

8 Day App Business Plan

The Treasure in Your Hand *Why Mobile Apps Can Make You Rich*

If used properly, the virtual world can be a potent tool to simplify and improve every aspect of your life.

As this virtual world evolves, apps are changing the way we communicate and relay information, allowing us to connect with people from Cape Town in South Africa to Montreal in Canada to Shanghai in China. Everywhere! The same way the internet opened up the international marketplace for thousands of businesses, mobile apps have done the same - but to the next level because **they are always in our pocket traveling with us wherever we go.**

We have apps that can control cars, allow us to play the piano, teach languages, and even recognize songs and name the artists. Apps are vital assets that streamline information, enhancing our lives while educating and entertaining us. But it's important to remember: with apps, it's all about convenience, convenience, convenience.

You have to understand this about the app business: At the core, it's about how consumers are using this technology to connect with the world. The more you are aware of this, the more money you can make with apps. You have to think like the customer if you're going to build your own app empire.

In this 8-day guide, you will gain essential insights into starting an app business, which will serve as the foundation of your success.

Day 1: Get a feel for the market.

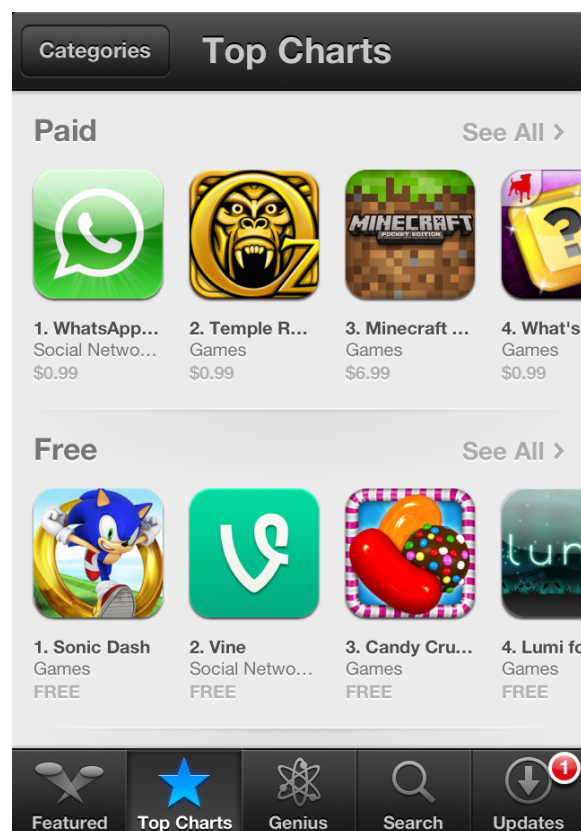
Note: this is the MOST important thing I can teach you. I have students that, after becoming full-time appreneurs and making thousands with their app business, still say this is the most important lesson they learned from me.

As with any business, your success will be directly related to your understanding of the marketplace. [The App Store](#) is the marketplace of the app business, so in order to understand the market, we have to study the App Store. This seems rather obvious, but you wouldn't believe how many developers I meet that don't understand this concept. They don't watch the market, follow the most successful apps, or try to figure out why those apps are successful.

In order to become a great app supplier, you must first become an app addict. That means spending this whole first day researching the market while downloading and playing with tons of apps (give yourself an app budget of \$100 to start).

Give yourself a full 8-hour period to go APP CRAZY. This training period is an investment in your expertise, which will become the lifeblood of your success. The more hours you rack up playing around and studying successful apps, the better you'll be able to understand their common traits and what users desire. Better yet, if you don't have a business partner, get friends and family involved to help with your research. This will also allow you to gauge different demographics and how they respond to particular apps.

So, how do you keep pace with the market? The best way is to study Apple's cheat sheet constantly. The App Store displays the top paid, top free, and top-grossing apps (the apps that make the most money, including free apps), almost in real-time. Apple provides the same lists in the individual app categories.



These charts are golden because they tell us volumes about the market. The best part is this information is freely accessible to anyone, at any moment (unlike the market info for basically every other industry).

Review these charts frequently, and keep a notebook of potential trends you spot. The more you review the charts, the more you'll start to notice little details, such as new apps that jumped the charts or emerging genres.

Example: when I first started my app business, I was really into yoga so I wanted to create a yoga app. I looked at the market and saw just a few yoga apps, but none that

had been really successful or hit the top charts (even in their category) so I didn't go through with it. Fast forward a couple years later and fitness apps are booming. If I had still wanted to develop this app, I would have continuously tracked all yoga apps during these years to watch for these apps rising the charts, THEN I would strike with my own app because I know the demand is there. People are actively LOOKING for these apps. If I had published these apps when I first thought of the idea, it would have sat on the store generating little to no revenue. My point: you want to see similar apps to your idea doing well or gaining traction because that means there are active users. However, the reason you do market research EVERY DAY is because you want to get in before there's an influx of the same apps. Luckily, this is easy to do when you're tracking and analyzing the market regularly because most developers aren't doing this – but more on this below!

Doing this repeatedly will also educate you on successful app design, marketing, various pricing models, and new app ideas. The research you're doing is simple, costs nothing, and it's actually fun!

Here are some questions to ask while you're researching successful apps in the market:

1. Why is this app successful?
2. What is its rank and has it been consistent?
3. Why do people want this app? (Look at the reviews, google blog posts.)
4. Has this app made the customer a raving fan?
5. Does this app provoke an impulse buy?
6. Does this app meet any of my needs?
7. Did I become a raving fan after trying it?
8. Will the customer use it again?
9. How are they marketing to their customers? (Check out the screen shots, icon design, and descriptions.)
10. What is the competitive advantage of this app?
11. What does this app cost? Are there in-app purchases? Advertisements?

Most developers will build an app and expect tons of people to find and download it right away. That rarely happens. You have to figure out what people are interested in and the kinds of apps they're downloading first, then you build your app based on that insight.

Do not build an app on an idea you THINK will be successful, build an app off an idea that can be backed by data and research.

Once you've put in the necessary research and feel you have a decent grasp on the market, it will be time to look back on the trends you discovered and explore some ideas for potential apps you can develop.

Note: if after Day 1, you feel like you aren't quite grasping the trends, don't rush this period! Spend the amount of time you need on this step as it's the foundation and most

IMPORTANT part of your business. I repeat, THE most important part of developing apps.

RESOURCES for Day 1:

- App Annie: <http://www.appannie.com/>
App Annie tracks your apps' metrics and has extensive app store data to help you make smart business decisions.
- SensorTower: <https://sensortower.com/>
Market Intelligence that apps can't live without.
- 148apps.biz: <http://148apps.biz/app-store-metrics>
Extensive app store metrics.
- Straply: <http://www.straply.com/>
App Store Optimization tools and research
- Google Keywords: <https://adwords.google.com/KeywordPlanner>
For ideas and trends research. This does not reflect what people are searching for on the App Store, but can give you insight into trends.

CONTINUED READING:

Games vs Utility Apps: Which is better? (<http://www.appempire.com/game-vs-utility-apps-which-is-better/>)

Video Tip: How to Pick Winning App Ideas (<http://www.appempire.com/video-tip-how-to-pick-winning-app-ideas/>)

Four Mistakes Entrepreneurs Make that Hurt Their App Business (<http://www.appempire.com/four-mistakes-entrepreneurs-make-that-hurt-their-app-business/>)

Day 2: Align your ideas with successful apps.

How do you know if the market wants your app? Again, you'll need to look at the [Top Apps chart](#). Are apps like the one you want to create listed there? If yes, you've got a potential winner. If not, keep looking. It's that simple.

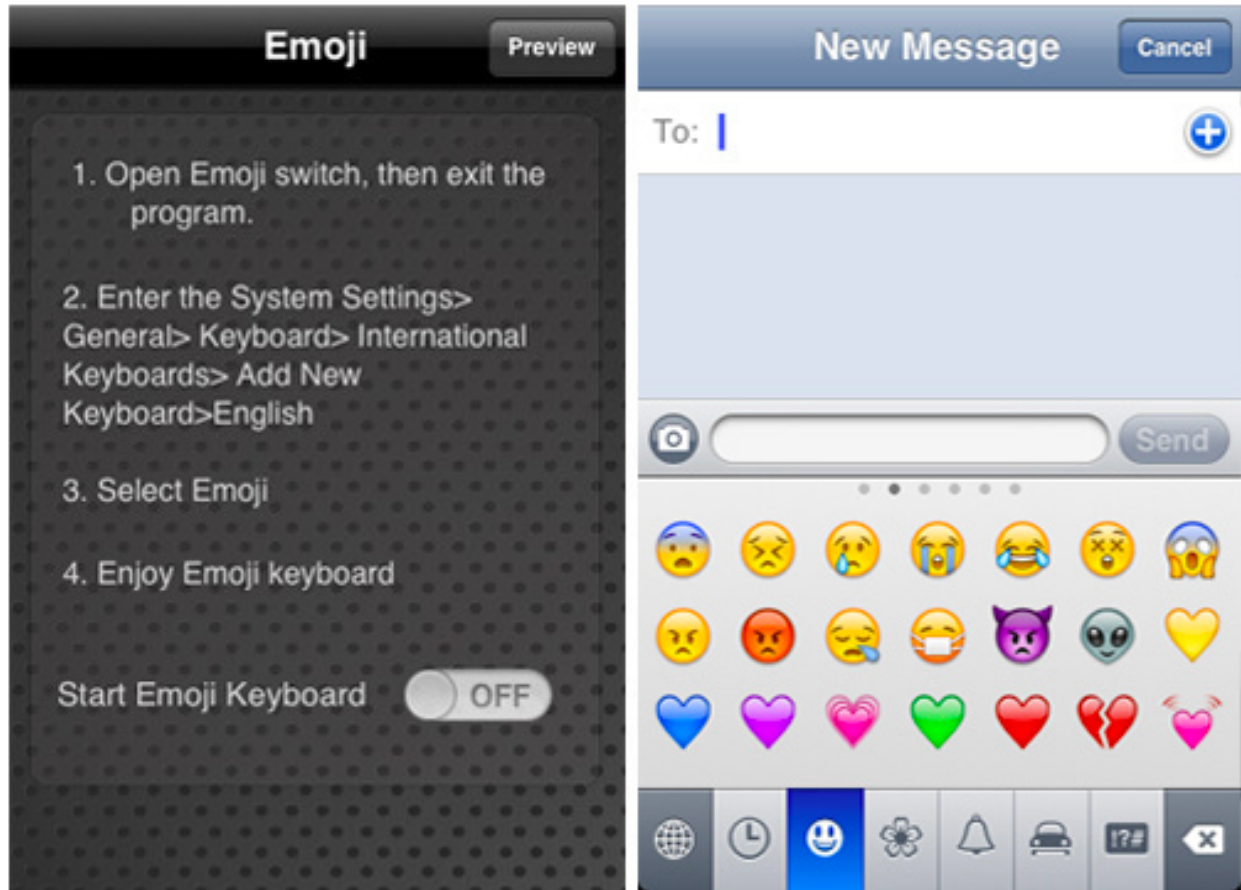
Don't hate; Emulate! When you follow in the footsteps of successful apps, you will have a better chance of succeeding because these apps have proven demand and an existing user base. This takes the guesswork out of creating great app ideas.

I can't stress the importance of emulating existing apps enough. It's easy for people to fall in love with their own idea, even if the market doesn't show an appetite for it. But this is one of the costliest errors you can make.

Unfortunately, developers make this mistake all the time. They focus on generating original ideas and spend a lot of time and effort creating those apps. When it doesn't work out, they go to the next untested idea, instead of learning from the market. Often times, they repeat this cycle until they run out of money and dismiss the app game. This doesn't have to be your experience.

Keep in mind that most ideas aren't original. Almost every app idea was inspired by an already existing concept. They emulated, and ADDED innovation. Angry Birds was not the first game of it's kind on the app store. But they created a user experience that was unlike any other. So don't mistake emulation for a lack of innovation. You need to take an existing, successful idea, and IMPROVE it to gain the market advantage.

A personal example of how to successfully emulate competitors is my Emoji app. First, I took a close look at what the market offered and downloaded all the major emoticon apps. I liked what I saw, but noticed that there was a lack of variety and limited functionality.



Screenshots from a competing Emoji app. The app (left) is opened once to provide the user with instructions on how to enable the Emoji keyboard (right).

I wondered how I could improve upon these existing apps, given that the Emoji keyboard had a limited number of emoticons that couldn't be increased. I was also curious how profitable these apps could be if they were only being used once.

I kept brainstorming until it hit me. I couldn't add more emoticons to the Emoji keyboard, but I could include unlimited emoticons within my app that people could send as images via text message or email.

I created an app that not only enabled the Emoji keyboard, but also contained an additional 450 emoticons within the app itself, which could be shared via SMS, e-mail, Facebook, and so on. The app was used constantly since users had to return to the app to send an emoticon.



Screenshots of my Emoji app.

The Emoji app was developed in two weeks. It followed the freemium model, meaning free with an in-app purchase option. The app hit the number one spot in the App Store's productivity category and the number 12 spot in the top free overall category within six days, raking in nearly \$500 per day. Bingo.

On Day 2, compile your top emulation ideas, and ask yourself these six questions:

1. Why are people purchasing this?
2. Can I do something to emulate this idea and take it to another level?
3. What other ideas would this app's demographic like?
4. How many other similar apps are in the market? (Visit TopAppCharts.com to find out.)
5. How successful and consistent have they been?
6. How does their marketing and pricing model work?

RESOURCES for Day 2:

- Same as Day 1 plus blogs and mobile news sites:

- <http://appleinsider.com>
- <http://www.insidemobileapps.com>
- <http://techcrunch.com/mobile>

- <http://techcrunch.com/tag/apple>
- <http://blog.flurry.com>
- <http://mashable.com/apps-and-software>
- <http://mobileorchard.com>

CONTINUED READING:

The Art of Improved Emulation: Brooke's SleepSmart App (<http://www.appempire.com/the-art-of-improved-emulation-brookes-sleepsmart-app/>)

Starting My Own App Business: How I Built My First App (<http://www.appempire.com/starting-my-own-empire-how-i-built-my-first-app/>)

App Math - Simple methods to know market numbers (<http://www.appempire.com/app-math-simple-methods-to-know-market-numbers/>)

Day 3: Design your app's experience.

You've studied the market, you see an opportunity, and you have an idea that could be profitable. Great! Now it's time to turn those thoughts into something tangible.

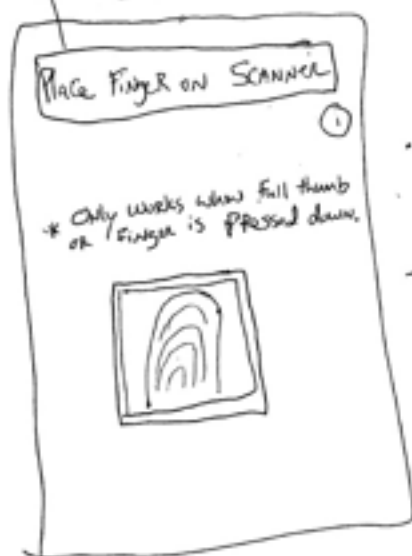
To convey your idea properly, you can simply draw it on a piece of paper. Maybe it will look like a 3-year old's artwork, but it will still convey what you're trying to do. Some people like putting this together in digital form, using Photoshop or [Balsamiq](#). Whatever you're most comfortable with, and whatever will give the programmers the details they need, is the way to go.

Another cool tool my students have used for those who want to draw on paper, but then move those drawings digital, you can check out: [Pop - prototyping](#).

For your viewing pleasure, here are the rudimentary drawings (a.k.a. wireframes) for my first app, Finger Print Security Pro. As you can see, it doesn't have to be pretty!

Hidden Button - hit to grant Access

* Default Access Denied



• Get Known Sounds



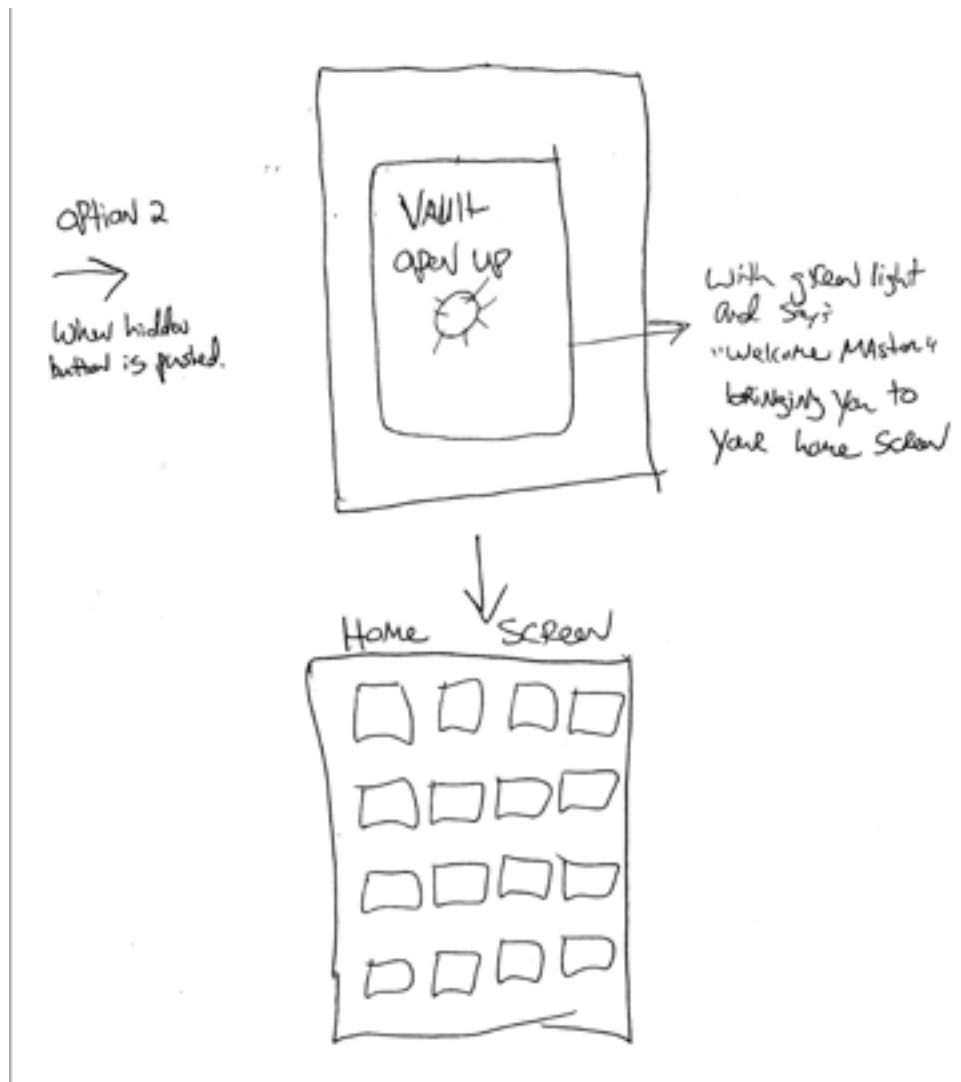
Print From 1st Screen

So prints go through a database w/ some thing to match

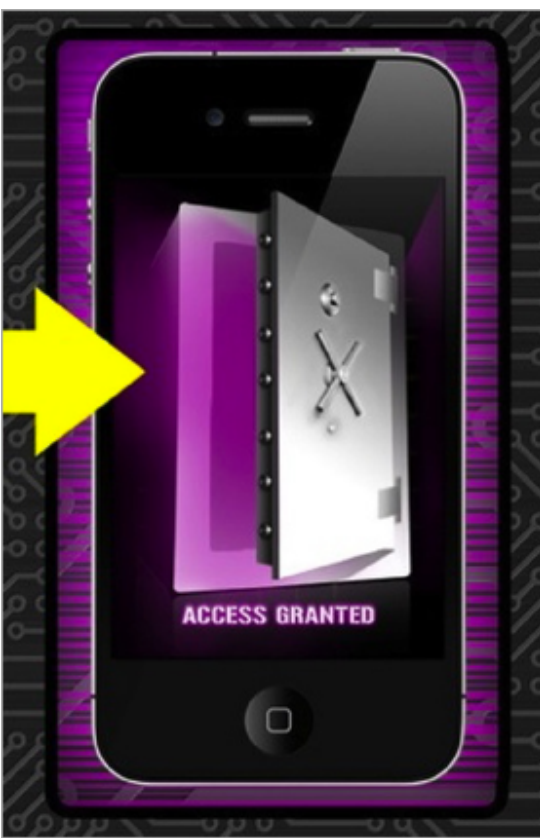
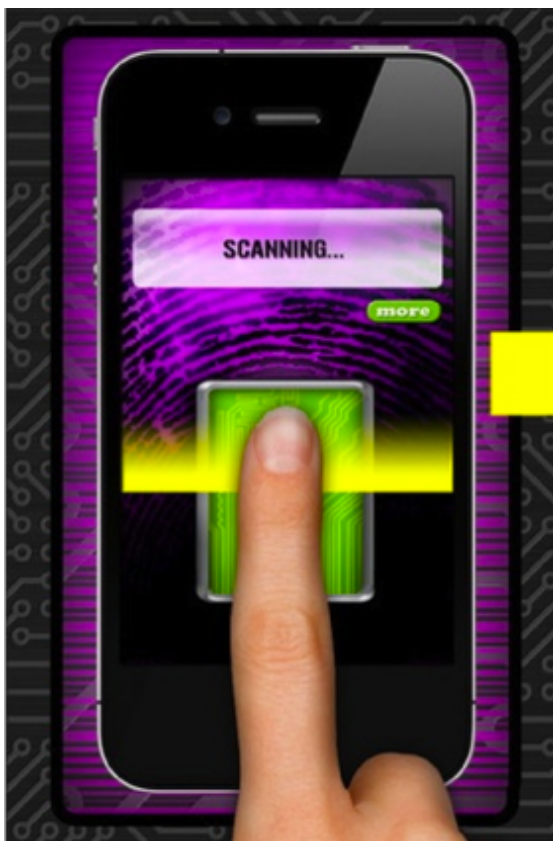
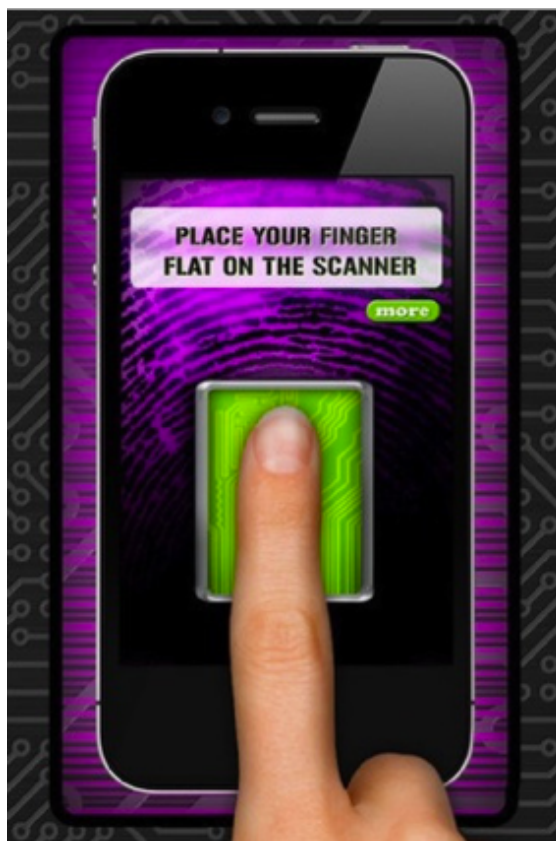


Default
Option 1

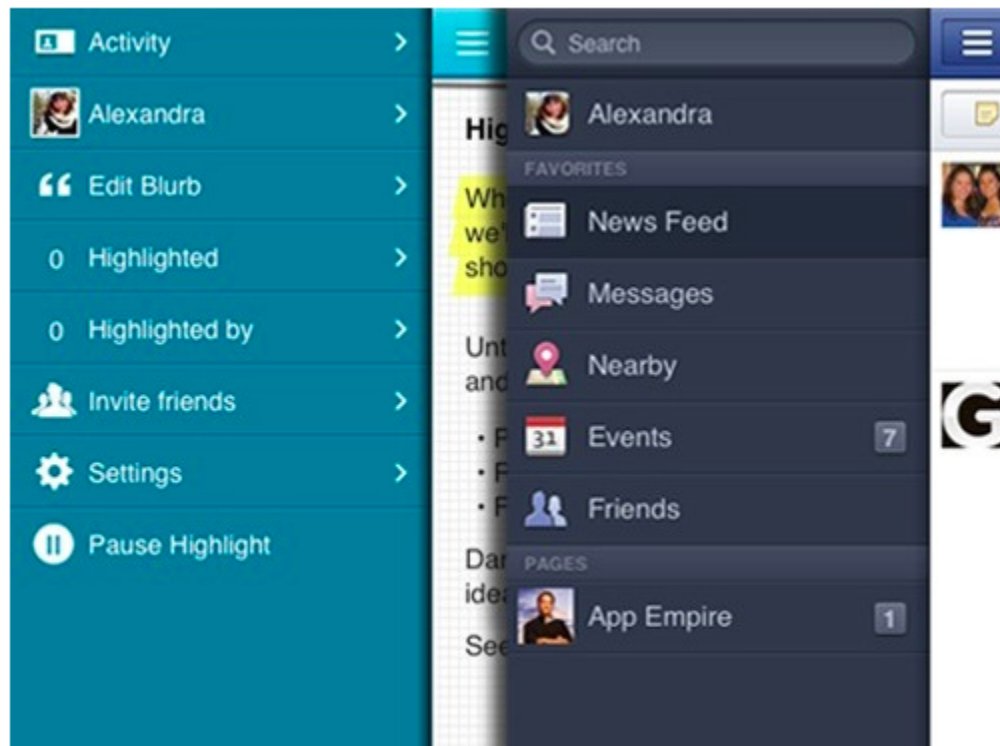




And here's how the app's final design turned out:



To make the design process easier, I look at certain apps in the App Store and reference them to show my programmers what I'm looking for. For example, I'll say, "Download the XYZ app. I want the ABC functionality to work like theirs. Take a look at the screenshots from this other app, and change this." I take certain components of apps that I'd like to emulate, and give them to the programmer so that we are as clear as possible.



Notice any similarities? Highlight's menu (left) emulated the style of Facebook's menu (right).

The clearer you are, the fewer misunderstandings and problems you will have once it's time to hand off your drawings to a programmer. The idea is to convey what the app will look like, where everything will be placed, and what happens if certain buttons are selected. This helps the programmer know what you want and will be a useful blueprint when designing your app. Do not be vague or ambiguous. Be extremely detailed, even if something seems obvious, things can get lost in translation. You should know what every part of your app will do. If you don't, you need to develop your idea more thoroughly.

You have to consider your design to be final before you can begin the coding phase. Inevitably, you will have ideas for additional features once you start testing the initial versions of your app. But if you decide to make major changes after a substantial amount of work has been done, it can frustrate your programmer. It's like telling the builder who just installed your fireplace that you want it on the other side of the living room. The news will not go over well. Most people don't realize this is what they are demanding of their programmer when they ask for big changes. That's why it's

important for you to take your time and carefully plan every aspect of the app before you submit it for coding.

Also keep in mind that new features can be included in future updates, and updates are crucial to sustainable success, so don't worry if you can't get everything in right away. Getting your app on the store is often more important than perfection because you can start collecting data, which will significantly improve your strategy.

A tip on good design practices:

If you look at the top charts, you will notice one thing: **QUALITY GRAPHICS**. This is extremely, extremely important for your app's interface ((user interface - or what the customer will see) *and* UX (user experience - how the customer interacts with your app)).

When designing your app's UI, I always tell people to **think like Apple**. What does this mean? It means creating sleek (i.e. modern), simple, intuitive designs.

What does intuitive mean? When a customer first opens your app, they shouldn't need a lot of explanation on how to use it. It should be intuitive in that they could start using it right away. Think simple and sleek (look at the top apps for reference).

Also, it's important not to clutter your app. Apps are meant to be simple and easy to use, it's different than a website, which means apps will need streamlined features. A good rule of practice is to write down all the features you want to have, and then focus on building out 3 of those features really well. You'll get more users by having an app that does 3 things extremely well, than a overcrowded app trying to do 10 things.

Remember: design like Apple. (If you look at the app's Apple features, you'll see the type of apps they look for.)

RESOURCES for Day 3:

- Balsamiq: <http://balsamiq.com/>

Balsamiq is a wireframing and mock up tool with a high focus on usability. Quickly come up with mock ups and easily share them with your developers.

- POP: <https://popapp.in/>

POP turns hand-drawn wireframes to interactive prototypes. Sketch the app on paper; take pictures and add hotspots to link them together; simulate on iPhone.

- Keynotopia: <http://keynotopia.com>

Keynotopia lets users design quick and easy interfaces and interactive mockups for web, mobile and desktop apps without touching a line of code.

- Skitch: <http://evernote.com/skitch/>

Get your point across with fewer words using annotation, shapes and sketches, so that your ideas become reality faster.

- Dropbox: <https://www.dropbox.com/>

Organize your ideas, save your mockups, store your files.

- Jing: www.techsmith.com/jing.html

Share images and short videos of your computer screen.

- AppDesignVault: <http://www.appdesignvault.com/>

Buy templates or get ideas.

CONTINUED READING:

7 Keys to Unlock an Apple Design Award (<http://uxmag.com/articles/7-keys-to-unlock-an-apple-design-award>)

3 Tips to Be a Hot Shot UI/UX Expert (www.appempire.com/3-tips-to-be-a-great-ui-ux-expert/)

Day 4 - 6: Hire a great iOS programmer.

Coding your own app, especially if you're teaching yourself at the same time, will take too long. The likelihood of you getting stuck and giving up is very high. It will also be unsustainable over the long run when you want to create several apps at the same time and consistently update your existing apps. After all, the goal is to get your time back and escape the long hours of the rat race. Therefore, programmers will be the foundation of your business. They will allow you to create apps quickly and scale your efforts.

Hiring your first programmer can be a lengthy process as if you're not feeling comfortable with your options, you shouldn't just choose whoever. This will be your right-hand man and the one who makes it all come to life. You need to be able to trust your programmer. On Days 4 - 6, you'll need to:

- post the job
- filter applicants
- interview qualified candidates (on Skype > face-to-face, no exceptions!)
- have them sign your NDA, explain your idea

- give them a micro-test

... all before coding begins! If this process takes more time than expected, it is time well spent. However, don't give yourself excuses. Make sure if you are taking more time, that you're using it wisely and not procrastinating out of fear or uncertainty of success.

Some people find this step intimidating, but it's an extremely valuable learning experience. Making great hires will help you avoid unnecessary delays, costs, and frustration in the future. You'll always be looking to add new talent to your team, so learning how to quickly and effectively assess programmers is an important skill to develop.

Let's get started. The first part of this step is to post your job to a hiring site.

Top Hiring Resources

These websites allow programmers to bid on jobs that you post. As you can imagine, the competition creates a bidding frenzy that gives you a good chance of getting quality work at a low price.

Here are a few of my favorite outsourcing sites:

[oDesk](#)— Its work diary feature tracks the hours your programmer is working for you and takes screenshots of the programmer's desktop at certain time intervals.

[Freelancer](#)— This site has the most programmers listed. They claim that twice as many programmers will respond to your ad, and I found this to be mostly true.

[Guru](#) and [Elance](#). Both of these sites have huge lists of programmers.

iOS or Android developer forums — Use these forums to create relationships with other appreneurs. Often times, they'll suggest great teams or programmers they recommend working with. You can also ask for advice on hiring your programmer from those who are going through it with you.

Below is a template of a job posting, followed by an explanation for each of its components:

* **Job Title:**

* **Description:**

Need a custom mobile app made (quickly!)

I need someone who is very experience in coding for iOS to create a feature rich, nice looking, mobile app.

Description of App Here: (Keep basic until NDA signed)

This needs to be a job than can be completed quickly.

Requirements:

- Someone who can not only program but can also do the UI design
- Someone who has at least 300 hours on Odesk
- Someone who has a 5 star rating
- Someone who takes pride in their work and makes polished looking products.
- Someone who speaks English well

Here's what you need to do next:

- Please reply with examples of your previous work including any user interface or design examples you have done before. If I like your previous work, I will reply and give you more details.

*** IMPORTANT when you reply start by typing the code words: Honey Badger

This way I know you took the time to read the complete description instead of just replying to all new submission like many people do!

Thanks,
Chad

Objective C, C Programming, Cocoa, iPhone, Corona

Skills Required: Add skills to significantly improve the quality of contractors that apply.

Job Type: ☐ **Hourly** - Pay by the hour. Verify with the [Work Diary](#).
☒ **Fixed Price** - Pay by the project. Requires detailed specifications.

* **Budget:** \$ US Dollars

<http://www.flickr.com/photos/appempire/7052387709/sizes//in/photostream/>

Enter the skill requirements—What programming languages do they know? For iPhone apps, the skills I list are: iPhone, Objective C, Cocoa, and C Programming.

Some advanced game apps require Unity 3D knowledge, this is quite pricey, so if it's your first app, I strongly suggest not going this route just yet.

Give a basic description of your project—Keep it simple and skill-specific. Tell the applicants that you will discuss details during the selection process. Do NOT reveal the

specifics of your idea or marketing plan. Use general descriptions, and request info on how many revisions (a.k.a. iterations) their quote includes.

Post your ad only for a two days—This way programmers have a sense of urgency to quickly bid on your job.

Filter applicants—I always filter applicants using these criteria:

- They have a rating of four or five stars.
- They have at least 100 hours of work logged.
- Their English is good.

Compose individual messages to all suitable applicants, inviting them to a Skype call for further screening. Most of these programmers will be overseas, which can present issues with communication and time zone differences. Therefore, a Skype interview is an absolute must before you can continue. Disqualify anyone who is not willing to jump on a Skype call.

Note: There is a difference between programmers and graphic designers. Sometimes one person can do both, other times it's best to hire a graphics programmer to work with your programmer. Graphics are SO important for apps, that you want quality. If your programmer has a recommendation of someone they've worked with before - even better! They always know how to work well together, but you still to interview that person as well.

The Interview: Essential Questions to Ask Programmers

Don't give away any of your specific ideas during this initial conversation. Just talk about general genres, like "a camera app," or "messaging app," etc. Whenever the topic comes up, say you'll be more than happy to discuss everything after they sign the NDA (**if you want a copy of the NDA template I use, see the bottom of this post**). Here are the questions you should ask each applicant before committing to anything:

- How long have you been developing apps?
- How many apps have you worked on? Can I see them?
- Do you have a website? What is it?
- Do you have references I can talk to? **THIS IS IMPORTANT.** Many developers will list app's they've worked on...but actually haven't. You have to followup with who they say they've worked for in order to **ENSURE** they've actually developed these apps.
- What's your schedule like? How soon can you start?

- What time zone do you work in? What are your hours?
- What's frustrating for you when working with clients?
- Are you working with a team? What are their skills?
- Can you create graphics, or do you have somebody who can?
- Can I see examples of the graphics work?
- What happens if you become sick during a project?
- What if you hit a technical hurdle during the project? Do you have other team members or a network of programmers who can help you?
- How do you ensure that you don't compete with your clients?
- Can you provide flat-fee quotes?
- What's your payment schedule? How do you prefer payment?
- Can you create milestones tied to payments?
- Do you publish your own apps on the App Store?
- How do you submit an app to the App Store? (Can they verbally walk you through the process, or do they make you feel brain challenged?)

Finally, mention that you like to start things off with a few simple tests (creating/delivering your app's icon and a "Hello, World!" app) before coding begins. You need to tell them this upfront so they aren't surprised after they have provided their quote. Most programmers are happy to get these tests done without a charge, but some will want a small fee. In either

During the interview, pay attention to how well they are able to explain themselves. Are they articulate? Do they use too much techno babble? Do they speak your native language fluently? Do they seem confident with their answers? How is their tone and demeanor? If you have any issues or worries, you may want to move on to somebody else. But if you can communicate with them easily and your gut is telling you "Yes," you'll want to proceed to the next step.

In either case, thank them for their time and mention that you will follow up with an NDA agreement if you decide to move forward.

Be prepared to spend a little extra time on developing your new system and

relationship. It's when appreneurs try to go *too fast* that they get in **trouble** or overlook something that was a **red flag**. Here are 8 steps to ensure you're getting the most out of your new developer:

1. Never pay money upfront.

It's just too risky. Don't pay for what you don't have.

2. Create milestones and only pay on deliverables.

Prior to hiring, have your milestones outlined and agreed upon by your developer in writing. Make your expectations clear about what you expect to receive for each milestone.

3. Ask if they offer a cheaper rate.

Many developers and designers will offer a discount rate on your first project with them. Then once you rehire them for your next project, it goes back up to their regular rate.

4. Pay with flat rates.

It reduces risk and helps your milestones be met in a timely manner. They have no reason to delay or inflate time spent on a feature if they're not getting paid for those hours.

5. Start small.

Don't make the first milestone half-way through your app's completion. Make the first assignment a small project. Did they communicate well? Were you happy with their work? Did anything feel off? If you have a bad feeling about someone, move on to someone else *right then*. It rarely works out or gets better.

6. Daily communication.

When first starting with someone new, you should be receiving a daily report and talking every day, even if it's just to say, "everything is going smoothly." Also, keep in mind that prior to hiring you should outline what you expect in terms of communication so they know what you expect daily.

7. Be so clear and detailed, it feels tedious and painful.

You've never worked together. Think of development like those "team building" exercises where one person is blindfolded while the other person tells them what to draw. You have to be *super* detailed about extremely obvious things because you *never know* what could be misconstrued, no matter how painfully obvious it is.

8. Offer incentives or bonuses.

I like to offer bonuses to hit a deadline faster or when they overall did an incredible job. I think it builds rapport when you show them you appreciate and reward great work. They'll be more incentivized to continue working for you.

Most of these rules should be followed even if it's *not* a new developer, but they become the most critical when starting a new relationship.

It *can* be difficult to find a great developer or designer, but they're out there! Sometimes it just takes a little longer to find and takes more interviews than you hoped. The point is to **not give up** when there's a road bump. *That's* what makes an entrepreneur successful. They understand that when things don't go their way, it's not a failure - it's a learning experience.

More RESOURCES for day 4-6:

- Evernote: <http://evernote.com>

Remember and act upon ideas, projects and experiences across all the computers, phones and tablets you use.

- Skype: www.skype.com/

Make internet calls for free and cheap online calls to phones and mobiles with Skype.

- 99Designs: <http://99designs.com/mobile-app-design>

Designers compete in your mobile app design contest. They submit designs, and you give them feedback.

CONTINUED READING:

Video Tip: Your Time is Value! Start Outsourcing Now (<http://www.appempire.com/video-tip-your-time-is-value-start-outsourcing-now/>)

Finding Developers: A World of Options (<http://www.appempire.com/finding-developers-a-world-of-options/>)

Day 7: Sign NDA & establish milestones

You must protect your ideas, source code, and any other intellectual property. These are the assets that will build your business, so you need to have each potential programmer sign an NDA before you hire them. Yes, it's rare to have an idea stolen, but it does happen.

As you're going through this process, you will be getting feedback on your programmers' responsiveness. For instance, if it's taking too long for them to sign the NDA, it might indicate how slowly the development process will move. Buyer beware!

Once the NDA has been signed by both parties, you can share your idea and designs with your programmer. At this stage, it's critical to ensure they have the skills to complete your app. You do not have any wiggle room here, especially on your first app. Either they know how to make it or they don't. You want to hear things like, "I know exactly how to do that" or "I've done similar apps, so it will not be a problem." You don't want to hear things like, "I should be able to do that, but I have to research a few things"

or “I’m not sure but I can probably figure it out.” If you hear those words, switch to an app idea they are confident about or run for the hills.

After you’ve found the best programmer for the job, you can commit to hiring them. On day 7, establish milestones and timelines during the quoting process (break up the app into several parts), and decide on a schedule and processes for check-ins that you’re both comfortable with (ask them directly how they like to be managed). Day 7 is when you’ll be setting up the operations on how your project will run, so be thorough.

Example milestones:

Part one:

Test - icon and or Hello World App

Basically, you want to measure their capabilities in some way. This can be anything from developing a basic icon, sending a simplified demo of one of your app's features, or a "Hello Word" app build.

0-25% paid upfront

Part two:

25-50% this is paid once 1/2 of the app is completed and approved by you

Part three:

50-75% this is paid once 3/4 of the app is completed and approved by you

Part four:

75-100% this is paid once the app is completed, approved and the final build and deliverables are delivered.

TIPS:

- Make sure to have a certain number of iterations (i.e. changes after project is over) approved upfront- IE: It could be 3-5 changes, icon changes, etc. Be clear with these to prevent confusion down the road.
- Do flat fees as much as possible. This way, if the developing team (or developers) does not bid the project properly you won't get hurt for it.
- After each milestone your developer must send you the code so you have your work and are constantly updated or so your work is safe. Regardless of bugs, you should be receiving code as you go along. If they will not provide you with this, do NOT make that milestone's payment. Let them know this will be a requirement ahead of time.

You will need to periodically review their work, from start to finish. I strongly suggest having a check-in session every day, or every other day, to ensure things are going as planned. Most applications go through multiple iterations during design and

development, and I won't release partial payments until I'm fully satisfied with each milestone.

RESOURCE for Day 7:

- Basecamp: <https://basecamp.com/>
The #1 project management application.
- Teamwork: <https://www.teamwork.com/>
Online project management software. Easily manage your projects and team online.
- Hackpad: <https://hackpad.com/>
Create and Organize. Take collaborative notes, share data and files, and use comments to share your thoughts in real-time or asynchronously.

CONTINUED READING:

Are You the Problem in Your Own Business? Don't Be Your Own Worst Hire (<http://www.appempire.com/are-you-the-problem-in-your-app-business/>)

Finding Developers and Speaking Their Language (<http://www.appempire.com/finding-developers-and-speaking-their-language/>)

Day 8: Begin coding.

Rather than jumping haphazardly into a full-fledged project, I prefer to gradually ramp up my programmer's workload by starting with a couple smaller tasks. You need to assess their graphics capabilities, implementation speed, and overall work dynamic (e.g. communication, time zone, etc.). If you're underwhelmed with their skills early on, you need to get out quickly. Remember: Hire slow, fire fast. It will pay off over the long run.

Here's my three-step process during the coding phase:

1. **Icon** —Ask the programmer or designer to create and deliver the icon of your app. You will probably have several ideas for icons, so pass them on and ask for a finished iTunes Artwork version of the icon. Then once you've decided on an icon, design the icon sizes around Apple's Guidelines for each product. Icon app sizes can be found here: [Icon Size Chart](#)
2. **Hello, World!**—Ask the programmer for a "Hello, World!" app. It's a simple app that opens up and shows a page that displays "Hello, World!", and it will take them 10 minutes to create. The idea here is not to test their programming skills, but to determine how they will deliver apps to you for testing. This app should include the icon they created, so you can see how it will look on your phone.

3. App Delivery—When the programmers are ready to show you a test version of your app, they have to create something called an “ad hoc” (a version of your app that can be delivered to and run on your iPhone, without the use of the App Store). This ad hoc version of your app needs to be installed on your phone before you can test it. The initial installation was a bit cumbersome in the past, but a new service called [TestFlight](#) has simplified the process. I ask all programmers to use this service even if they have not used it before. They will be able to figure it out, and you’ll be able to install your test apps with a few touches on your phone.

Now that you’ve begun coding, it’s important to look for ways to increase productivity, lower costs, and systematize your business.

Here are 3 ways I systematize my business from the beginning:

1. Templates/POS/Automation

What in your business can be automated? What can be done by other people? Really think about that. It's easy to find yourself doing something because you're the fastest and best at it, but if you can create templates or POSs to send to others, or even better automated actions, while you work on the *really important* things, then you're not only optimizing your business - you're preparing your business for growth. Some examples:

- Receiving a lot of email? Most emails are about the same types of topics. Create an autoresponder or template.
- Are you uploading all your own apps to iTunes? Turn this into a template or video and give to a PA or TA. Even important systems like, how to choose good keywords, can be handed off to someone else to get your app in the store faster, then just makemupdates for testing.
- Building a new app? Reuse successful screenshot templates, icons, and copy. You already know it converts well - just adjust it to fit the current app and submit. Your production times will drastically increase.

2. Buy stock.

There are tons of high-quality resources *already made* that you can use for your apps instead of building from scratch. You can use [stock photos/graphics](#), [stock music](#) and sounds, and most importantly you can [buy and reuse code](#) to *significantly* lower production costs. Whether you're reskinning an app or building a custom app, you can use these stock options to save costs and work faster.

3. Outsource.

There are a lot of talented people on freelance websites that you can use on a per project basis to get things done quickly. And for really cheap too! Half the battle is with *yourself* and realizing that you don't have to control every little detail. Empower your team or your hires to make decisions themselves. Once you realize that imperfections are OK and don't always affect the bottom line, you'll find yourself producing things *much* faster and more efficiently.

Give yourself a pat on the back — you've made serious progress! But don't get too caught up with yourself, because a big mistake many appreneurs make is publishing their app to the store and thinking they're done. Once your app is on the store, now it's GO TIME. This is when you'll use ninja marketing and monetization strategies to generate revenue and bring in customers from all over the world.

MORE RESOURCES for Day 8:

- Testico: <http://www.testico.net>

You can to put the icon you've created on your iPhone or iPad home screen, and generate automatic previews of standard icon sizes on the standard iPhone screen.

Stock Websites:

- <http://www.pond5.com> - Video, sound effects, music, after effects, photos and illustrations

- <http://audiojungle.net> - Audio

- <http://graphicriver.net> - Graphics

- www.istockphoto.com/ - Graphics

CONTINUED READING:

Carter Thomas: The secrets of the App Store (<http://www.appempire.com/carter-thomas-chad-mureta-the-secrets-of-the-app-store/>)

Optimizing App and When to Move On (<http://www.appempire.com/optimizing-apps-and-when-to-move-on-interview/>)

How can I improve my app? (<http://www.appempire.com/how-can-i-improve-my-app/>)

How to Master Anything w/ Tim Ferriss (<http://www.appempire.com/how-to-master-anything-interview-w-tim-ferriss/>)