media platforms. This adds to the viral capabilities of the app. The use of the Android Intents approach of sharing, means that the utility of the sharing will only get better as the user installs more apps on their phone over time.

# POST A PHOTO

---

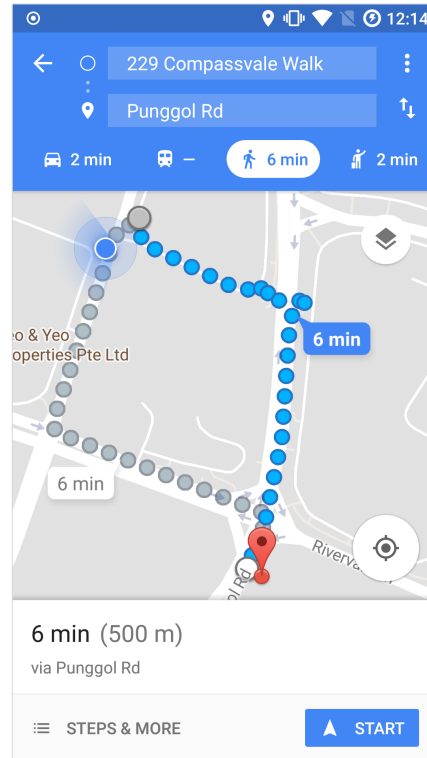
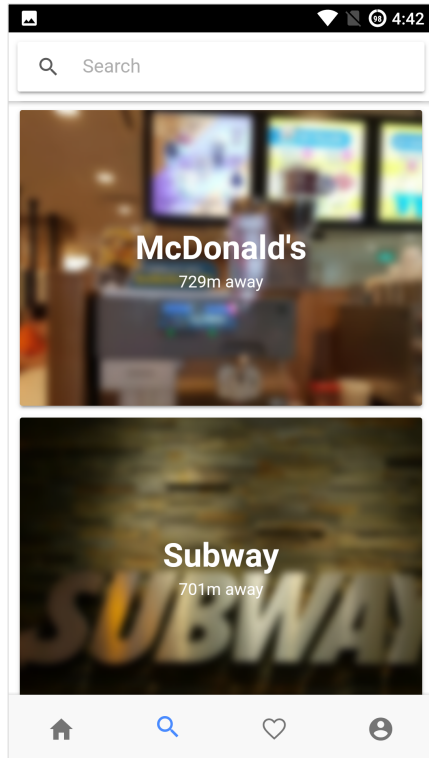


Other than viewing pictures taken nearby, users can also contribute by posting their own photos taken at the location. By clicking on the floating action button (FAB) on the Home Feed, the app will launch the phone camera, which the user can then use to take a picture. After taking the picture, the user can make adjustments by cropping or rotating the picture.

Once the picture is confirmed, users are brought to the New Post page, where they will specify the location that the picture was taken at. Every photo posted must be tagged to a particular location. This is in line with the app's theme of location based photo sharing, and encourages people to share photos based on what is happening around them.

# EXPLORE LOCATIONS

---

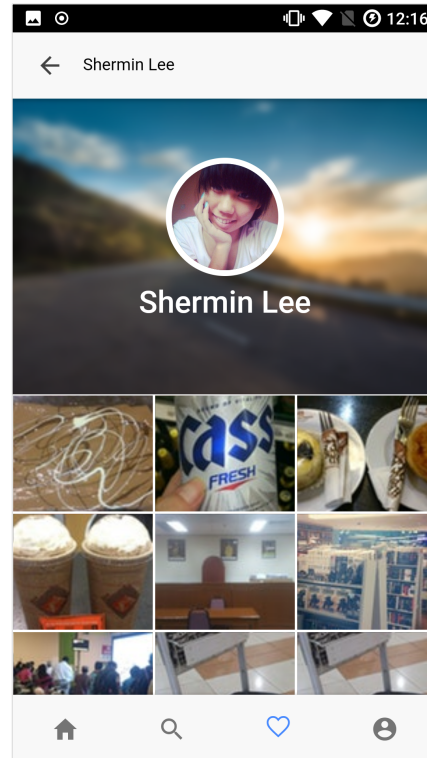
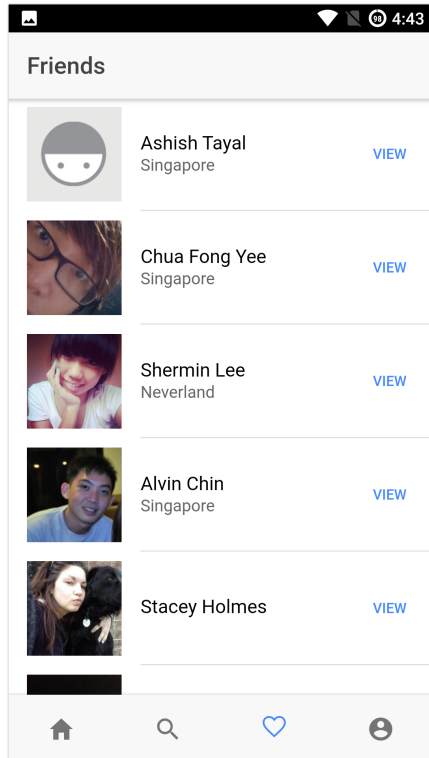


Another fundamental aspect of Fotosquare, is the sense of exploration. Exploring new locations is highly encouraged, to discover new places, see new things, and share new pictures. In the search tab of Fotosquare, a list of recommended nearby locations is provided to the user. As with the Home Feed, this list is customized and will change based on your GPS location data. The distance from your current point is also listed.

Tapping on any of the locations will launch the Google Maps app, and provide navigation directions on how to get to the place. This is handy for when users are exploring new places that they have never been to before. Alternatively, there is a search bar at the top of the page, to allow users to find a location, assuming they have a place in mind already.

# SOCIAL CONNECTIONS

---

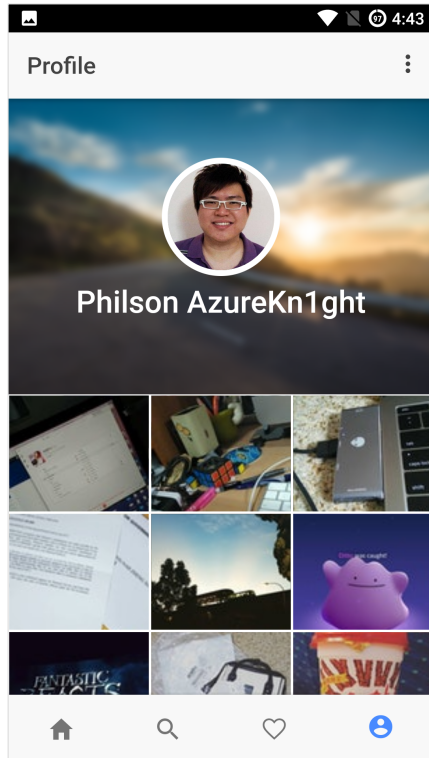


To provide an element of social connections, the Fotosquare app also allows you to view a list of all your Foursquare friends. From there, you can tap on “View” to see a friend’s profile page. Their profile page will list all of the pictures which they have taken to date.

Clicking on any of the pictures will launch the integrated photo viewer feature, that shall be explained in the next section. The social features of this app help users keep in touch with their friends to see what they are up to. It is a key focus of Fotosquare, and shall be continually improved and added upon in future updates.

# PERSONAL PROFILE

---

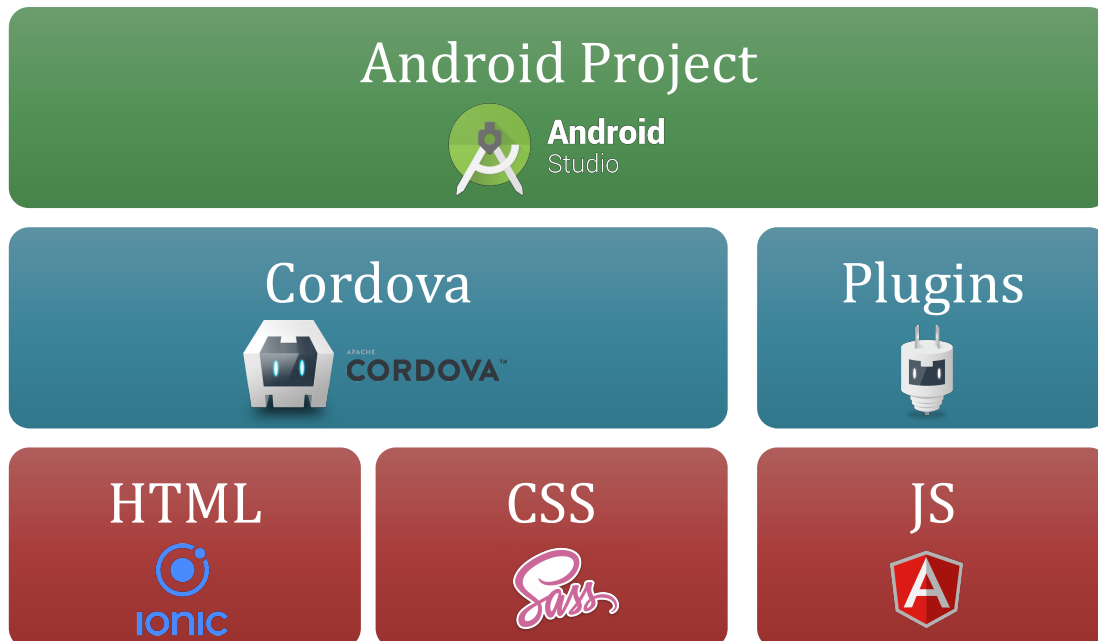


The personal profile page allows users to view all the pictures they have taken in 1 page. The photos are ordered reverse chronologically, with the most recently taken photo listed at the top. Tapping on any of the photos will open a fullscreen image viewer. The photo viewer supports pinch to zoom gestures, and also lists the location where the photo was taken at.

Similar to the Home Feed, users can tap on the share button to open up the Android share sheet to share the photo with other installed Android apps or post to other social media platforms. This function makes use of the Share Intent, and works with any other Android app that supports it.



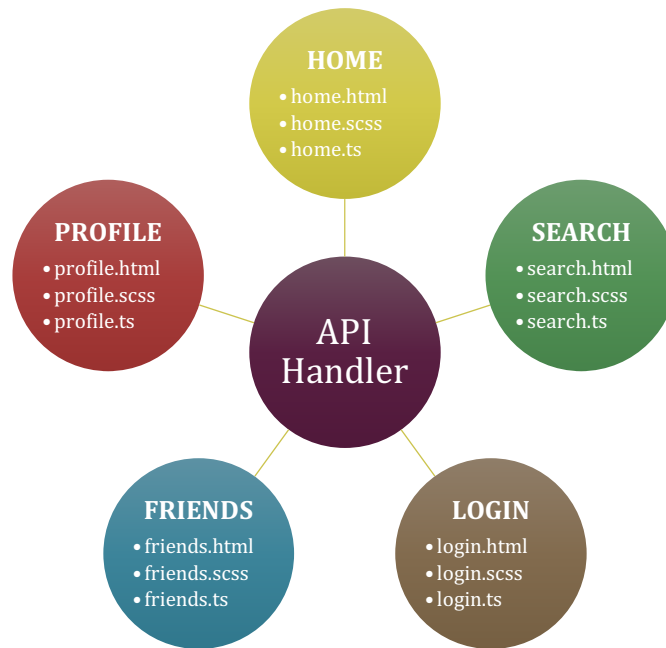
# ARCHITECTURE



Fotosquare was developed using the hybrid app approach. Which means, most of the application is coded using web based languages like HTML, CSS and JavaScript. Specifically, the **Ionic 3.0 Framework** was used in the development of this app. Which means, the entire coding process was done using HTML, Sass (which compiles to plain CSS), and **Angular 4.0** which runs TypeScript (which compiles to plain JavaScript).

The Ionic files are then compiled using **Apache Cordova** into the native Android and iOS projects. Additionally, Cordova provides some 3<sup>rd</sup> party plugins that allows the app to programmatically access some native hardware features, such as cameras, just by using JavaScript. All of these plugin files and source code files are then compiled into an Android project.

The Android project files can then be built either through **Android Studio** or via command line into the actual *apk* files. The *apk* file can then be distributed via the Google Play Store and ran on the Android devices. The project uses web standards compliant code which allows greater cross platform reach, to target not only iOS and Android, but potentially other OS platforms as well. <https://cordova.apache.org/docs/en/latest/guide/support/>



There are 5 main pages in the app – Home, Search, Profile, Friends, and Login. There are also 2 supporting pages – Post and Friend-Profile. The interactions of the main pages are as illustrated in the diagram above. Each page comprises of 3 files – *html*, *scss*, and *ts* files. The *html* and *scss* files will dictate the layout structure and styling of the page respectively. Then the *ts* file will dictate the application logic related to that page. Finally, all the pages will pass data to and fro through the API Handler which is coded in the file: *foursquare-api.ts*

The fotosquare app does not store any database locally or on a first party server. All the data is routed to and from the Foursquare API by means of asynchronous API calls. For example, the Login and posting of new photo is handled at the backend by Foursquare servers. In order to optimize performance, and prevent the app from hanging, these API calls are made asynchronously using JavaScript *Promise* Objects. This ensures that the app is not stalled and waiting constantly for the JSON return data.

In a way, the method in which this application is coded and structured follows similarly to the 3-tier architecture approach. The *html* and *scss* provide the Presentation Layer display, the *ts* files handle the Application Layer logic, and all the data passes through the API Handler which acts like the Data Layer. This sort of separation helps make the application more flexible and maintainable.

# PLUGINS

A full list of Ionic dependencies is listed in the file: *package.json*

```
82     "cordova-plugin-inappbrowser": {},
83     "cordova-plugin-x-socialsharing": {},
84     "cordova-plugin-camera": {},
85     "cordova-plugin-file": {},
86     "cordova-plugin-file-transfer": {},
87     "com-sarriaroman-photoviewer": {},
88     "uk.co.workingedge.phonegap.plugin.launchnavigator"
```

This project mainly leverages on the above stated Cordova plugins.

The *inappbrowser* plugin allows the app to display an integrated browser window to facilitate the Foursquare OAuth 2.0 login.

The *socialsharing* plugin allows the app to pass data natively to Android Intents, thus enabling the use of cross app sharing of media.

The *camera* plugin allows the app to launch the native Android camera, capture an image, and receive the taken image to process.

The *file* and *file-transfer* plugins enable the app to make http file transfers, thereby allowing the app to upload images to the Foursquare API.

The *photoviewer* plugin provides the app with a native photo viewer functionality that supports the pinch-to-zoom feature.

The *launchnavigator* plugin allows the app to launch the Google Maps app installed on the phone to provide directions navigation.

# RELEASE PLAN

As of 17<sup>th</sup> Nov 2017, Fotosquare has been released onto the Google Play Store. The link to the Play Store listing can be found at:

<https://play.google.com/store/apps/details?id=com.fotosquare>

Given the ability to cross compile the code for hybrid apps, an iOS version is already in the works, and slated for release by the end of the year.

# FUTURE WORK

Due to the time constraints of this project, I was unable to fully complete all the functionalities that I had intended to include in the release. Some of the additional features that will be worked on in future include:

- Photo editing and custom filters
- Ability to add and find friends
- Video sharing capabilities
- Built in messenger functions
- Social feed and user tagging

These additional features will be added to the list of possible functionalities to be developed. They will then be ranked and prioritized for iterative development using the Agile Model SDLC approach, for future releases.

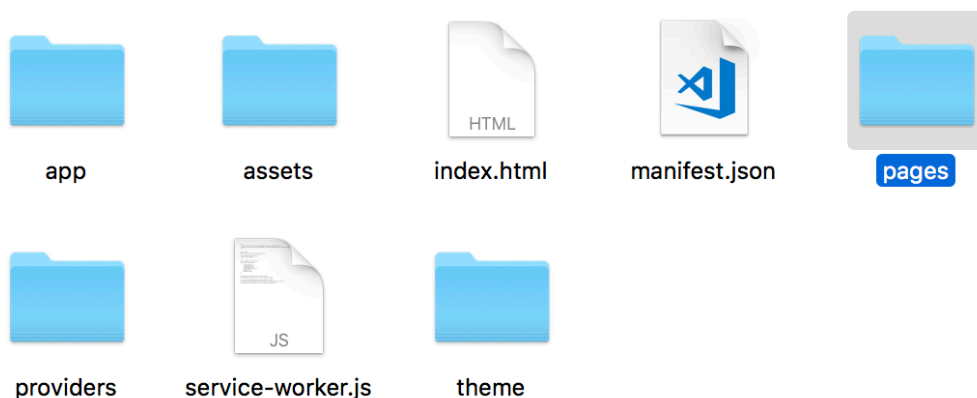
# APPENDIX

## APPENDIX A

---

This section details the folder structure of the source codes of this project. The pre-compiled Ionic source codes can be found in the following folder:

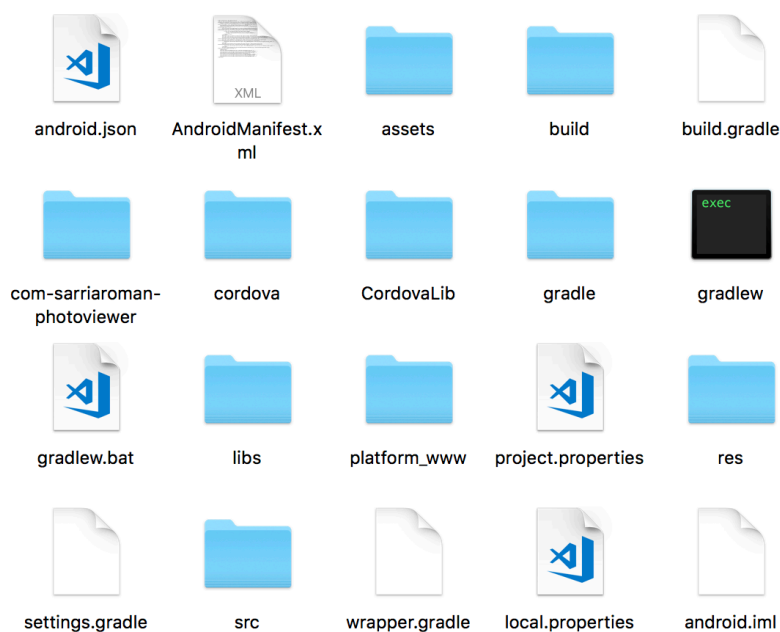
**fotosquare > src**



The compiled Android project can be found in the folder:

**fotosquare > platforms > android**

This project folder can be used to build the *apk* file with Android Studio.



Similarly, the compiled iOS project can be found in the folder:

**fotosquare > platforms > ios**

This project folder can be used to build and run the iOS app with Xcode.

