Tech Tips For Building Your App

So you have your great idea and you're sitting down to start writing your app. Where do you begin?

The Android training guides are a great resource for beginners and experts alike. These guides range from building your first app to supporting different devices, as well as best practices for improving user experience and optimizing for performance.

For iOS, there are also design resources available to help you optimize your applications to conform to the latest version's design best practices.

A great supplement to these guides are live presentations that discuss various coding designs and tips for building quality apps. In particular, the DevBytes series provides easy to consume bytes of information on important concepts in Android and iOS. The Android Design in Action series highlights user interface best practices from real applications, and links to samples that show how you can implement these best practices in your own application.

User Interface Best Practices

Here are some things to think about when developing your application's user interface.

Make the app open quickly. When a user opens your app, they want to interact with it right away. Ad Mob research shows that 72% of users in the US, and 78% of users in China believe a good app is one which opens quickly, and ranked this as more important than other characteristics of an app.[15] To accelerate loading, only load resources that you need for your main screen, and load other resources on a background thread.

If you plan on showing a full screen interstitial ad when the app opens, only wait for a couple of seconds. If the interstitial doesn't run fast enough, just move on without showing one.

Use high quality graphics. Users value applications with high-quality graphics. In five major markets we surveyed, daily app users ranked 'quality of graphics' as the most important feature of game design.[16] One simple improvement is to include higher resolution images for higher density devices, so the resources look sharp on every device.

Build for Multiple Devices. Think about what your app should look like on a phone versus a tablet. Consider including different XML layout files for different screen sizes on Android, or separate xib layout files for iPhone and iPad.

Build for Multiple Versions. Your app should be consistent with the design of the platform. For example, iOS6 relied on skeuomorphism (design imitation of real-world objects) whereas iOS7 adopted a flat design. Apps previously built for iOS6 should be updated to follow iOS7 design guidelines for users running iOS7, to provide a consistent experience across all apps. On Android, you'll want to maintain compatibility with some older versions in order to reach a broader audience, but also take advantage of new design patterns to cater to newer devices.

Use Density-Independent Units. On Android, different devices have different densities. Use density-independent pixels (dips) instead of raw pixels when defining the size of your view, so that the view is the same physical size on each device.

Storing Application Data

When designing your backend to store your application's data, think about what type of data you're looking to store and which storage mechanism best handles that data. Is the data structured? Should it be readable by other apps? Are you only looking to store a few settings across user sessions? Answering these questions will help you choose the best native storage mechanism(s) for your app, like these for Android and these for iOS.

An alternative to storing data on the user's device is to store it on servers outside of the application. Moving data to an external server allows you to build a single backend for multiple platforms, as well as track the user's status across different devices if you have a signed-in user.

You can choose to run your own server, or integrate with an Infrastructure as a Service (IaaS) solution to help you reliably scale your application. Once such service is Google Compute Engine, and there are other solutions in the market. If you want to take advantage of this scalability without having to worry about managing the implementation details of IaaS, you could use a Platform as a Service (PaaS) solution such as Google App Engine. These services make it easy to set up a common data storage solution for all versions of your app.

Permissions

Imagine you're about to download a note-taking application when you're asked to enable location access. Or how about access all of your contacts? Would you download it? Probably not. Some permissions are more sensitive that others, so only request permissions that are absolutely necessary to run your application. You can ask your friends to download your app and tell you if the permissions you're requesting are reasonable.

Location - Applications should NOT ask for location access for the sole purpose of providing better ad targeting, as it may discourage users from downloading your app. Of course, an app that lets you search for restaurants in your area may already need access to location, and can then also pass this information to request more targeted ads.

Contacts - Many users are afraid to give apps permission to their contacts because they're afraid that the app could spam all of their friends. Make sure to have a convincing use case before asking for this information.

Device IMEI - Not only are device-specific identifiers sensitive, but they also aren't great at identifying a single user. If the user has a phone and a tablet, representing a user by their device ID means you treat this user as two different users depending on the device they're operating. A better implementation would be using Google+ or similar sign-in mechanism to log in a user, or use the Advertising ID if you're just using this identifier for advertising.

"

> App Engine enabled us to focus on developing the application. We wouldn't have gotten here without the ease of development that App Engine gave us.

BOBBY MURPHY
CTO and co-Founder,Snapchat

Testing

Before you release your app, you should test on array of devices and OS versions and sizes. On Android, there are some OS differences in Samsung's Galaxy devices compared to the stock Android Nexus devices. And devices like the Kindle Fire have a very different Android experience. Even on iOS, the newer iPhones are a different size. You'll want to make sure your app looks and runs good on all devices.

You may also consider having a beta program for a subset of users to try out a new version before it's publicly released. This way you could address any reported bugs before launching worldwide.

Once your app is out in the wild, you may be stuck on what to work on for the next version. The comments left on your app can be a good place to start. If users are reporting bugs or crashes, address those issues to satisfy your users before working new features. Even if you're getting conflicting feedback, determine what aspect of the app has the most volume of comments, and spend more time thinking about how to best

RECOMMENDATION

If you are planning to go alpha- and beta-testing route, it's best to start with a small group of alpha testers (perhaps employees of your company, or a few fellow developers) and then move on to a larger group of beta testers.

Through alpha- and beta-testing functionality, Google Play Developer Console lets you test two different versions of your app at the same time in addition to your production version.

App Development Tips

Abide by design best practices

Follow the Android training guides and iOS design resources to conform to the platform's latest design best practices.

Build your app for multiple devices. Use different layout files for phones and tablets and take advantage of the additional screen real estate.

Storing Application Data

Figure out what your data storage needs are, and select the appropriate data storage mechanism.

Consider using an external server so you can build a single backend.

For scalability, use an IaaS or PaaS solution.

Snapchat uploads hundreds of millions of photos daily using Google App Engine.

Minimize permissions usage

Only ask for permissions that are necessary for your app

Don't use Device IMEI to define a unique user - use the Advertising ID if you're only using this ID for advertising.

Perform extensive testing
Test on array of devices and OS versions

Consider having a beta program prior to a full launch

Check your app's comments to help guide future development

Increase app engagement and installs with Google+

Over-the-air installs

If you have a website, when users sign in with Google, they can be prompted to download your app directly to their Android device. Set up is easy: you simply register your project and app clients via the Google APIs Console to link your mobile app to your web application.

Over-the-Air Installs provides an effective growth vector on Android. When a user signs in with Google, we detect eligibility (own an Android device + don't yet have your app), and then bring up the Google Play screen. On average, 40% of users download the app during registration.

Google+ Sign-In

Are you seeking an easier way to grow users with hassle-free, one-click sign-in? Google+ Sign-In is a simple, trusted way to let people sign in with their Google credentials.

With minimal effort, you can add a trusted registration system that's familiar to users and consistent across devices. Google+ Sign-In lets users skip high drop-off registration forms and avoid having to remember yet another username and password. This makes registration faster and more secure, and reduces support costs for forgotten usernames and passwords, and you don't need to build your own authentication system.

Learn how music app Songza grew their user base with Google+ Sign-In in this case study.

CASE STUDY

Songza and Google+ Sign-In

A trusted sign-in option lets Songza users discover the right music

Songza is a music app that streams just the right music at just the right time. Whether working, relaxing, or at the gym, Songza users hear an expertly-curated playlist that fits the moment. By enabling Google+ Sign-In, Songza increased their registered users across all platforms.

Increased registrations across all platforms

Since launching Google+ Sign-In, daily registrations have increased across all platforms and sign in options. Google has quickly become the most popular sign-in option for Songza users on Android, with 40% of users choosing Google. Google+ Sign-In also helps increase Android distribution. 37% of web users opt-in to instantly over-the-air install the app on their devices, leading to an improved, consistent experience across mobile and desktop.

Listening to the right music at the right time is now easier

Google offers people a simple and secure way to register with Songza by authenticating with their Google credentials. Songza has added app activities into Google Search results, so people can see the most popular playlists being listened to by Google users on Songza. Surfacing interesting, popular content from Songza in Google Search provides more opportunities for new users to discover the service.

> Implementing Google+ Sign-In has significantly grown Songza's user base and has enabled easy discovery of popular app activity in Google Search.

ELIAS ROMAN
Co-Founder & CEO, Songza