



How to stand out from the crowd

There has to be something in it for the user.

Your customer has to gain something from using your app. Whether they get a laugh, wisdom, or efficiency, there has to be something in it for them.

Keep your features list limited.

You are designing for a portable and mobile device. Just a couple of main features does the trick.

With *Gratitude Journal*, I was tempted to add features like ambient music and graphs. Those things weren't core to the main functionality and only add to development cost and time. Dumping those things was a smart move.

Don't design for everyone.

Select your target audience and keeps your design simple and aligned specifically for that group⁶.

I was tempted to make *Gratitude Journal* a basic run-of-the-mill journal, thinking I would have a wider customer base. But there's nothing special with just a journal.

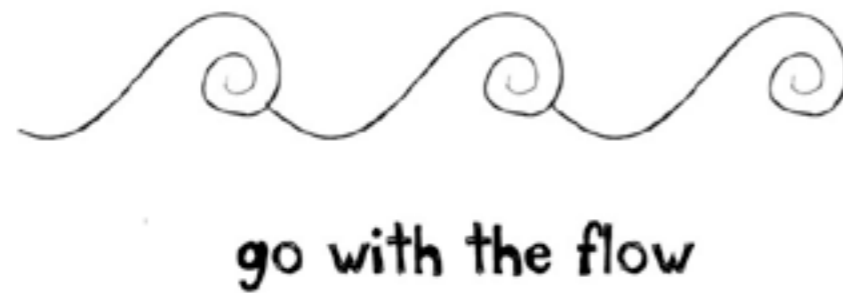
Thankfully friends kept me on focused on staying within my niche. In the end, my app stands out from the rest of the journal apps.

Surprise your users.

Sneak in a little unexpected extra for your users. Something simple and fun: it can be an animation, feature or style. Just keep it subtle.

My quotes feature was my subtle surprise. After creating a journal entry, the user is rewarded with a quote for the day. This little feature has received an amazing amount of positive feedback. Reward your audience and they'll reward you back.

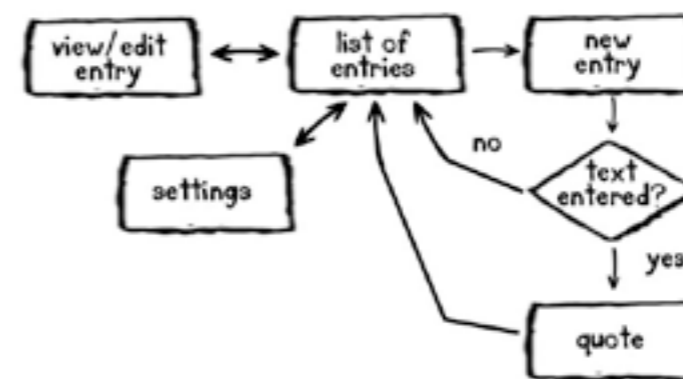
⁶ Learn more about "tribes" in *How to Successfully Promote Your App*



Create a screen flow diagram

A simple diagram outlining exactly how each screen links to the others forces you to think about the app's fluidity. It's also valuable information that your developer will thank you for.

Below is the final flow diagram I created for *Gratitude Journal*. I normally use a tool called Omnigraffle to create my screen flows but you can also use Powerpoint, Google Presentation or OpenOffice Draw. Anything that lets you draw some boxes and arrows will work.



Even if you only have a couple of screens, this one image will save you a lot of explaining in your specification. Pictures are always far more powerful than paragraphs. Not only are they quicker to create, they translate well into any language. Your developer will thank you for it.

➡ ACTION

1. Create a screen flow of your app.
2. Paste it next to your Product Definition Statement
3. Add it to your design specification.



Picking the right developer

Look at their prior work

Start with requesting to see their portfolio, if they don't already have examples of their work on line. Be sure they tell you exactly what they did for each project in their portfolio because it might be just a little update on the app rather than build the entire solution.

Also ask them for sample code you can review. Even if you can't make heads or tails of it, you want to make sure they know how to develop on the iPhone and aren't using your money to learn the very basic first steps.

If you have a friend who understands Objective-C, have them take a quick look at the code to make sure it's clean and well documented. Be sure to praise your friend publicly for their help after you go live.

Ask about their current backlog

Find out how many projects they have going at a given time. They may not be completely honest, but you have to ask. You don't want your project to be the neglected step child because it's not the highest earning project for them.

The number of projects depends on the number of developers on the team. If there's just one developer, two or three projects is the most you want them dealing with at one time. Any more than that, and you are looking at long delays.

Ask for their start date

Make sure they can start a week after all contracts and agreements are finalized. Excitement for a new project has a limited shelf life, so you want to tap into it as early as possible and keep the momentum going.

Ask how long it will take to develop

Request a time estimate for the project, and to make estimating a little easier break it down into phases. We're all terrible estimators and if your project is more than six months, expect it to be off. This isn't the developers or your fault. It's just too hard to predict the future.

Break the project into smaller things. The smaller it is, the easier it is to estimate. It