Analysis Me

# Table of Contents

1. **Introduction**
2. **Motivation**
3. **Related work**
4. **System design**
5. **System implementation**
6. **Experiments and analysis**
7. **Conclusions and future work**
8. **Appendix**

**Introduction**

With the development of IoT(Internet of Things), cloud and data science, mobile ads has becoming a more and more powerful industry. Different from traditional ads we have seen on TV or other medias, mobile ads display on phones, tablets and other mobiles devices in various kinds, such as pictures, texts, video, html5 and links. Combined with cloud and data, mobile ads can target ads customer more concisely and instantly at anywhere and anytime.

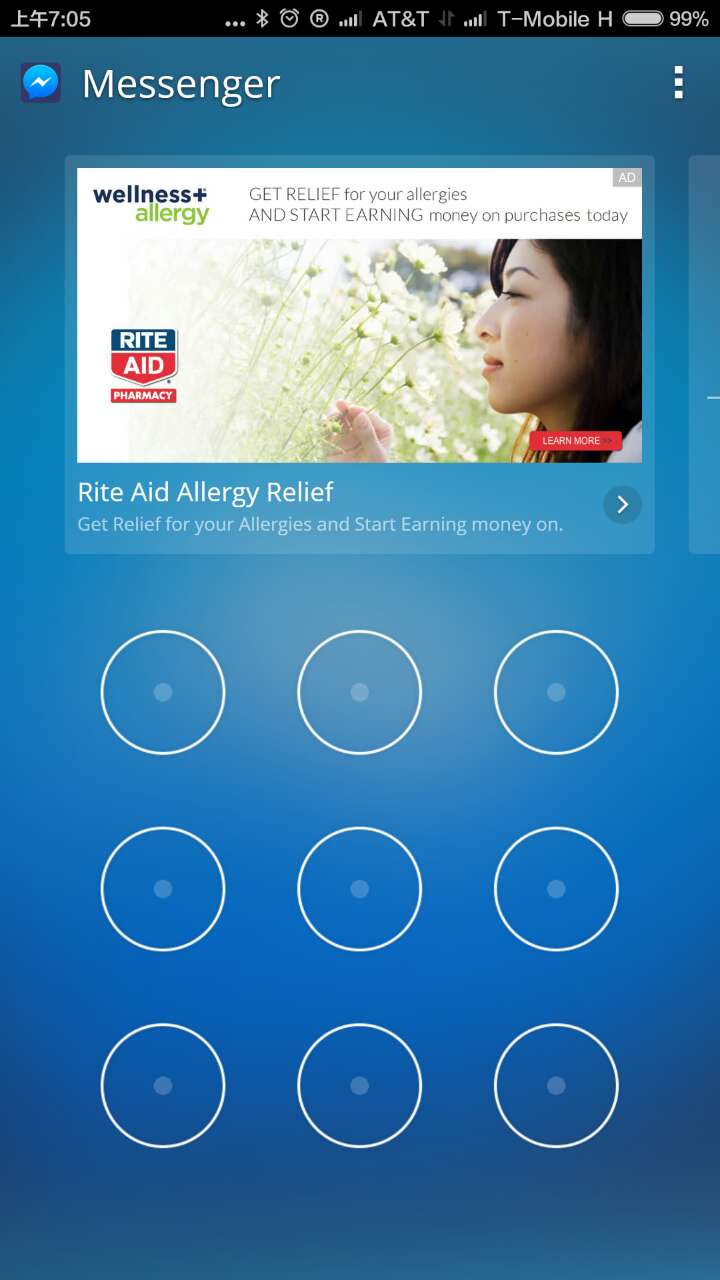
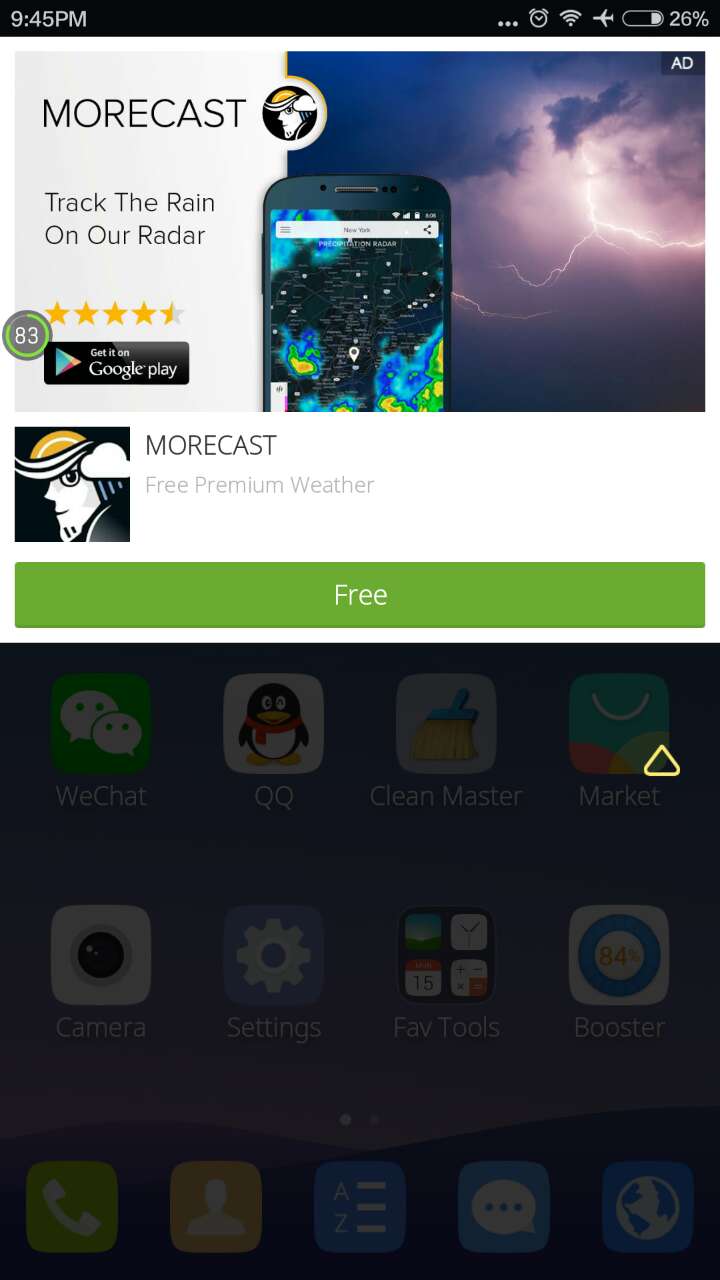
  

Figure 1 Mobile Ads Samples

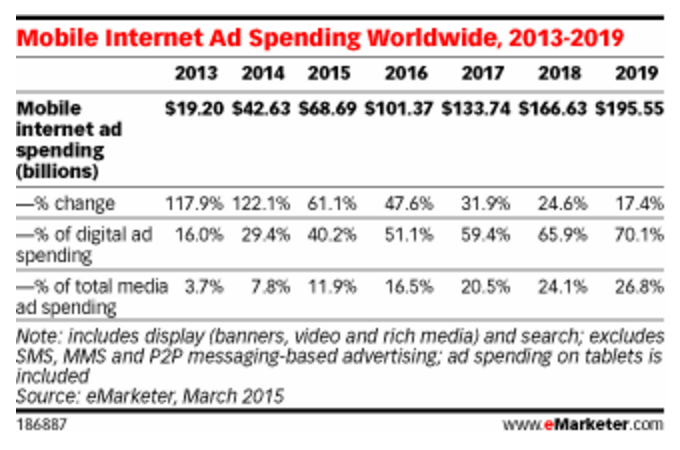
As reported, mobile ads industry is a hundred billion level business. In 2016, the mobile and internet spending will be over $100 billion and more than 50% will be digital ad spending. Especially in China and other developing countries,, accordingly the leading advertisers are also changing to devote to mobile ads market. Even in developed countries, there are still growing customers in mobile devices[1].

Figure 2 Mobile Ads Market Statistics

**Motivation**

Traditional advertising formats on mobile devices suffer from poor click through rates and are considered to be invasive and non-intuitive for the end user. At present, industry experts are advocating a new advertising format called Native Ads. However, there isn’t sufficient metric to validate the effectiveness of this new advertising format. Since native ads require significant effort by the developer to incorporate the ads in their apps, a quick comparison of the efficacy of this format will assuage the fear of the smaller app studios.

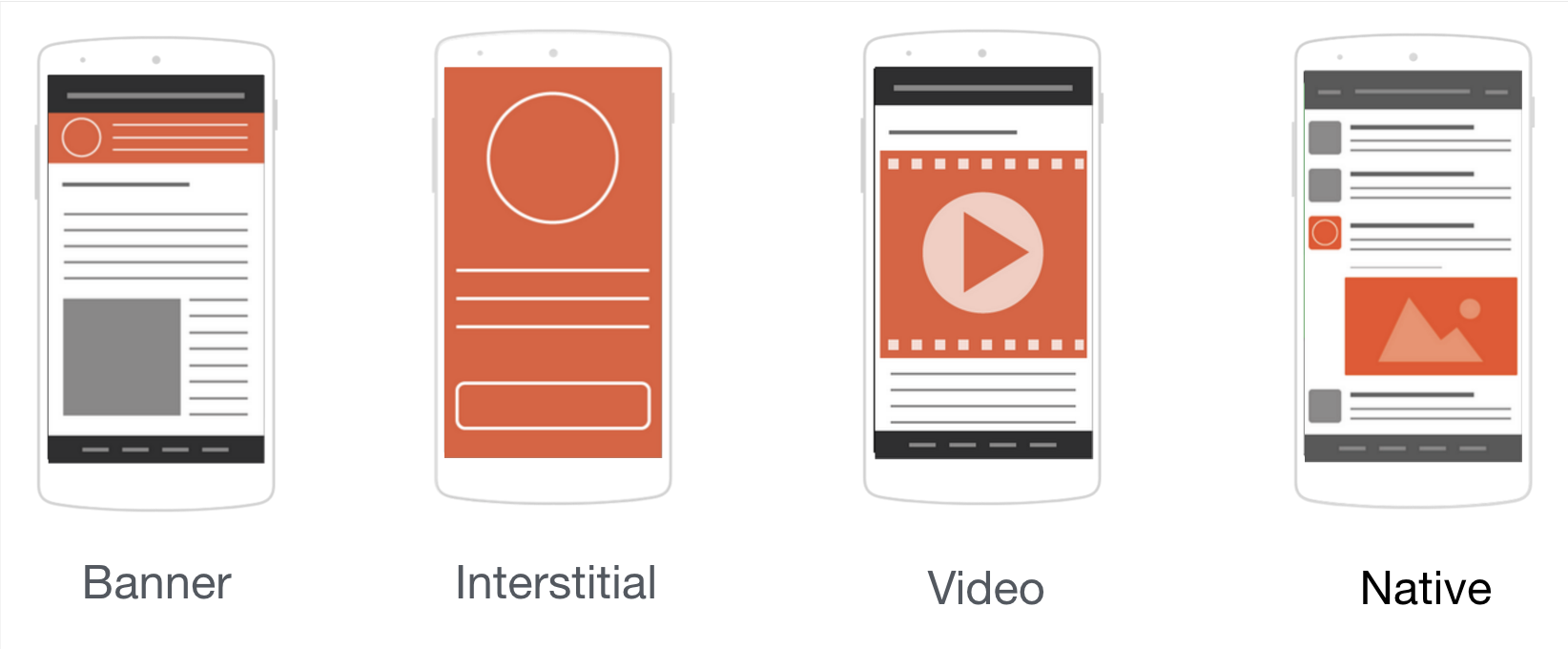


Figure 3 Ad Formats

The above figure shows all the available ad formats for mobile devices. Although banner ad formats have been extremely popular among mobile app developers (publishers), it is not able to get enough users to click on them. So, banner ads tend to have lower performance. In order to boost the monetization efforts of mobile app developers, the mobile ad industry has been experimenting with various other ad formats like ‘Interstitial’ format, ‘Video’ format and the most new entrant is the ‘Native’ ad format. Although Interstitial ads have high performance and garner more impressions, it comes at the cost of being extremely intrusive and poor UX design. On the other hand, Video ads have shown great promise with the ability to gain high engagement while delivering an immersive full screen experience. However, video ads have delivered this mostly for game apps. Finally, Native ads has emerged as a new trend pioneered by Facebook and has delivered on its promise of high performance while not disrupting the user experience.

Below, you will find the anatomy of a Native ad. The primary elements are the small icon, title, description, big picture (or video), call to action button. By using either one or more or all these five elements, the app developer can configure the layout of an ad to fit perfectly with their existing app user interface. This way, the app can deliver ads without disrupting the user experience.

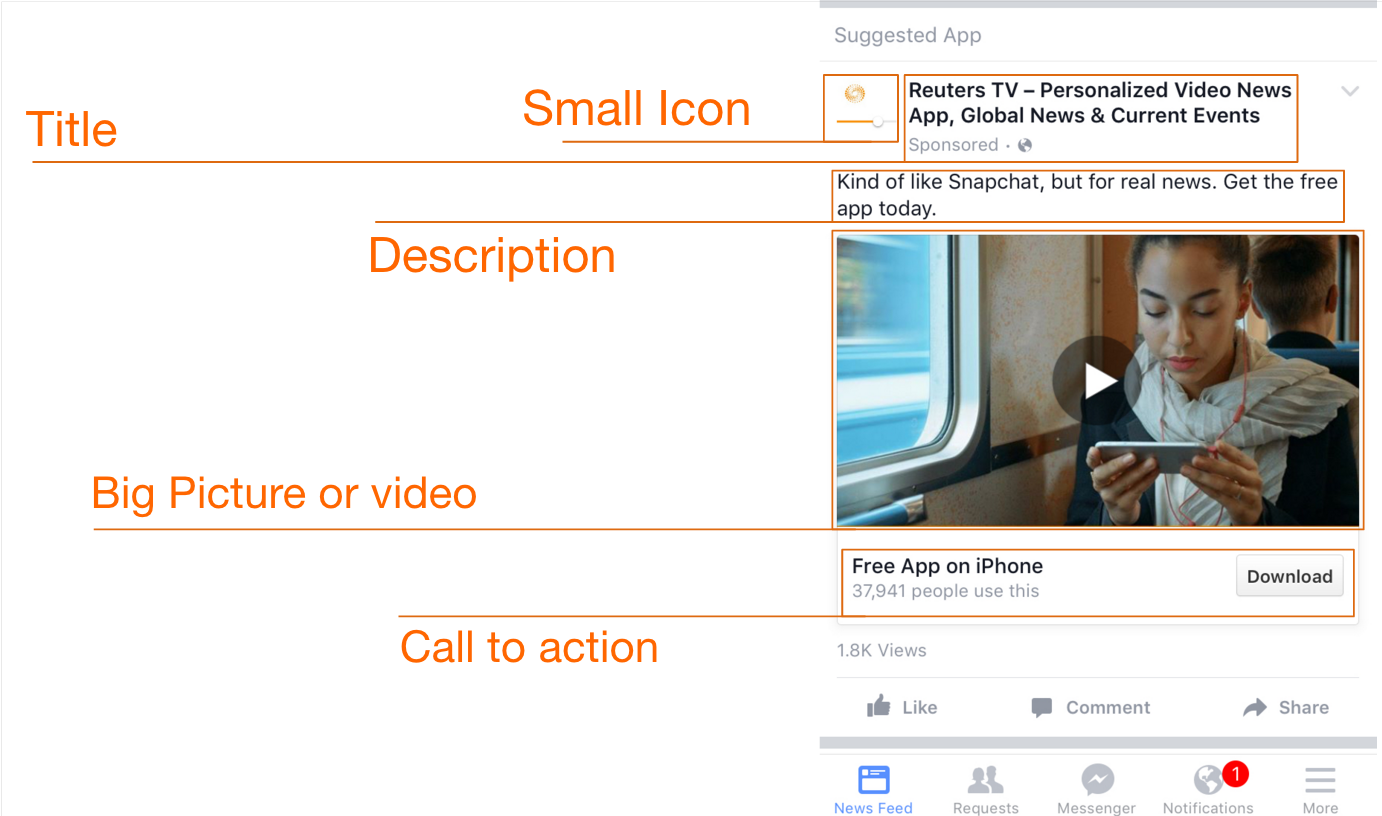


Figure 4 Anatomy of Native Ad

**Related Work**

Since the advent of Native ads, many app developers/publishers have adopted this advertising format within their apps. Cheetah Mobile[2] has been successful in boosting their ad performance (CPMs) as well as increase the user experience in their apps with Native ads offered by Facebook Audience Network[3].

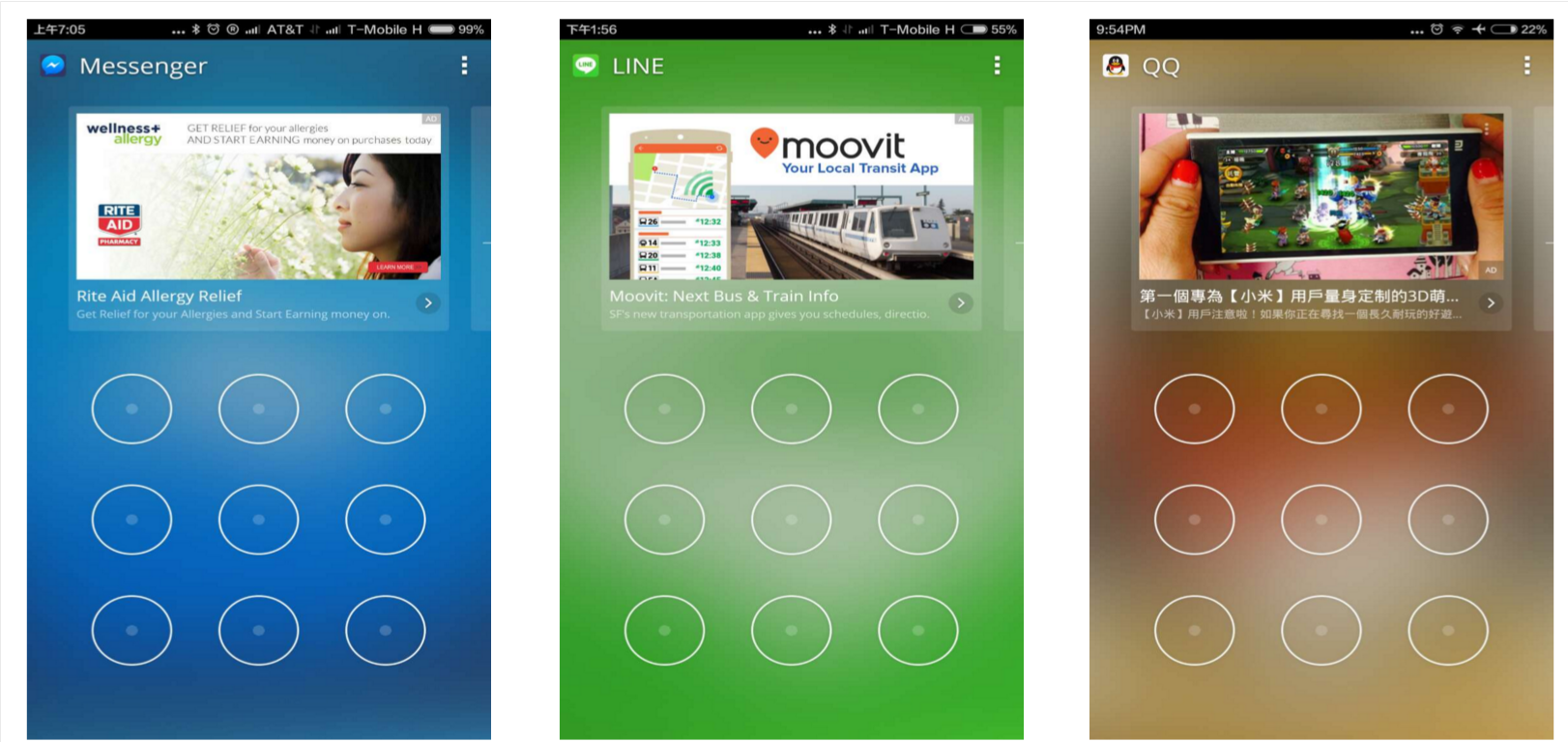


Figure 5 Cheetah Mobile Native Ads

The above image shows the use of native ads with a Cheetah Mobile app.

Other app publishers like Linkedin, Buzzfeed and Facebook itself have worked towards offering Native ads within their applications. By looking at the below image, you will observer that although all the ads look different, they are native ad implementation where in the layout has been configured to fit in naturally with the user interface of the existing app.

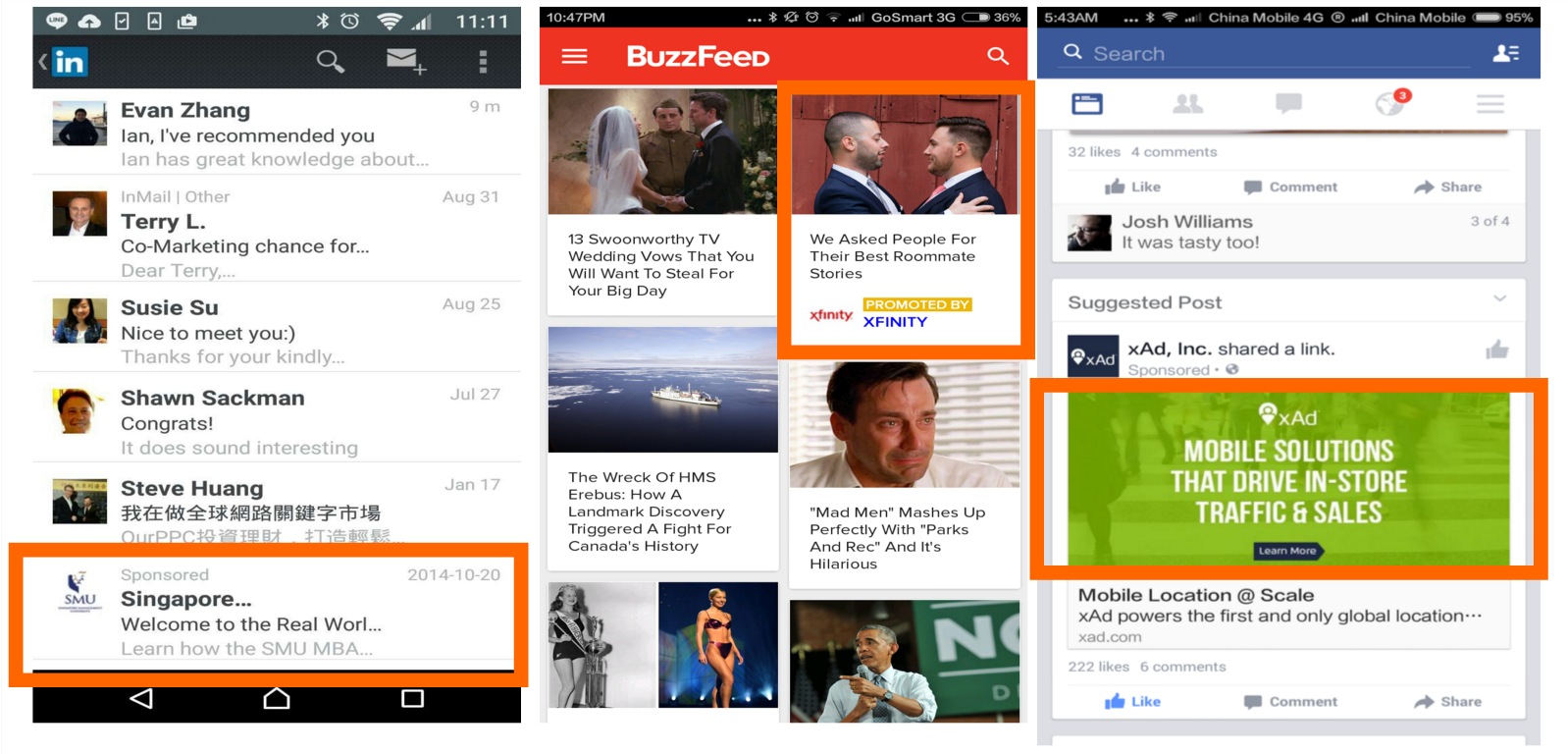


Figure 6 Native Ad examples

**System design**

**1. PicKick**

PicKick is an intelligent photo android client application. It can easily customized to collaborate with a Cloud Server to provide a multimedia service for users. Users can take photos and videos and upload to server with thumbnail maintenance at client side and location-based, time-based, map-based gallery is provided for thumbnail.

When necessary, user can click to download original photos from Server. Meanwhile, panorama feature is also supported.

The technologies has been used are Android SDK/NDK, openCV, Google Map API, Facebook and Yahoo Ads API.

There are 7 stages in our system design. The first stage is **Feature Design**, we consider to implement Photo Taking/Photo Editing/Photo Gallery features. Because users would like to edit their photo’s orientation or size after it was taken and they also like to gallery all the photos categorized on GPS location or taken time or real map markers. The second stage is **Module Design**, we consider using the MVC architecture, which includes UI/UX module, NDK/JNI/API-library module, Mobile Ads module and Google Map API module. The third stage is **UI/UX polish** because we have two times play test and a lot of feedback from real users. The fourth stage is **Construction**, which includes classes/api definition and implementation. The fifth stage is **Feature/Integration Test**. Because the prototype has rough features which need white/black box tests and monkey play test. Then the sixth stage is **Ads Embedding** and the seventh stage is **Play Test**.

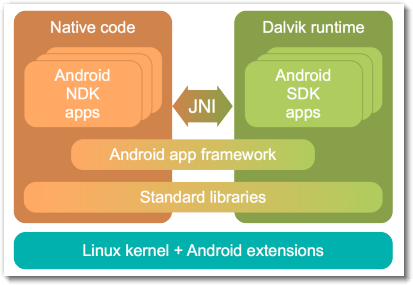


Figure 7 Android SDK/NDK interfaces

**2. What the GIF**

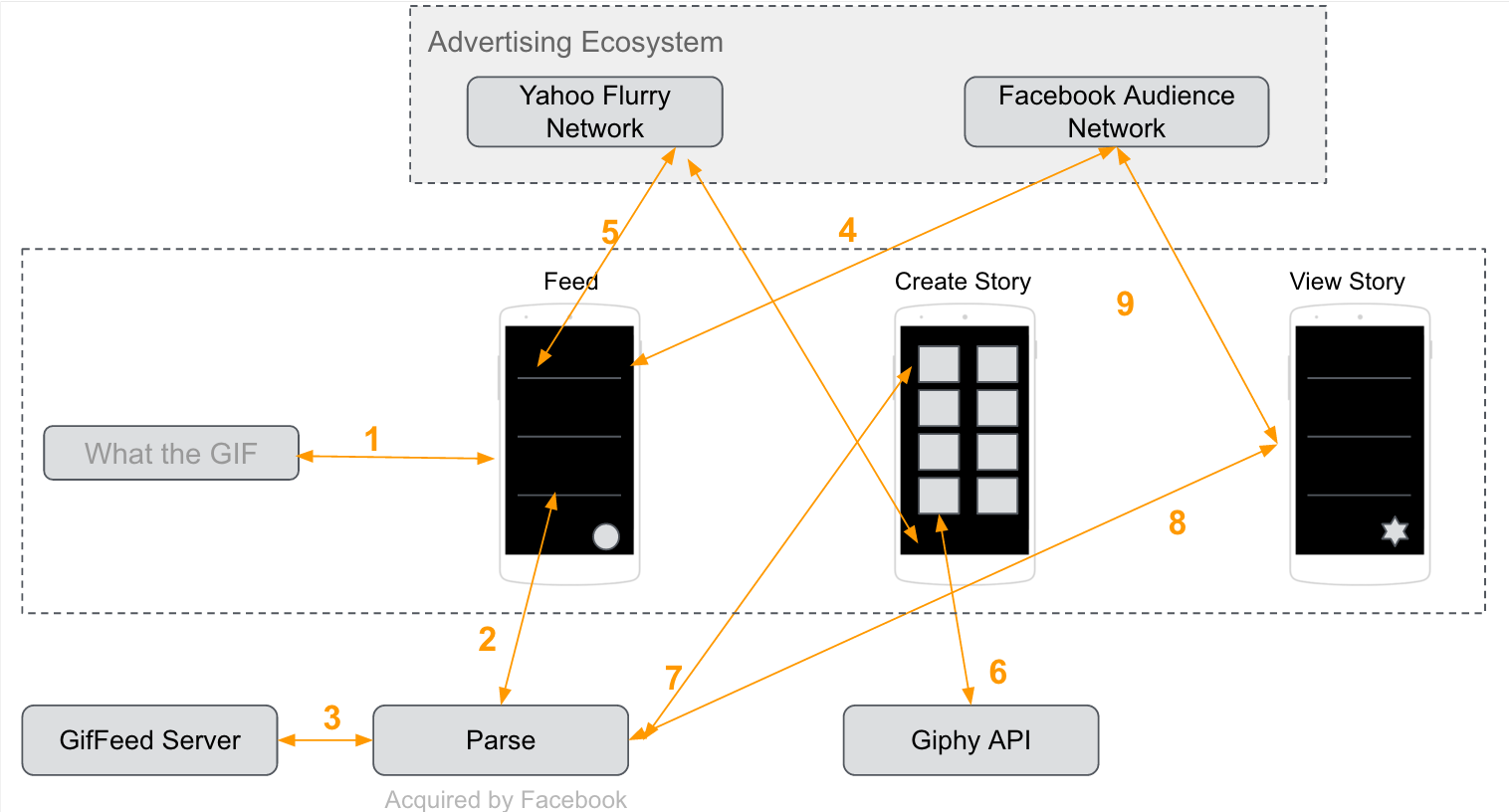


Figure 8 System Design

The above figure shows the system design for ‘What the GIF’ app. The core layers of the system are the ‘Advertiser Ecosystem’, the Android mobile app and the ‘Web Services’.

Advertiser Ecosystem: The Advertiser ecosystem consists of the various advertising network used by the android application to fetch advertisements to display within the application. We have used two advertising networks to help serve ads within the app. First, Facebook Audience Network is an advertising network introduced by Facebook to serve mobile ads to app publishers. They have a mature Facebook SDK to help developers integrate the ads within their individual apps. Using their SDK, we fetched both banner ads and native ads to be displayed within the What the GIF app. Second, Yahoo Flurry Network is an advertising network by Yahoo to serve ads within mobile applications. They also have a SDK to help developers integrate the ads within the application. Again, we used both banner ads and native ads from the Yahoo Flurry Network to serve ads within the application.

Android App: We built ‘What the GIF’ Android app with the intention to provide users with the capability to create fun stories using animated images (gifs). Given the dynamic nature of content, we realized that it is possible to integrate native ads which will blend in seamlessly within the application. Using playtesting, we narrowed down on some of the screens within the application which are ripe for displaying advertisements. Although, the app includes both banner and native ads, we intend to contrast the experience of banner ads versus native ads within our app.

Web Services: The backbone for making the app work are the web services. We primarily rely on the Giphy API[4] for the repository of Gifs. Giphy has made their API available to developers to fetch and display gifs from with their own repository. We have integrated their API in screens which require Gifs to be displayed. In order to persist stories created by the users and to make them available to other users, we have used the Parse[5] object store for saving the stories created by the user. The Feed screen and individual story screen rely on Parse to populate data on the screen.

**System implementation**

**1. PicKick**

We use Android SDK to implement view and widgets and most part of model and controller. Android NDK is used to cross compile openCV to do advanced computer vision algorithm.

The classes we implemented are based on MVC architecture, which include the following classes :

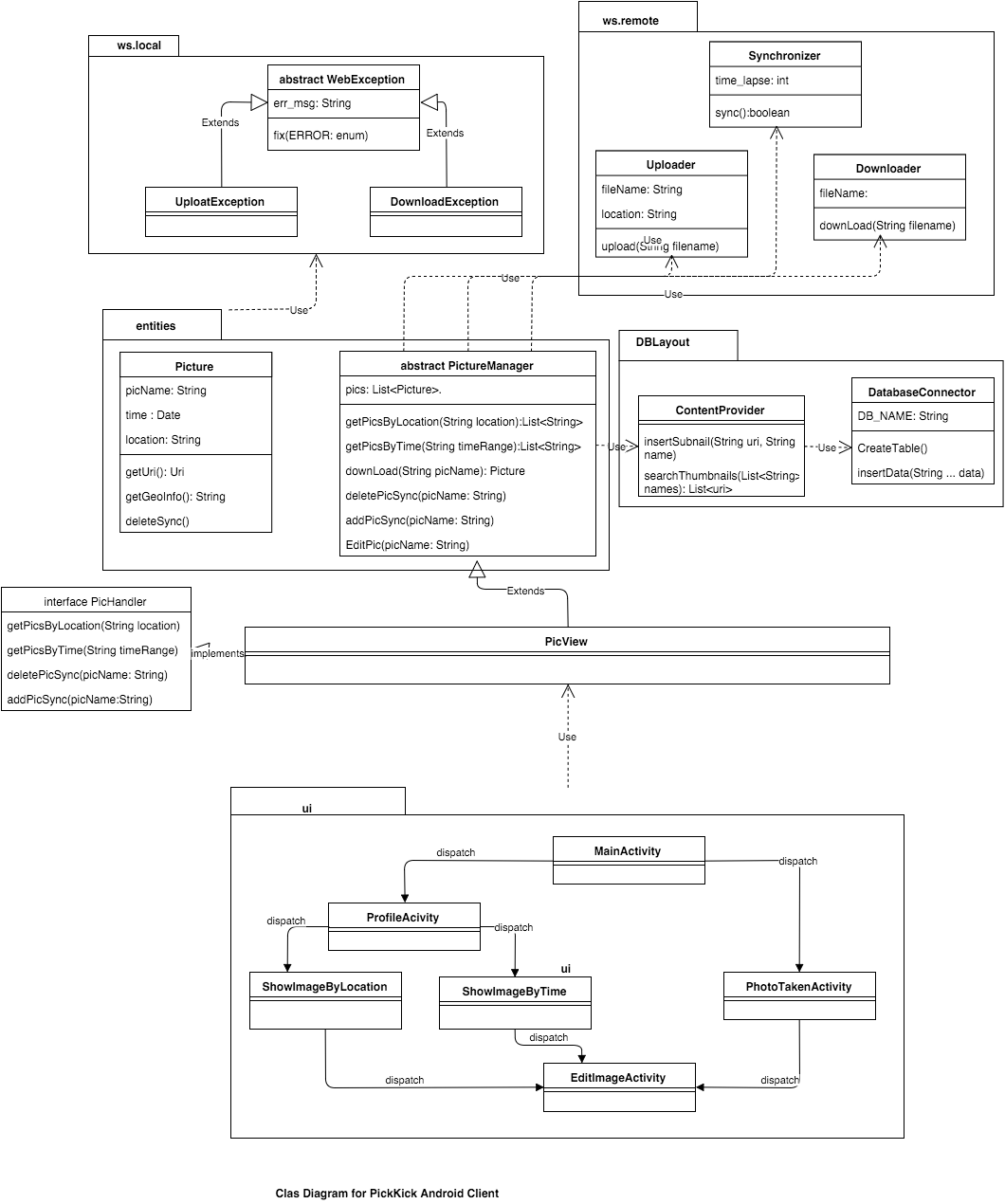


Figure 9 PicKick Class Diagram

The algorithm in Panorama is as follows: Input images are resized and feature extracted and matched by different points. Then images are resized and warped to stitch together to composite a final panorama pictures.

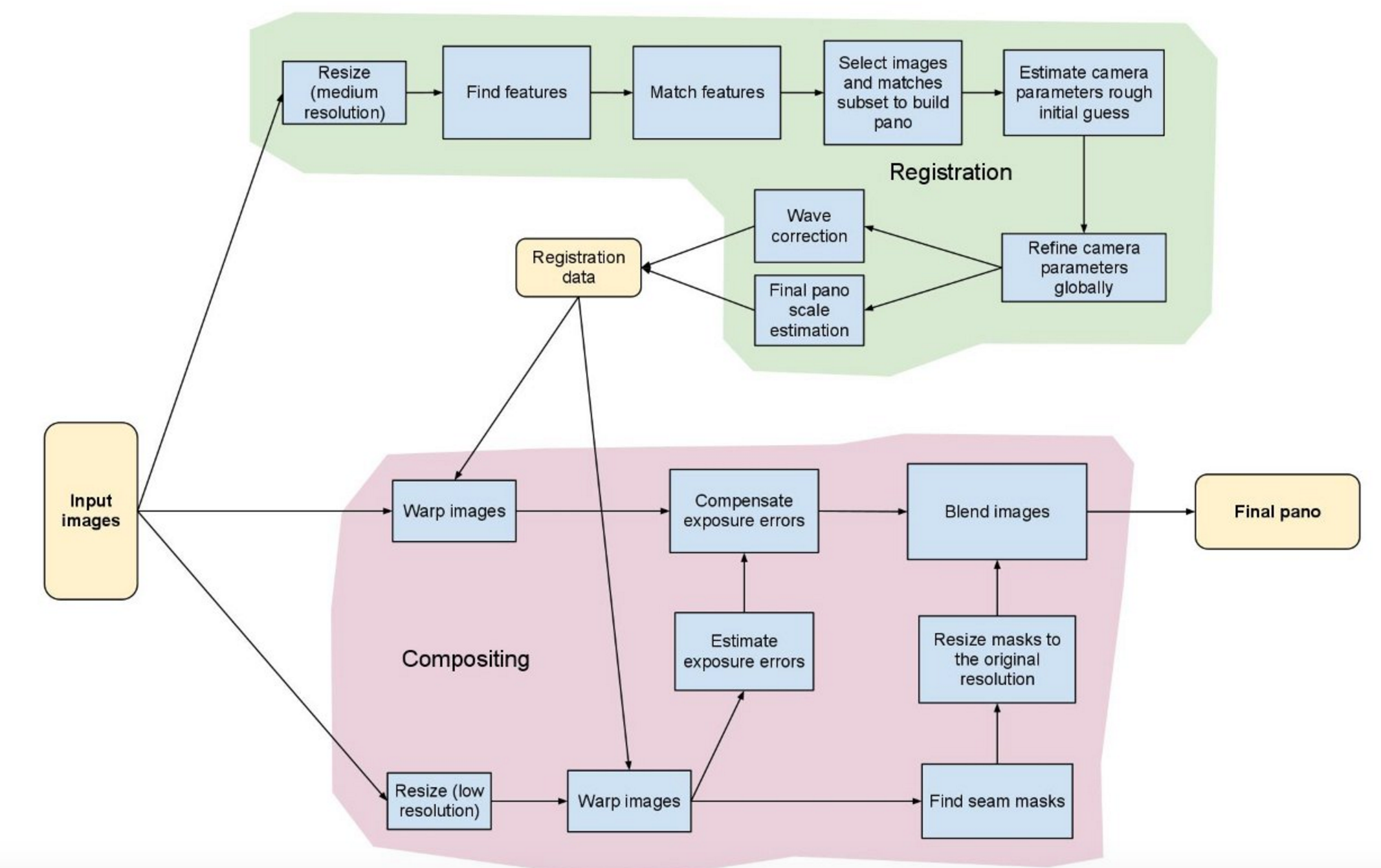


Figure 10 Panorama Algorithm Pipeline



Figure 11 Panorama Field Test Picture

Regarding the Ads, we use Facebook SDK to integrate Banner ads on the login screen, main menu screen and Interstitial ads between the main menu and Google map. And also we use Yahoo SDK to embed Video Ads when users click gridview item to gallery the single picture.

**2. What the GIF**

Figure 8 shows the system design of our app ecosystem. ‘What the GIF’ app has been built using the Android SDK[6] provided by Google. Since our app relies on Gifs to create and show content, we have used the Giphy API provided by Giphy. Using their service, our app is able to access labelled gifs and sticker from their repository. The gif/sticker search functionality also relies on the Giphy API. Search requests made to their endpoint results in a JSON object/array which contains the links to gifs (in various sizes and formats) which are then fetched to be displayed within the app.

To incorporate Ads within the application, we have relied on Facebook SDK which provides access to their Facebook Audience Network as well on Yahoo Flurry SDK[7] in order to serve ads by Yahoo Advertising network. Stories created by users are stored in the Parse object store. We have used the Parse Android SDK[8] to store the story data and to fetch user created stories.

**Experiments and analysis**

**2. What the GIF:**

* **Experiments Timeline:**

The figure 12 below represents the timeline of events leading to the current version of What the GIF. The initial phase prior to version 1.0.0 was devoted to requirement gathering, understanding various monetization techniques and designs for the eventual application. We worked to understand the stakeholders in the ad-monetization process like the publisher, advertiser, ad network, aggregator and their roles in the process. We then designed and conceptualized an application keeping this in mind i.e. an application that seemed relevant to today’s trends & needs and potential ad-based monetization in the app. What the GIF is basically an application that aims to redefine content creation and consumption through the medium of GIF based stories. It is essentially the next generation visual blogging platform and the application is crowd-sourced & open. It provides the necessary tools to create stories using a vast repository of GIF’s & Stickers provided by Giphy[4]. Content creation process is just a few simple clicks. Ad’s provided are relevant in-stream and sync with the app UI and lends itself naturally to the app. The app supports banner ads, Native Image & Video Ads to determine their effectiveness. They are spaced out in different manners based on our feedback & study.

1. The initial version was an internal beta launch and supported the following features: view stories, read stories, create stories, fetch GIFs & Stickers from Giphy.
2. Version 1.0.1 was a play store beta launch that had bug fixes, DB changes, Search feature & Auto GIF Playback.
3. Verson 1.0.2 was an open launch that improved the GIF & UI experience & integrated the Facebook Banner & Native Ads.
4. Version 1.0.3 was a live play testing version. It had improved story rendering, bug fixes, in-stream Native Ads in the story details screen & Fb video Ads.
5. Version 1.0.4 was the second bigger play testing. It had support for multiple Native Ads, bug fixes, Yahoo Banner Ads & faster responsive UI. The play testing was conducted with a much bigger audience (Lots of CMU students)
6. Finally based on the feedback from this version we built a much improved version which is currently live on the play store. It currently has Yahoo Native Ads, Video Ads, improved card based story details view, refresh functionality & search suggestions. Also a couple of art-work in app was re-designed.

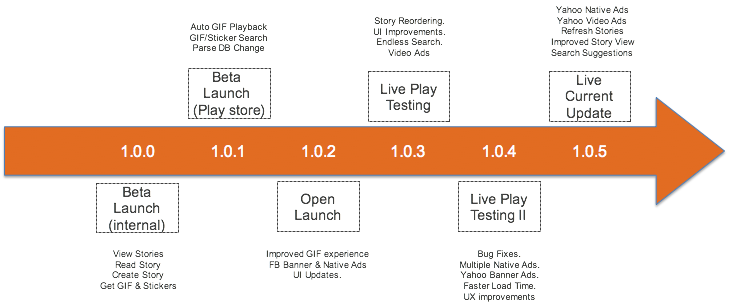


Figure 12 What the GIF Timeline

* **Experiments Analysis:**

Through the course of this project we gathered a lot of useful feedback in terms of the Ad Networks & their performance.

1. Graphical View of Application Feedback

The figure 13 shows a comparative analysis of gradual user increase across each version alongside a gradual decrease in bugs and somewhat stable feature request count in each release cycle.

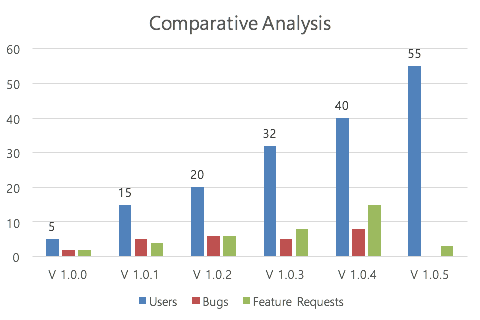


Figure 13 The Comparative Analysis

Figure 14. gives a snapshot of the most requested features in the application. As you can see share was the most requested feature, followed by the ability to create GIFs & finally personalization.

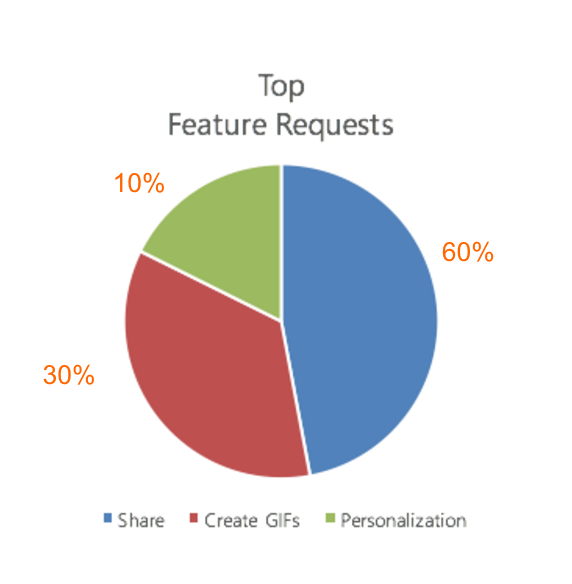


Figure 14 Top Feature Requests on WTG

1. Facebook Audience Network

We extensively used the Facebook Audience Network to integrate Facebook Native & banner ads. It provides targeted ads to the mobile application. [3]The Audience Network essentially requires configuration of various ad placements in the application namely the banner ad placement & the native ad placement for the screens. This is done through the dashboard on developers.facebook.com per application. Once done the placement IDs can be used along with the Android SDK provided to integrate ads. Each placement ID can be optimized for a higher E-CPM or fill-rate or a balanced one, depending on the application needs. The integration process is not too difficult since the SDK is pretty well documented. The code samples are useful for simple cases, but more details needed for advanced integration like multiple ads using an Ad-Manager to render the ads. This is not provided and requires reading the documentation thoroughly to do. Bear in mind a buffer period was needed since the application needed approval from facebook before we could go live.

* FB Ad’s Performance Analysis

The Figure 15 gives you a comparative view of the number of requests made for banner ads versus native ads in the story details. This goes on to prove a lot of people actually went past the main screen quickly into the story details screen. Not too many people waited on top of the main screen waiting for the banner.



Figure 15 FB Ad’s Request Performance Charts for WTG

Figure 16 below gives you a view of the impressions of the 2 types of FB Ads. Clearly again the impressions on the Native Ad are more than those on the banner Ad. This goes on to prove that users spent time at-least looking at the Native Ad far more than the banner ad.



Figure 16 FB Ad’s Impression Performance Charts for WTG

Figure 17 below gives a snapshot view of the clicks and interaction with the 2 Ad types. The interaction with the Native Ads was not bad considering it came after the banner Ads. Additionally there was some feedback on how the native ads fooled the user to be content which potentially resulted in fewer clicks along with the fact that no other content on that screen was clickable.

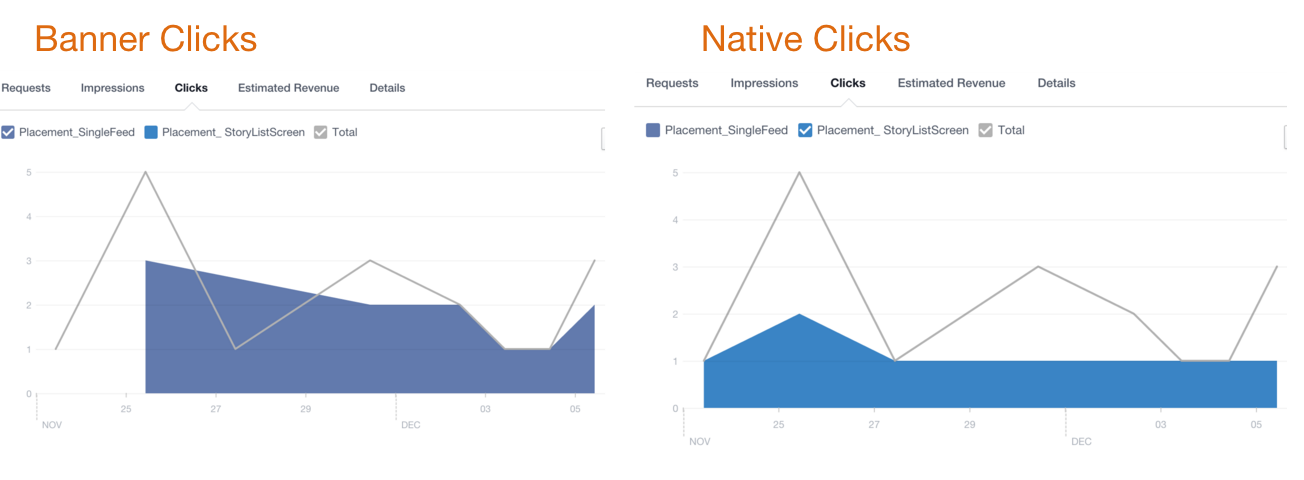


Figure 17 FB Ad’s Clicks Performance Charts for WTG

1. Yahoo Flurry Ad Network

We set about integrating Flurry Ad networks into this application with the expectation of comparing Ad performance with other network. The flurry network also requires configuration of the application on their web dashboard. We declared placements for the bottom banners and Yahoo Native & video ads to use in stream. Flurry provides its own SDK in the form of a JAR file. It has heavy dependency on the Google Ads Library and requires the user to provide a bunch of permissions. The integration is pretty tedious because the documentation is pretty outdated in terms of steps. Also it does not use some of the best practices hence making it difficult to use in newer apps using latest features. We ended up using Yahoo Ads for displaying bottom banner ads in the story creation screens namely the GIF, Stickers and search screen. The Yahoo Native and Video Ads are part of the main Story List screens seen on app launch.

* Yahoo Ad’s Performance Analysis

The figure 18 below has critical portions to it. The left half gives a snapshot of the performance of our Yahoo Ad Placements. As you can see the Story screen native Ad based on feedback had decent interactions and a very high e-cpm. The banner ads from yahoo on the search & GIF/Stickers also had good impressions, fill rates and request. The top personas of the users based on the analysis was social influencers, bookworms & geeks. Also the majority of our users interacting with Yahoo Ads were males. These were some interesting insights.

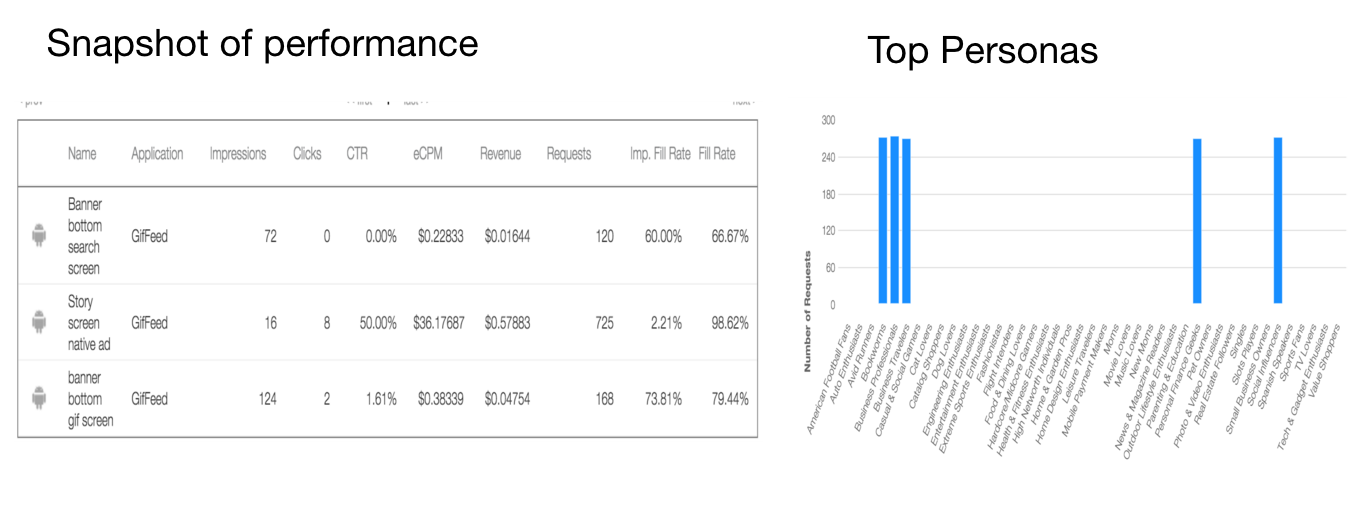


Figure 18 Yahoo Ad’s Performance Analysis

1. Ad Placement Analysis

The figure 19 below gives you an analysis of the various placements. As you can see the banner Ad at the top of the home screen was reported by users as a very annoying and intrusive experience. Likewise the Bottom banner on the GIF/Search screens was in-fact very effective and got significant transaction from the users, which was not expected by us.

The Native Ad integrated in the story details was expected to provide a great in-stream blending experience, but many users reported that it fooled them to be part of the content and did not realize it was clickable. Users requested that content be clearly labelled as sponsored and identified as clickable. Some Native Ads also were irrelevant to the content. Finally. the users requested the ads be spaced out evenly in the content stream. Another request was to keep the content creation process Ad-free.

We adhered to some of the above feedback and noticed a change in the interactions and impressions.

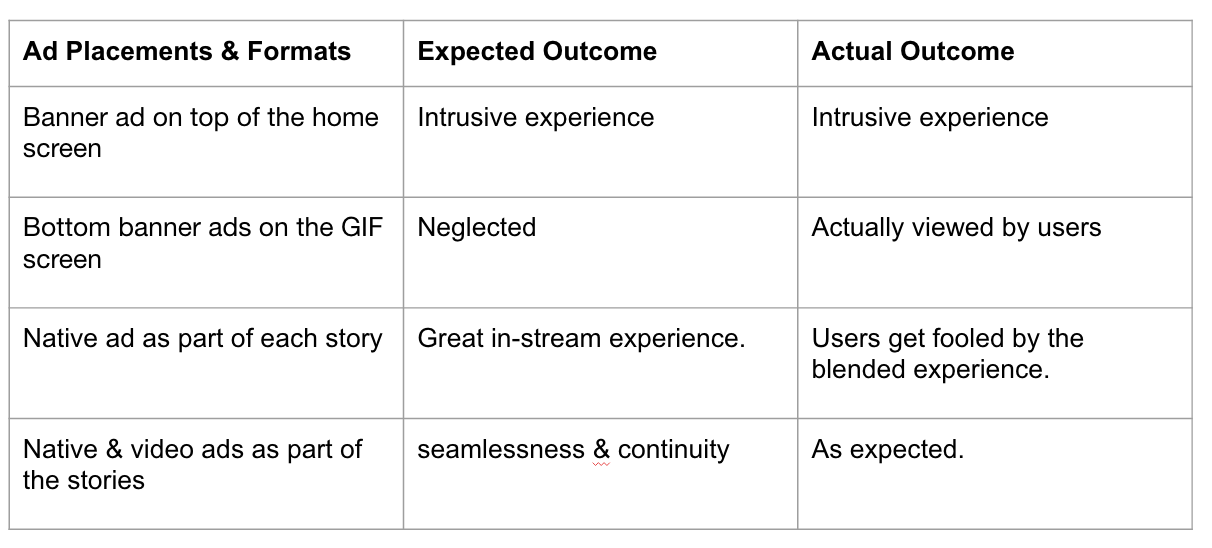


Figure 19 Analysis of the Various Placements

**PicKick:**

* **Experiments Progress:**

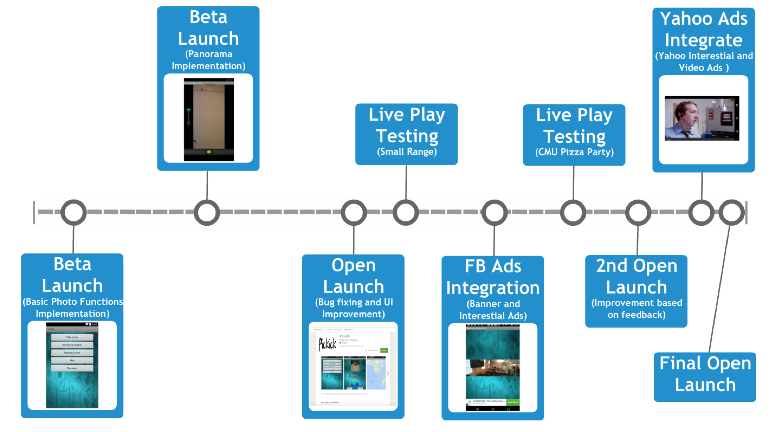
To finish our project Redefining Mobile Advertising, the PicKick went through the totally 9 phases. In the first two months, we devoted to implement the basic functions of the PicKick. The PicKick is a mobile tool to manage the photos’ storage. The user can take basic photos and panorama photos and store them on our cloud server. The system allows users to see the photos based on the time and location, and the Google map view provides a clear view of the image location.

Figure 20 The experiment timeline for PicKick

On the first phase, we implemented all the basic functions and allow users to take and store images. Then the second phase, we focus on integrating the Open CV panorama function into our system. After finishing the first version app, we applied for the Google Developer account and released it shortly after fixing the bugs and improving the UI. Then we conducted a small range play test and collected some feedbacks about the UI and page flow logics. From the beginning of the fifth phase, we began the Facebook Audience Network integration. We integrated two placements in the system, the Banner and Interstitial Ads. Then we launched second round live test quickly with a large range among CMU students, faculties, staffs and Cheetah Mobile sponsors. At this round, we collected lots of useful feedbacks and found quite a lot points to improve. On the final two rounds of the development, we fixed the issues and integrated interstitial video Ads of Yahoo Flurry into our system. Currently, we release the third version app to the Google Play Store in the final open launch.

* **Experiments Analysis:**

Based on our experiment, we collected a lot of the data about different Ads platforms and formats. The following section will present and discuss these data.

1. Ads Platform Comparison:

* Facebook Audience Network:

Facebook Audience Network is provided in the Facebook Android SDK. For android developer, it is pretty easy to integrate Facebook SDK. The Audience Network platform provides the generated codes for developer to use directly. After choosing the placement display format, the facebook will generate the codes based on your situation. The user can insert the generated code section directly without any problem. Therefore, using audience network platform can save the developers lots of time.

Besides the convenience for developers to integrate the Facebook SDK, the Audience Network has a really easy and clear tool for developers and business managers to monitor and measure the performance of the monetization. The performance page provides the line chart and table for users to view the Requests, Impressions, Clicks, Estimated Revenue and Details of different placements. Besides, users can also category the throughput based on country and date.

Although the Facebook Audience Network has really excellent usability, it still has very strict authentication process which may make developers feel inconvenient. Normally, developers need to wait for about 3 days to 1 week before they got passed, which delays the developers developing progress.

The Facebook Audience Network was our first goals to integrate, at the fifth phase of development, we successfully finished integrating the Facebook SDK.

* Yahoo Flurry Network:

Like the Audience Network, the Yahoo Flurry Network is another popular Ads Network in market nowadays. It also enables the code generation function for the developers. However, the generated codes is less helpful compared to the Facebook Audience Network. Considering only core codes are generated, developers still need to understand the tutorials and put the core codes on the right place.

The advantage of the Yahoo Flurry Network is that the platform provides very functional and diverse tools for business managers and developers to monitor the performance. The users can analyze the data from the perspective of Usage, Retention, Audience, User Acquisition, Events, Errors and Technical[7]. The report is very detailed and developers can respond and improve the Monetization strategy very quickly based on these details.

On the eighth phase, we followed the instructions and mapped the Yahoo Interstitial and Video ads to the PicKick.

* Twitter Mopub Network:

The Twitter Mopub also has a very wide users base. The platform is especially good for the enterprise users, since there are plenty of tools to measure the performance of the Ads strategy. However, the SDK integration experience is really bad. During our integration process, we found many mistakes and material omission exist in the tutorial. The downloaded sample is not able to run because of the build path problem happening.

1. Facebook Ads Performance Analysis

After integrating the Facebook Audience Network SDK, we kept collecting the data in half the month. The data is presented as the following diagrams:

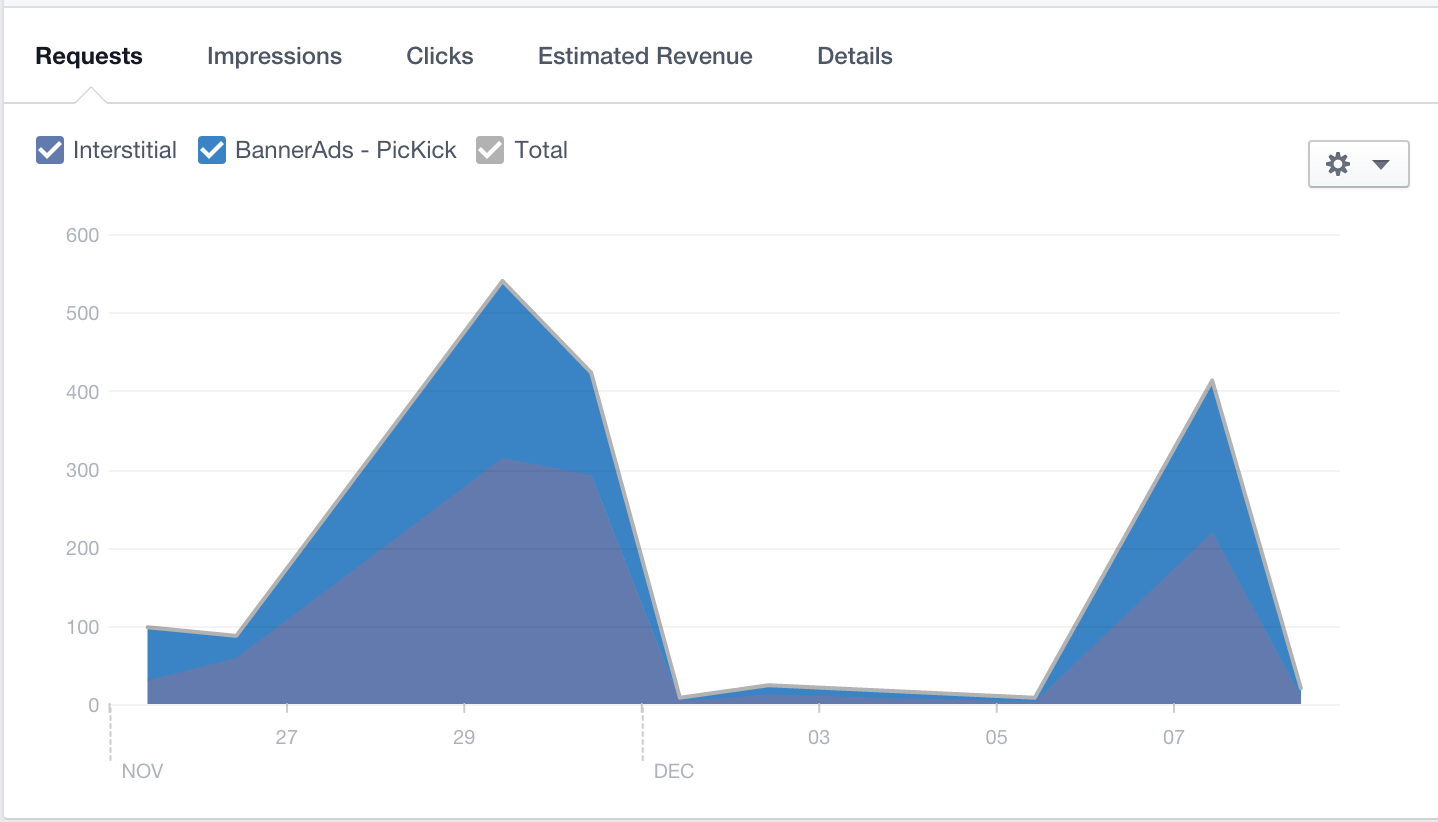


Figure 21. The Request Throughput of Facebook Interstitial and Banner Placements

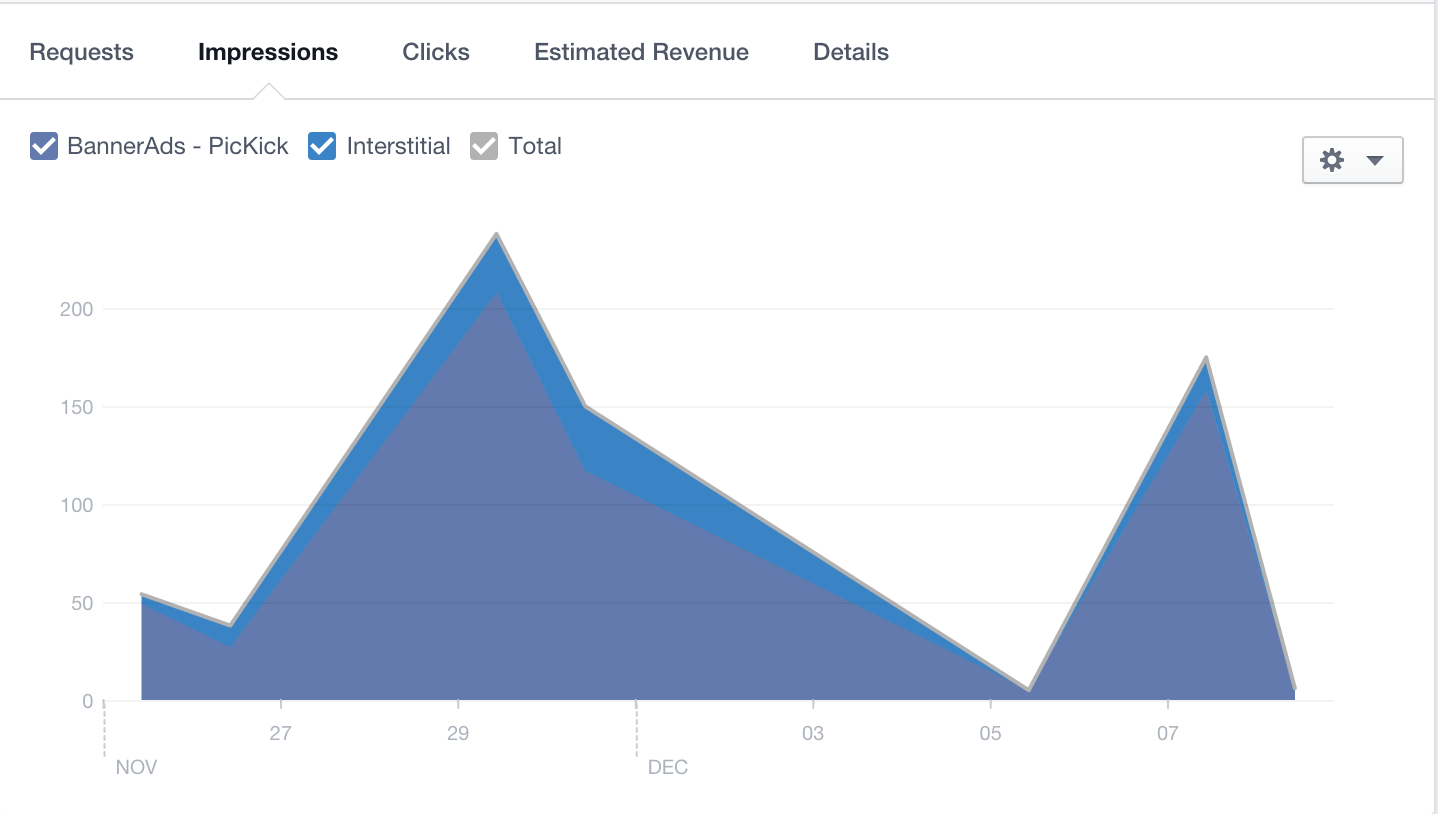


Figure 22. The Impression Rate of Facebook Interstitial and Banner Placements



Figure 23. The Click Rate of Facebook Interstitial and Banner Placements

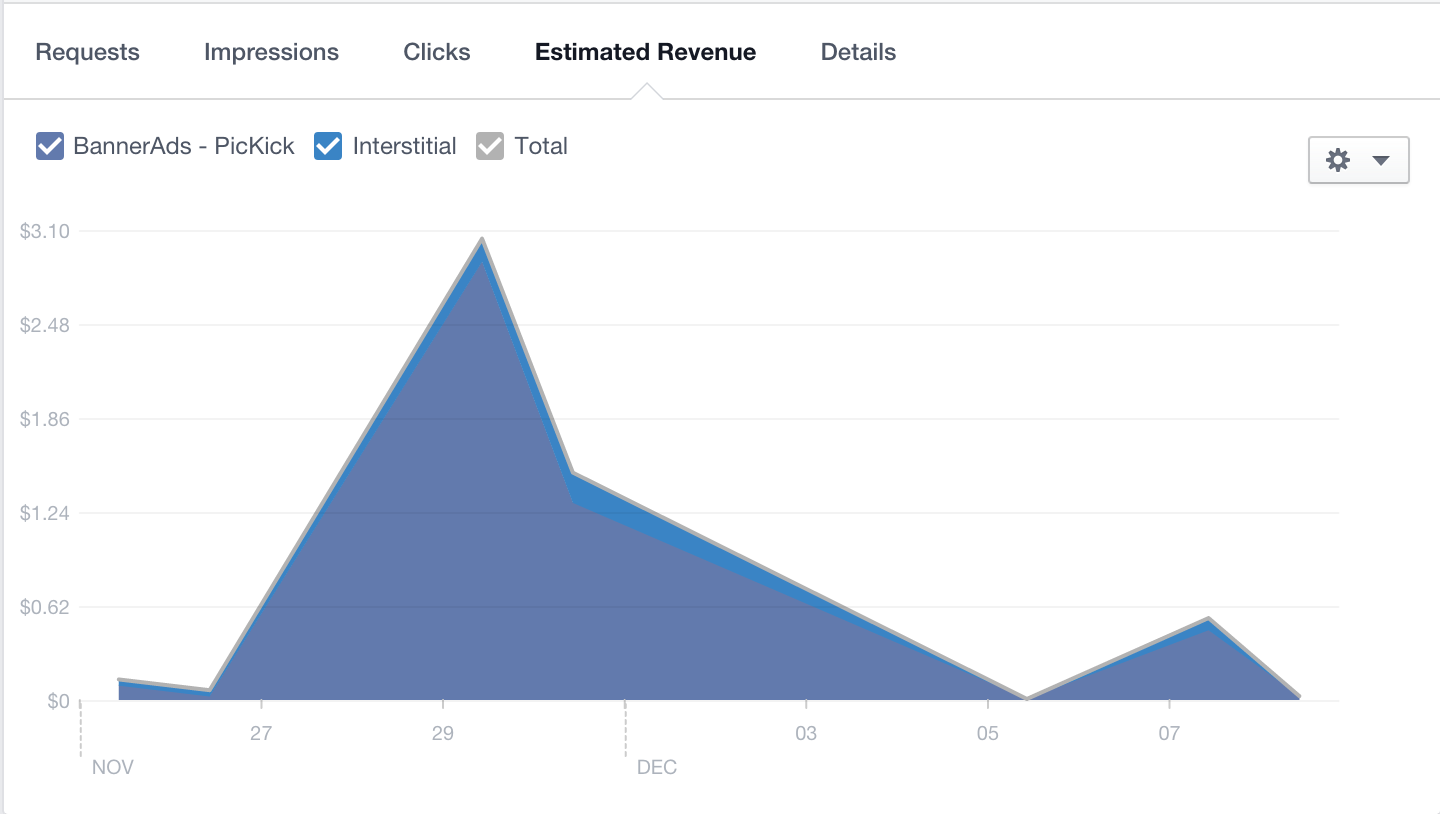


Figure 24. The Estimated Revenue of Facebook Interstitial and Banner Placements

From the data we collected, we can see the banner ads’ request is less than the interstitial ads. That’s probably because we put most banner ads at the main page and login page, which only requests once and lasts for a long time. However, the interstitial ads is in the transition from the main page to map page, if the users like to view the photos on the Google Map, more interstitial request will generate. This result can be confirmed by checking the impression rate and click rate of the Banner Ads and Interstitial Ads. The banner ads create much more benefits than the interstitial ads. The data of impression rate and click rate means the users give much notice to our bottom banner ads. From the Estimated Revenue data, we can also confirm that the revenue relies on more impression and click rate instead of the request.

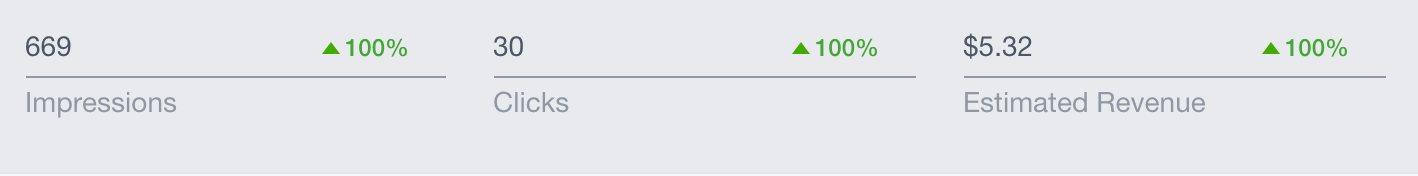


Figure 25. The Performance of Facebook Revenue of PicKick

The above figure shows the total revenue we collected for half a month, the estimated revenue reaches our expectation considering our customer base is not very large.

1. Yahoo Video Ads Analysis

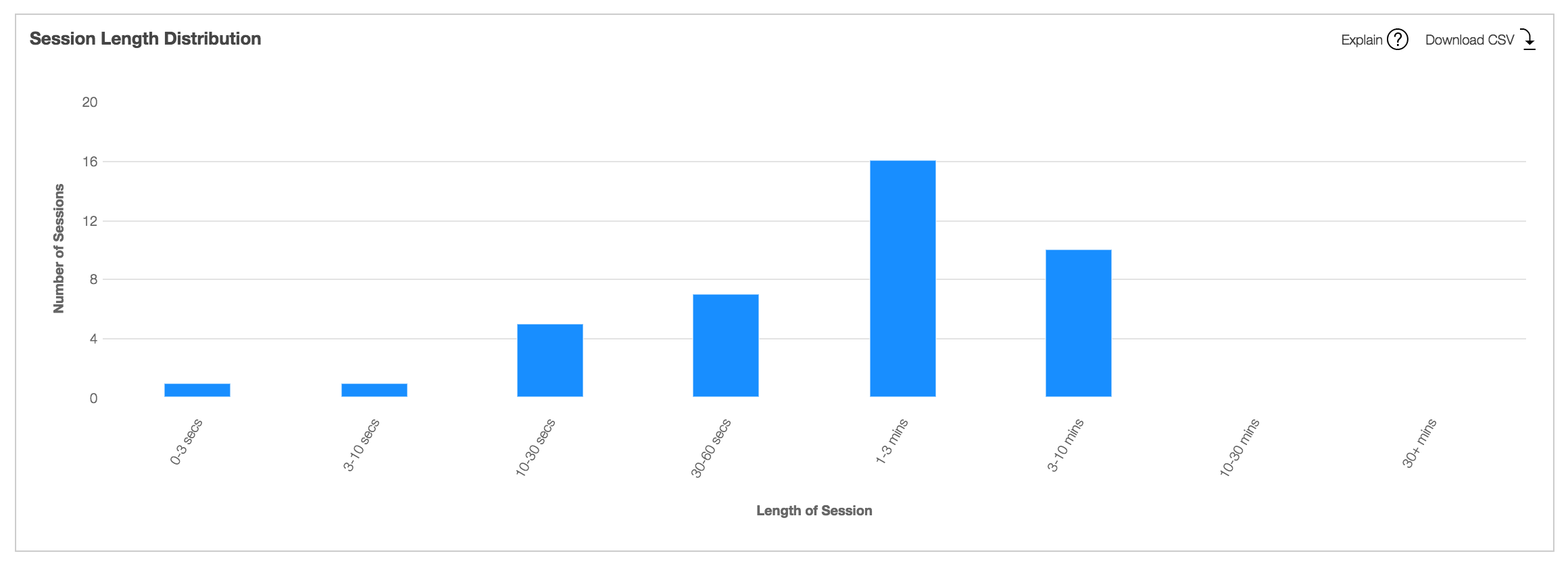


Figure 26. The Session Length Distribution of Yahoo Interstitial Video Ads Placements

We also collect many yahoo data, and the most special one is the interstitial and video ads’ session length distribution. Although we set the user is only possible to skip our ads after 15 seconds, it is really surprising to know most users will stay to see the ads for 30 seconds to 3 minutes. Based on this result and a long time researching, we can assume that the video ads attract many users because of its interesting. Compared to the Facebook’s image interstitial ads, the video interstitial ads apparently is a better ads strategy.

1. Live Playing Test Feedback

On our sixth experiment phase, we hold a pizza party to have large enough test users base. The feedback of this play test can be summarized as following:

For PicKick:

* The efficiency of Panorama function needs to be improved, a lot users complained that wasting them times to wait
* We should improve the location category precision so that fewer image will be overlapped

For our Ads integration

* The interstitial ads sometimes may confuse the users. Considering this perspective, Yahoo has better user interaction than Facebook since the user can still see the content of the app through the thin image cover.
* To our surprise, most users don’t think the bottom banner as impact their actions.
* The cooperation of interstitial ads and geo map will request more memory and sometimes crash the app in many old version devices because of the out of memory issue.

1. Advertising Result Summary

According to the data we collected, we carefully analyze the result and create the following summary for the PicKick’s future Ads strategy:

* Considering most of our users don’t mind the bottom banner ads, we should keep them as they are and believe they can cause more impressions.
* The image interstitial ads will confuse the users and cause some memory leaking problem, it is not a good idea to adopt them on the transition between two pages.
* The interesting video ads can attract many users’ attention and increase the impression rate.

**Conclusions and future work**

Through the whole practicum process, we get the following conclusions:

* The native ads have better in-stream experience and it will become the future of the Ads market.
* To enhance the usability of the app, the advertisement content should be really relevant to the app content.
* The structure of the ads and app content can be similar, which would not influence the user actions, but the user should be able to realize the which part is the advertisement.
* The banner and interstitial ads, especial the image interstitial ads will be intrusive and impact the flow and user experience.
* The ads integration should not be overwhelming, 1-2 ads in the same page is reasonable.

By comparing the different platforms and formats, we truly feel the ads market will be promising in the future for the mobile app monetization. There are three suggestions we think the developers should take into considerations in the future.

1. The Ads can be improvised and changed based on different user’s retention and user category, which attracts more users.
2. To explore more benefits, the developer should consider optimizing the ads strategy. The sandwich model is a good option to adapt. By using this model, developer can combine different Ads Networks and amplify the benefits of the monetization. Besides, the Mopub’s real-time bidding method is also good to use for the Ads publishers to monetize their apps more effectively[11]. The bidding process will cause more eCPM for publishers because the ads networks or DSPs will be willing to add the price simultaneously.
3. Some incentivized ad networks which can give the rewards to users are also possible to attract many users. Nowadays, Kipp Rewards, NativeX, Supersonic Ads have found the potential of this field and provide the relevant functions. Developer can consider these networks to earn more revenue by attracting more users to click.

**References**

[1]<http://www.emarketer.com/Article/Mobile-Ad-Spend-Top-100-Billion-Worldwide-2016-51-of-Digital-Market/1012299>

[2] <https://developers.facebook.com/docs/audience-network/case-studies/cheetahmobile>

[3] <https://developers.facebook.com/docs/audience-network>

[4] <https://api.giphy.com/>

[5] <http://parse.com/>

[6] <http://developer.android.com/sdk/index.html>

[7] <https://developer.yahoo.com/flurry/>

[8] <https://parse.com/docs/android/guide>

[9] <http://docs.opencv.org/2.4/modules/stitching/doc/introduction.html>

[10] <http://www.android100.org>

[11] <http://www.mopub.com/wp-content/uploads/2014/09/Pub-How-WhyRTB-One-Pager-3.pdf>

**Appendix:**

1. **GitHub Links:**

* **PicKick: https://github.com/bbfeechen/MobileAds**
* **What the GIF: https://github.com/JW-Vinay/GifFeed.git**

1. **Access information:**

* **PicKick:** [**https://play.google.com/store/apps/details?id=avator.pickick&hl=en**](https://play.google.com/store/apps/details?id=avator.pickick&hl=en)
* **What the GIF:** [**https://play.google.com/store/apps/details?id=com.giffedup&hl=en**](https://play.google.com/store/apps/details?id=com.giffedup&hl=en)

1. **Please see our GitHub for more required documents.**