



Designing the iPhone

User Experience

A User-Centered Approach
to Sketching and Prototyping
iPhone Apps

Suzanne Ginsburg

Case Study Contents

Case Study 1 • Windspire	CS1-1
Case Study 2 • Aardvark Mobile	CS2-1
Case Study 3 • Foodspotting	CS3-1
Case Study 4 • Not For Tourists	CS4-1
Case Study 5 • MUSE	CS5-1
Case Study 6 • Prototyping at Dan4, Inc.	CS6-1
Case Study 7 • What's Shakin'	CS7-1
Case Study 8 • REALTOR.com	CS8-1
Case Study 9 • Sonos	CS9-1
Case Study 10 • FlightTrack	CS10-1
Case Study 11 • USA TODAY	CS11-1
Case Study 12 • Voices	CS12-1
Case Study 13 • Convertbot	CS13-1



CASE STUDY 1

Windspire

CREATE WITH CONTEXT is a Silicon Valley–based consultancy doing research, innovation, and design for digital products. The firm’s unique approach connects people, context, and technology to help clients create successful digital products. Create with Context offers a broad set of services to organizations around the world, including making existing products easier to use, defining next-generation products, and entering new markets. Create with Context provides services across web, mobile, consumer electronics, and desktop environments to emerging technology start-ups, government agencies, and Fortune 500 companies.

Can you tell me more about the Windspire products?

When most people think of wind turbines, they imagine the large ones found along rural highways. But the Windspire is completely different—it’s small enough to fit in your backyard (30 feet tall but quite narrow) and makes little noise. More important, you don’t need a lot of wind for the Windspire to work. The 1.2-kilowatt turbines can operate in areas with a minimum average speed of ten miles per hour.

How did you approach the project?

Mariah Power, the company that created Windspire, wanted to create more awareness around the Windspire and the potential for people to obtain them. Customers were often uncertain about whether they could benefit from a turbine and would ask, “Do I have enough wind to power a Windspire?” To better understand their customers’ needs, we recommended ethnographic-style interviews in three regions: the San Francisco Bay Area, rural Kansas, and Washington, DC.

What did you learn from the research?

Through these ethnographic-style interviews we discovered two distinct user groups: the Green tribe and the Easy Eco tribe [FIGURE CS1.1]. The Green tribe wants to leave the world in a better place—they monitor their electric bills, drive hybrid cars, reuse plastic bags. In contrast, the Easy Eco tribe makes environmentally friendly decisions because they want to fit in—they would be embarrassed to forgo recycling or drive an SUV.

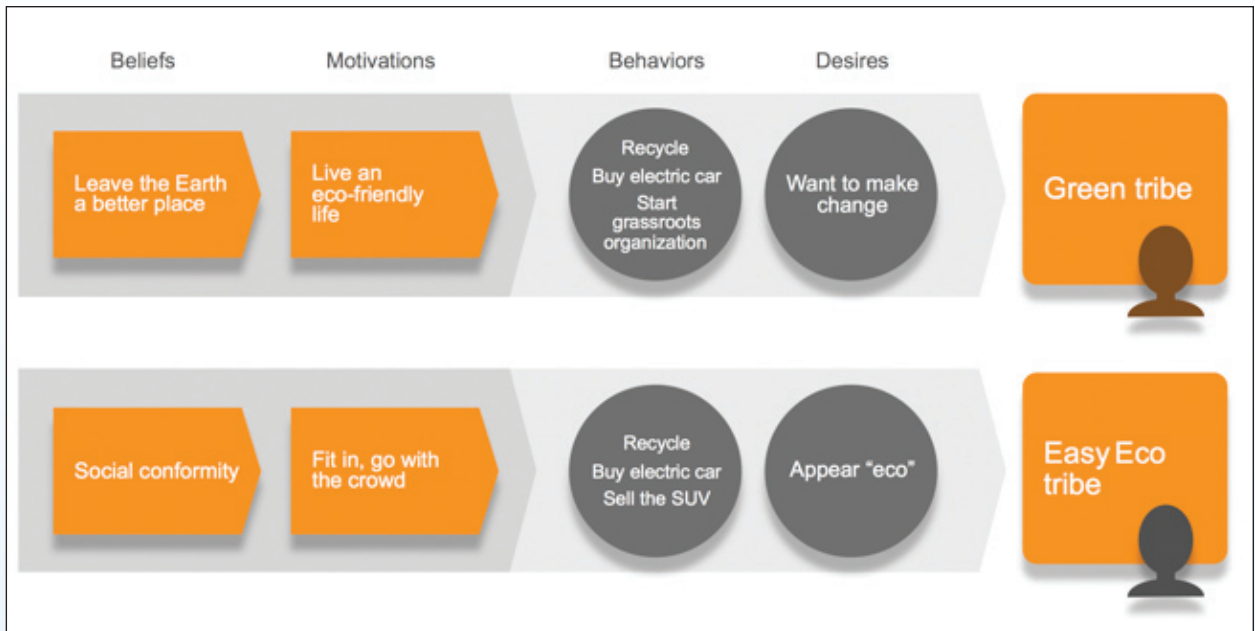


FIGURE CS1.1 Windspire user groups: Green tribe and Easy Eco tribe

What happened after the user research?

We started brainstorming solutions for these two user groups. Helping users determine the cost of the turbines was easy to accomplish on the web site, but measuring the wind speed was impossible. That’s when we came up with the app idea. Have you seen the Zippo app? You can blow on it to put out the “flame.” If the Zippo app can detect the user’s breath, we thought, “Maybe we can create an iPhone app to measure wind.”

What was the outcome of your technical investigation?

We discovered that it would be possible to use the phone’s microphone to capture wind noise. After filtering out ambient sound, an algorithm could convert the wind noise into a decibel rating that corresponds to wind speed. Once we knew this was possible, we started sketching our ideas and building prototypes.

Were you able to test your early designs with prospective users?

Absolutely—we conducted two rounds of user testing. The first was with a paper prototype; the second was with static images on the iPhone [see [FIGURES CS1.2–CS1.3](#)]. One of the main findings from the first round of testing was that the copy didn’t speak to the Easy Eco tribe—we didn’t have any concrete examples of the benefits. Another significant problem was related to how participants were holding the iPhone—many were covering the microphone and pointing it in the wrong direction.

How did these findings impact the design?

To address the copy issue, we refined the messaging to show how the wind turbine could help people save money and protect the environment. For example, one of the benefit statements says, “Installing Windspire will help

reduce your home’s carbon emissions.” In the case of the app’s microphone and orientation, we included an illustration with text showing how and where to point the app.

What happened when you tested the app the second time?

Our efforts at fine-tuning the messaging paid off—in the second round of testing, the participants had a better understanding of the wind turbine benefits. Moreover, the new infographic solved the microphone and orientation problems [see FIGURES CS1.4–CS1.5].

How have users responded to the app?

Feedback has been very positive, and we’ve been excited to see that it has generated a lot of buzz in the media around wind energy. People all around the planet have measured the wind in their backyards, at local parks, and at their workplaces and have left their marks on the map for others to see. And the app has been mentioned in the media, including the *New York Times*, *Fast Company*, CNET News, and Reuters. ■

(Windspire icon and application screenshots courtesy of Create with Context)

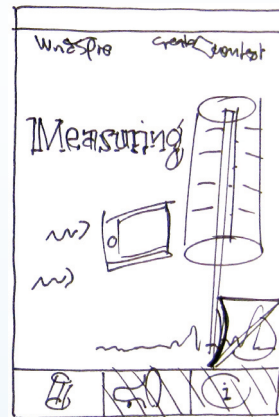


FIGURE CS1.2 Early sketch of the Windspire iPhone app

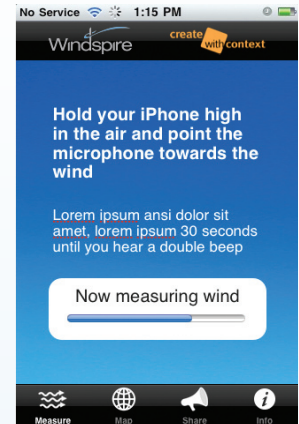


FIGURE CS1.3 Instructions before the usability study

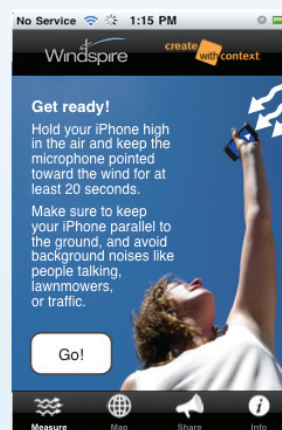


FIGURE CS1.4 Final Windspire app design (capturing wind)

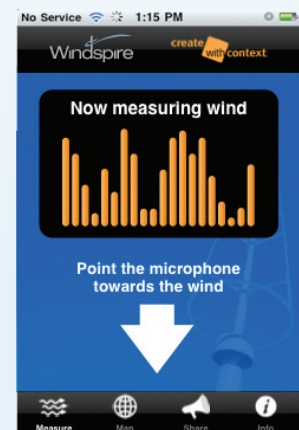
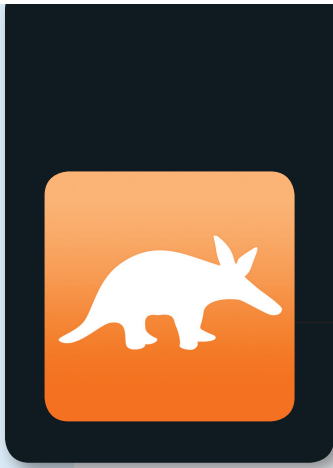


FIGURE CS1.5 Final Windspire app design (measuring wind)



CASE STUDY 2

Aardvark Mobile

AARDVARK is a new kind of tool that lets you tap into the knowledge and experience of friends and friends of friends. Send Aardvark a question (from iPhone, <http://vark.com>, IM, email, or Twitter) and Aardvark will discover the perfect person to answer in minutes. Rob Spiro is the cofounder of Aardvark, where he leads user research and product design.



FIGURE CS2.1 The Answer screen shows questions users can answer on Aardvark.

What inspired Aardvark to build an iPhone app?

We launched Aardvark over Instant Messenger in March 2009, and almost immediately users were asking for a way to use Aardvark on the go. That made sense—Aardvark is a great tool for situations when you’re not sitting in front of a computer. Aardvark can find quick, helpful answers from your network, which is extremely useful for local recommendations (Can anyone recommend a coffee shop near the train station?), suggestions for activities (What’s a great way to spend a few hours before dinner?), second opinions (Should I really buy this GPS device?), and lots more.

Once we decided to adapt Aardvark for these on-the-go situations, it was a quick jump to building an iPhone app. Not only is the iPhone the largest mobile platform for new software, it’s also a social device that you use to connect with your friends—and Aardvark is all about connecting you to friends (and friends of friends) in the moment. [Screens from the app are shown in **FIGURES CS2.1–CS2.3**.]

How did you approach the project?

We’re extremely design-driven and user-driven; that is, we always start with product design before building any technology, and we always start our product design with user studies. Being “user-driven” in our product development is more than an attitude—it’s a rigorous and ongoing process.

The first step in our process was to validate the concept. We had a lot of users requesting an iPhone app, but would they really download and use it? How could we conceive of the Aardvark iPhone app so that it would fill a recognized user need but wouldn’t compete head-on with formidable and entrenched players? To answer these questions we ran a series of interviews with both existing Aardvark

users and nonusers whom we recruited via Craigslist. We had people walk us through their typical iPhone app usage, watched them browse through the App Store (and react to the various offerings), and showed them a number of splash screens for a potential Aardvark app.

What was the outcome of the user studies?

Ultimately we were convinced that there was, in fact, a manifestation of the concept that would make a successful app. People were responding positively to a subset of the splash screens, and those reactions made sense given the usage patterns that we observed.

We then started designing the basic user experience. Given the guidance we already had from our initial interview series, our design team came up with a series of mock-ups of app descriptions, splash screens, and core app screens. We were able to get the mock-ups directly onto the phone by using a free tool called LiveView. We then brought in another group of five to ten people to react to these mock-ups, using a usability-testing method called talk-aloud protocol, which encourages people to verbalize the actions they're taking.

At this point (and not before) we started writing code to build a rough first draft of the app. The first draft had very limited functionality—our goal was to release something quickly that would enable the core functionality (asking a question and getting an answer via push notification), without any of the secondary features that would come later. We released this app to a group of ten users, all employees of Aardvark, using the iPhone's Ad Hoc Distribution method.

We then began a process of collecting feedback, running usability tests with the early users, refining the app, and releasing a new version via Ad Hoc Distribution. Each time we released a new version we increased the size of the test group, ultimately reaching the 100-user limit. We made sure to have in-depth email exchanges and phone calls with each and every tester to delve into their feedback. With that much qualitative feedback we were able to

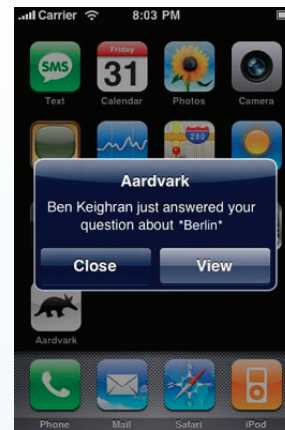


FIGURE CS2.2 Aardvark sends a push notification when users' questions are answered.

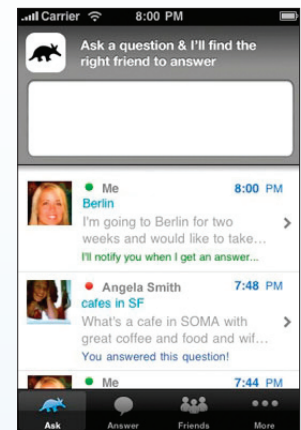


FIGURE CS2.3 The Ask screen shows the user's previous questions and answers on Aardvark.

notice patterns that signified problems or opportunities, then react quickly. We also quantitatively tracked usage, using software from Flurry to track funnels.

How did the research impact your initial designs?

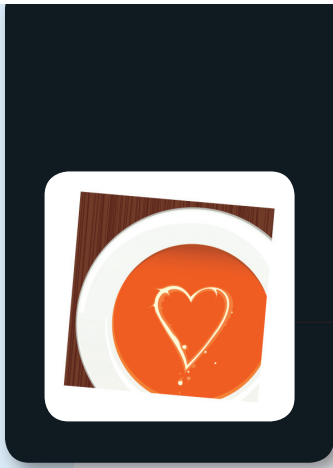
At each stage our user research fundamentally altered the app. Based on our first interview series that helped us validate the initial concept, we knew that the core functionality would have to be asking a question and reading an answer. Our research taught us that the asking screen and the history screen should be combined (initially they were separate), that we needed an answering experience to allow for a more passive engagement (initially there was no answering experience), that people wanted to see a list of their friends using the app, and on and on, even to the position of the buttons and the shape of the icons.

How have your users responded to the app?

The response has been extremely positive. Every day people are getting helpful answers from their network within minutes, helping them make decisions and find information in all sorts of circumstances. Major publications like the *New York Times*, *Macworld*, and *TechCrunch*

have given us glowing reviews, and download numbers have been amazing.

We built a feedback form directly into the app, and since launch we've received thousands of pieces of feedback. We've also benefited immensely from tracking usage patterns via Flurry and finding opportunities to improve. We've since launched two updates to the app based on this phenomenal feedback: adding new features, tweaking existing interfaces, and fixing bugs that weren't surfaced by our initial test group. We are hard at work on our next update now, continuously iterating on the same design and development process and continuously improving. ■



CASE STUDY 3

Foodspotting

FOODSPOTTING is a visual local guide that lets you find dishes instead of just restaurants. This web and mobile app is powered by Foodspotters, who earn points and recognition for sharing foods they've spotted while enabling Foodseekers to find whatever they're craving and see what's good at any restaurant. Foodspotting was founded by Alexa Andrzejewski, a user experience designer formerly of Adaptive Path, and Ted Grubb, a Rails engineer behind Get Satisfaction.

What inspired you to start Foodspotting?

When I visited Japan and Korea a year ago, I discovered dishes I'd never heard of before, from okonomiyaki to tteokbokki. Upon returning to San Francisco, I mined sites like Yelp and Chowhound in search of these dishes, only to find the local guides too broad and the discussion boards too unstructured. As I shared my frustration, I quickly found I was not alone.

While everyone I talked to saw the need for a dish-centric dining guide, everyone also had the same question: "How will you get that kind of data?" But as an interaction designer with a particular interest in what motivates people, I looked around and saw people taking pictures of their food all the time and showing them off on blogs and Twitter, and I had a hunch that harnessing existing behaviors could be the key to building a rich collection of dishes and where to find them.

How did you approach the project?

After the ideas had percolated in my head and in notebooks for a few weeks, I knew I needed to both vet the idea with potential users and attract cofounders and teammates to make it real. To make the ideas tangible, I created a concept poster—a tool that I'd developed for a previous mobile project at Adaptive Path. [See the concept poster in Chapter 6, "Exploring App Concepts."]

Concept Posters are a powerful way to articulate what the experience of using something could be like without getting into the design details. Using simple star people and sketches of metaphors to illustrate how Foodspotting is like "a personal food passport" and "a picture menu for any restaurant," I was able to get others excited about the vision and gather feedback that would inform future design decisions and prioritization.

By showing this poster to potential users, I was able to see which aspects of the Foodspotting experience were most compelling to people (the picture menu for every restaurant) and which were interesting but not essential (food flavor profiling and recommendations).

How did you proceed after your initial research?

I began the design phase not in front of my computer, but out at restaurants with friends, with a pocket-sized Mole-skine in hand. As we looked at the menu together, I was able to ask relevant questions, like “What do you think about when deciding what to order? What do you wish you knew?” I’d sketch up an interface idea on the spot and refine it through discussion over dinner. The ideas that emerged in the real-world context of use were almost always more interesting and relevant than the ones that I’d come up with sitting at home. Not to mention it was a great excuse to eat out with friends and try new places.

I filled a notebook with sketches, including pieces of interfaces (like stickers for your food photos) and main screens (what do you see when you’re looking foods at a particular restaurant?) before moving to the computer. Then, starting with grayscale mock-ups built using a stencil toolkit that I created for InDesign [available at Adaptive Path’s website], I began translating these sketches into screen designs. I thought about relevant scenarios—you’re at a restaurant and you want to see what’s good there; you’re going to a neighborhood and want to know what’s around; you’re looking for a particular food in a particular area—and created the screens needed to support these scenarios.

Were you able to user-test your early designs?

To test these designs on potential users, I printed them on card stock and cut them out [FIGURE CS3.1]. I carried this pocket-sized deck of cards in my pocket, and whenever someone was interested, I pulled them out and talked them through it. I would walk people through the scenarios, asking both high-level questions like “Who does this app seem to be for? What makes it different from other food apps?” and detailed questions like “How would you

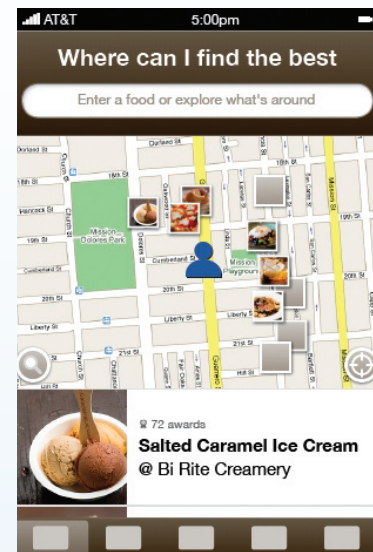


FIGURE CS3.1 Foodspotting start screen before usability research (Courtesy of Alexa Andrzejewski)

look up a certain restaurant?” Although I was able to get quality feedback based on the paper designs, when the number of wireframes got beyond ten, pulling out the right card at the right time became a bit tedious and not always the most convenient approach for spontaneous user testing, so I quickly looked for an on-device option.

Making an iPhone-friendly web site is not quite as obvious as it might seem: You need to create an image map that won’t load screens in a new window, you need to have a way to get rid of the browser chrome so your images are full-screen, you need to preload screens so that they load immediately. While I was able to use JavaScript code from WebKit¹ and HTML image maps to make my first prototype, I eventually found a Fireworks prototyping tutorial from UNITiD² that did all of the JavaScript and image mapping for me. Using this method, I was able to translate my wireframes into prototypes in less than two hours, so I could always test the latest version of my designs on users.

1. www.webkit.org.

2. UNITiD Interaction Action, “Prototyping for the iPhone Using Fireworks,” <http://unitid.nl/2009/04/prototyping-for-the-iphone-using-fireworks-cs3/>.



FIGURE CS3.2 Foodspotting start screen after usability research

Of course, when I was pressed for time, I could always export my designs as 320 x 480 JPEGs and simply add them to the Photo Albums on my iPhone. I could flip through these similarly to the paper prototype.

How did the user testing impact your designs?

About halfway through the design phase of Foodspotting, when I was feeling a bit stuck, one great conversation led to an “Aha!” moment that revolutionized the designs thereafter.

I was sharing the idea and designs with Megan Casey, the founder of Squidoo,³ when she expressed what I thought was the obvious: “What if your app really highlighted food photos and took advantage of all those people taking pictures of their food all the time?” While my initial thought was “Of course! That’s the whole point,” this feedback was the knock in the head that made me realize: Sure, that may have been my intent all along, but it’s clearly not coming across to users in the designs!

Foodspotting felt like every other local guide, not the “tantalizing collection of the best foods and where to

find them”—the words of my original pitch. Reframing the problem as “designing an app that highlights food photos,” I was able to completely revamp the designs, producing a beautiful interface concept that spoke this power much more clearly.

Foodspotting was part of iPhone Dev Camp. Can you discuss how that influenced your designs and process?

With detailed wireframes of this new, visual version of Foodspotting in hand, I went to iPhone Dev Camp to look for developers for Foodspotting [FIGURE CS3.2]. For better or for worse, up to this point my designs were driven by the ideal experience and my awareness of what was possible in other apps, not on the capabilities or limitations of the iPhone SDK. iPhone Dev Camp was a great opportunity to get in touch with the realities of iPhone development. I realized that all iPhone developers aren’t created equal. Some specialize in OpenGL and graphical games but have never used Interface Builder, whereas others can build beautiful self-contained apps but have never built an app using external data or an API.

iPhone Dev Camp was also a great opportunity to consider what a “weekend version” of the app could look like. What was the minimum thing that we could build that would actually have value? Thinking about the “weekend version” versus the “ideal version” was an exercise I had to come back to again later in the process, as I became more aware of the added complexity that each additional feature produced. Realizing that I’d rather have a simple app sooner than a full app later, I had to cut back on features and focus on what was essential several times until I came up with a set of wireframes that could reasonably be produced before the end of the year.

How have users responded to the app?

We launched the Foodspotting web site in January 2010 and the iPhone app [FIGURE CS3.3] at South by Southwest in March 2010, where we introduced it through a Street Food Scavenger Hunt and demoed its capabilities at a Tech Cocktail event. It was a great opportunity to test people’s reactions firsthand. Since we’d previously

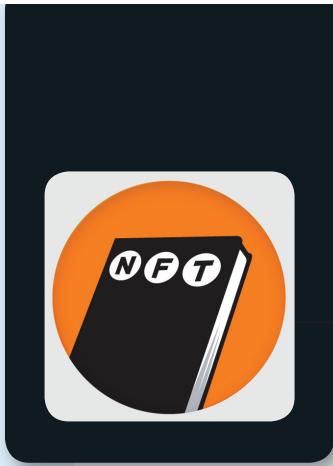
3. www.squidoo.com.

debated how much to emphasize food photos, we were surprised to realize that the idea of a “visual local guide that lets you see the foods around you” resonated with people even more than “finding and rating dishes instead of just restaurants,” the original value proposition.

People hear about the app and say, “I’ve been doing this already! Finally a place for me.” By giving this activity an identity and rewarding people for doing it, we’ve been able to attract over 20,000 food sightings in less than three months since launching. ■



FIGURE CS3.3 Foodspotting start screen version 1.3.2



CASE STUDY 4

Not For Tourists

NOT FOR TOURISTS is the ultimate guide for the savvy city dweller. Whether you've lived in your neighborhood for 55 years or 55 minutes, NFT will help you navigate and explore the city like a local. Written and designed by in-the-know city slickers themselves, NFT is mobile-oriented, cartographically inclined, and to the point. The *Not For Tourists* black book guides started in 2000. The companion series of iPhone apps launched in June 2009.

Tuft and Co. is a digital studio that creates innovative and engaging human-centered experiences for forward-thinking companies. Their passion is for digital pursuits and the people who use them, embodied by their work with rich Internet and mobile applications, customer-focused web sites, and user experience strategy and consulting.

How did NFT and Tuft and Co. get connected?

We had heard through the grapevine that NFT was casting about for a partner in developing an iPhone app. As a studio, we were already fans of the brand, so in the end we approached them about the project. What developed was an ideal partnership between the two companies to bring the NFT iPhone guides from concept to reality.

What inspired Not For Tourists to build an iPhone app?

The core NFT audience is fairly tech-savvy and “plugged in.” In addition to the books, NFT customers can access their web site and a mobile-accessible site (WAP). With the proliferation of smartphones, especially the iPhone, many people started asking about a companion iPhone app. NFT realized they had a unique value proposition since their books are compact and built for on the go—great qualities for an iPhone app. We tried to capture that on-the-go ethos with some of the features in the app itself, such as geolocation and embedded maps, which let you use the app anywhere (e.g., plane, train, or subway).

How did you approach the project?

At Tuft and Co. we try to design the user experience holistically. Our first step is always user research. We interviewed NFT editors as well as people currently using the books. From there, we developed a series of core user personas and scenarios which helped focus everyone on the key interactions and functionality of the app. As we update the app, we often refer back to these original personas, in addition to current user feedback.

How did you proceed after your initial research?

We began by sketching app designs and creating a paper prototype. We had hundreds of sketches on the wall, each representing a different scenario or interaction point,

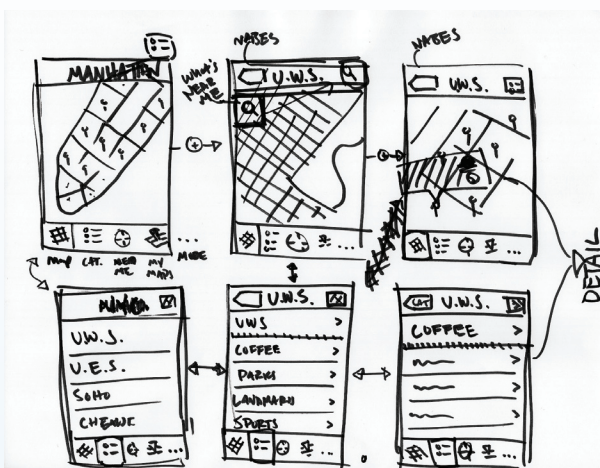


FIGURE CS4.1 Early flow and sketches

screen by screen [see **FIGURE CS4.1**]. The walls are the perfect platform for brainstorming and envisioning the information flows and how a person interacts with the application. We often burn through dozens of paper prototypes before committing to anything on-screen.

From the paper prototype, we created static screens with initial design elements and put those into an album on the iPhone. This gave us a better sense of how it would feel to use the app and helped us make many important decisions—how someone would interact with the maps, what features were useful, which ones were distracting. This prototype was also the jumping-off point for technical development.

Were you able to user-test your early designs?

We are huge believers in user testing. We conducted two different studies with the NFT app—a “man on the street” study and an internal beta study. For the “man on the street” study we wanted to get a sense for how the app would be used in the user’s natural environment. We followed participants with a video camera as they used the app and asked them questions about their experience.

It was an excellent opportunity to see how people would really interact with the app.

For the internal beta study we recruited people from NFT and Tuft and Co. who were not involved in the design process. They were given a list of tasks to complete with the app (they installed it via Ad Hoc Distribution) as well as a simple questionnaire which was returned via email. We ended up with fewer high-level insights, but it led to some significant UI improvements. A few bugs were also caught and fixed (which was one of our testing goals).

How did the user testing impact your designs?

There were several things that were changed based on user feedback. We found that users preferred lists over navigating by map and that functionality such as accessing maps at will—not being tied to having a cellular or WiFi connection—was very important to our users. As a result we decided to bundle the app content within the app; an Internet connection is required only for occasional updates. [**FIGURE CS4.2** illustrates later-stage user flows.]

How have your users responded to the app?

The feedback on the NFT app has been very positive overall. As with most application development, we continue to listen to customer feedback and make improvements. What we hear most often is that users want more and longer reviews. The reviews in the printed book are informative and succinct. However, NFT created a feature called “radars” for the web site, which are longer and more in-depth. Over the past few months we have been working to incorporate these into the app. We were recently promoted by the Apple iTunes Store for the holidays, which seems to be a good seal of approval in and of itself. [**FIGURE CS4.3** shows the final screens for version 1.0.]

What is the biggest lesson you learned while designing and developing this app?

That an app is never really “done.” It’s common to continually push updates to our apps, because of a new feature, a content or functionality update, or a change in the

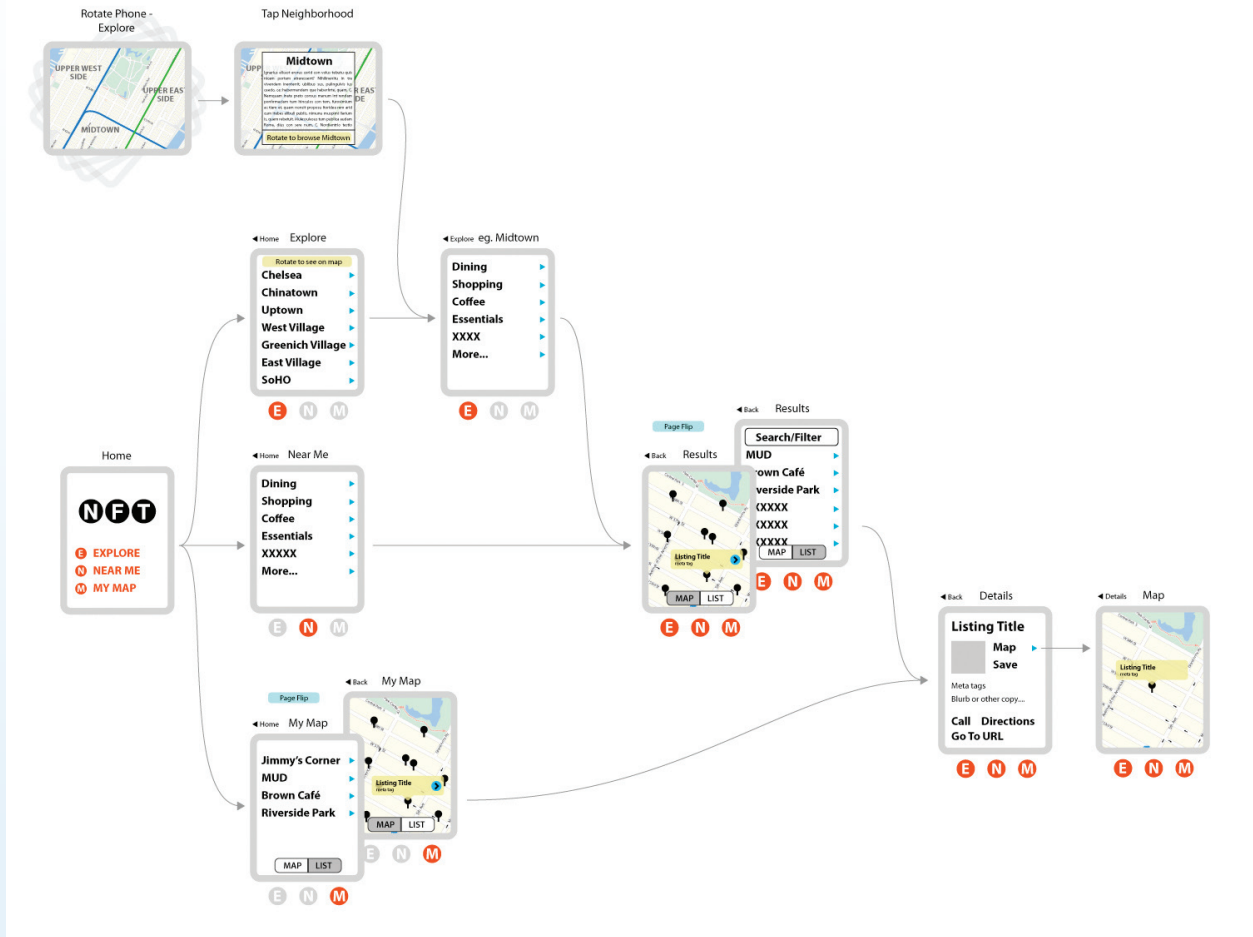


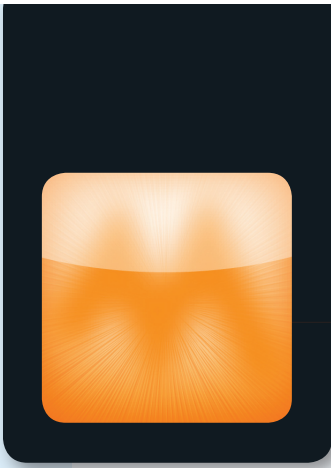
FIGURE CS4.2 Later-stage flow and sketches

software platform. Our partnership with NFT has been fantastic in that we are all able to keep the conversation going, to keep brainstorming on how the app could be improved. Unless you are specifically putting out a one-off app, it's important to remember that an app is dependent on a continual development life cycle and to plan accordingly. ■

(Not For Tourists icon and application screenshots courtesy of Tuft & Co.)



FIGURE CS4.3 Screens from Not For Tourists Manhattan 1.0



CASE STUDY 5

MUSE

MARGEIGH NOVOTNY is vice president of strategy and experience at MOTO Development Group, where she leads a cross-disciplinary team of design and technology professionals who develop next-generation product/service platforms for entrepreneurs and Fortune 100 companies. Prior to joining MOTO, Novotny founded the interaction design practice at Smart Design, where she drove the development of a wide range of user-focused products from housewares, mobile devices, and media servers to interfaces for automobiles, airplanes, and buildings. Novotny began her professional life as an architect with a focus on digital information and physical interaction with the environment.

How does MUSE work?

MUSE is an interface that visualizes your music library as a grid of dots; each dot is a track, and all tracks are playing. Tracks are organized into “columns” which reflect master genres such as classical, jazz, R&B, rock, alt, and so on [FIGURE CS5.1]. The tracks that I listen to most frequently rise to the top of the column (warmer-colored dots), and those I listen to less frequently float to the bottom of the column (cooler-colored dots). As the user drags a finger across the grid, the dots “play,” and the effect is like manually tuning a radio from station to station. The result is that users immediately understand the connection between touching the dots (tracks) and controlling the music.

Once users finds a track they like they can stop there; the track will continue playing and a playlist will be generated based on that track. Or users can stop on the track, lift their finger, and draw a loop around a group of tracks (dots) they want to include in the playlist. This allows for an easy way to, for example, create a playlist of my favorite jazz and R&B tracks. Or, conversely, a way to force me out of my listening ruts and easily generate a list by gesturing around a bunch of songs I listen to less frequently. It’s sort of a combination of Shuffle and Genius functionality. [FIGURE CS5.2 illustrates the different ways users may select tracks and create playlists.]

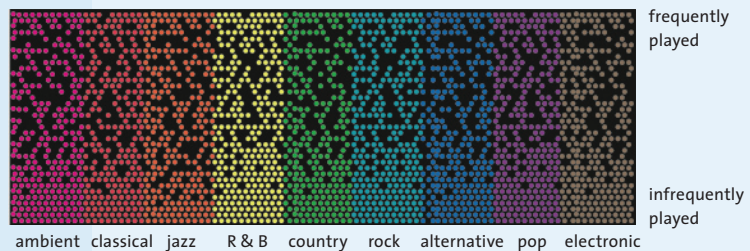


FIGURE CS5.1 Method for “blind navigation” of an audio collection

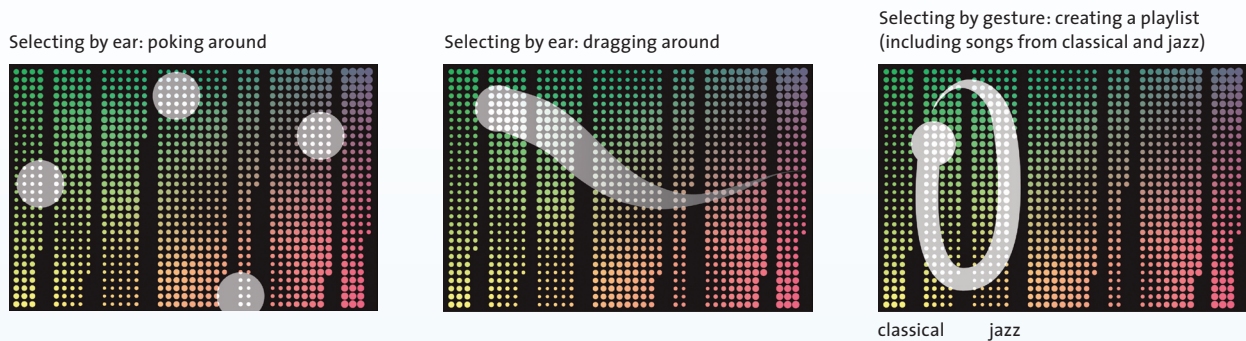


FIGURE CSS.2 Interaction concepts for selecting tracks on MUSE

What inspired you to create MUSE?

MUSE was born out of a desire for a more right-brain tool for navigating music libraries and creating playlists. Current methods for creating playlists are left-brain and laborious. The goal was to create something that didn't require reading or scrolling—a more intuitive interface that would allow users to just listen and feel music—and then use those results as the basis for a playlist. This interface was inspired by the beauty of flipping around an old-fashioned radio dial where songs hooked you by emotion, with the promise that the feeling would continue on that station. That's the experience I wanted to re-create.

I've always been drawn to tools like *SmartMoney's* "Map of the Market"¹—visualizations that take massive amounts of data and render it a way that's easy to process. I'm also a big fan of interfaces that allow the user to navigate in emotional or ambiguous ways—an example of this in the music realm is a web app called Musiccovery,² which offers a simple way to establish connections between music and mood and generates a (rather baroque) playlist visualization with jumping-off points to other artists and moods.

A gestural touch interface struck me as an obvious way to generate playlists quickly, because it's such a natural way

to manipulate information. iTunes on the iPhone didn't really adapt much when it migrated from a PC keyboard/mouse interface to a touchscreen device. It was an obvious opportunity to apply this idea to the iPhone. MUSE is designed to work on top of the iPod on the iPhone: It provides complementary functionality that extends the iPod app by providing another way to manipulate, visualize, and navigate the songs in your music collection.

What were your design goals?

Lots of interaction designers came out of the 1990s with web design interaction conventions baked into their model for designing interfaces on any platform (e.g., tabs, left-hand navigation, page-based structure, etc.). This is why we have seen these conventions replicated, to poor effect, on platforms like cable TV. That legacy is burdensome and inappropriate for many devices and content types. A mobile phone is a very different creature, and you have to optimize your design for the affordances of the phone. Likewise, iTunes isn't optimized for exploring the emotional content of music; it has real limitations.

Our goal for MUSE was to provide a compelling, ear-driven interface to allow blind navigation of the user's music library. We wanted to create an interface that was immediately understandable through gestural interaction, so that the functionality is revealed simply by using the app. We had a lot of discussion about whether the application

1. *SmartMoney*, "Map of the Market," <http://www.smartmoney.com/map-of-the-market/>.

2. www.musiccovery.com.

needed to replicate functionality available through the iPod app (e.g., search, use of metadata such as album covers, etc.) and concluded that this should be a complementary interface rather than a comprehensive replacement of all of the functionality in the iPod app. So in the end our goal was to make the interface so simple that even a two-year-old could use it. The MUSE interface is visually minimal, by intention, to allow for full focus on the audio.

What is design pliability, and how does it come into play with MUSE?

Maybe the best way to explain pliability is by reference to the architect Louis Kahn. Kahn is famous for asking building materials what they want to be as a way of getting to forms that reflect the nature and affordances of the material:

...and you say to Brick, “What do you want Brick?” And Brick says to you “I like an Arch.”

That’s the basic message of design pliability³: to let the content and medium drive the form of the interface. Take Google Maps; there are Google Maps interfaces controlled by a mouse, a D-pad, and a touchscreen, and the level of effort required to accomplish the same task varies a lot (i.e., some interfaces for exploring Google Maps content are just more “pliable” than others). *Pliability* means the extent to which the hardware or software interface enables an easier, more fluid and intuitive approach to navigation—so that up means up, for example. In a pliable interface, the linkage between eye and hand, or between action and response, is essentially seamless. Hardware/software interfaces are not all created equal, but in every case you want the interaction to be as effortless as possible by optimizing the interface for the specifics of the platform and the content.

MUSE is designed to encourage navigation by ear and as such provides a pliable interface for interacting with music. It’s a way to prioritize your senses of touch and

3. Jonas Lowgren, “Pliability as an Experiential Quality,” *Artifact* (2006).

hearing over a reliance on song titles and album covers. It gives the users random access to all the music they have, in a way that lets them hear what they’re choosing as they slide their finger across the interface.

How did your team approach the project?

MOTO’s design team is completely integrated with our internal technical team, so the collaboration between designer and developer was very tight. We didn’t start MUSE with a fully detailed spec. Instead, we worked back and forth: The design team made prototypes while the software team focused on emulating behavior and functionality. The design came together as a dialog between what was desired and what was achievable on this platform.

Along the way, we discovered some limitations about what the iPhone lets you do. We started developing MUSE on the first-generation iPhone—which meant we had no access to the APIs for the iPhone’s iPod application. Initially, we got around that by using a jail-broken phone, but luckily the iPod APIs became available when the Generation Two iPhones were introduced. That enabled us to transform the project into a real app, since the new capabilities provided authorized access to the iTunes library.

Another limitation we discovered is that the iPhone’s processor isn’t really fast enough to do all the buffering required to enable real-time audio playback as you move your finger across a sea of tracks where all tracks are playing simultaneously. To address that, we had to tune the software to mask the latencies that occur when moving from one track to the next. For the team, that involved a lot of back-and-forth. We had to work hard to refine the behavior to enable the features we wanted, given the capabilities of the hardware and the software.

How did the design evolve from the initial concept?

The look and feel of MUSE changed a lot from where we originally began. We started with a very basic visualization, a grid of pixels or clusters of objects. Gradually, the visualization evolved from a banal grid into a much more

interesting aesthetic inspired by the ideas of optical art—abstract two-dimensional images that create the illusion of texture or depth. Op art is very minimalist, but it’s evocative, and functionally it provides clear feedback to show where your finger is at any given point by magnifying an area of the grid. The switch to op art made the interface much more engaging, because as you drag your finger across the grid it creates a contour that looks magnified. The function of the app remained the same, but this visual expression moved the user experience in a new direction. [FIGURE CS5.3 shows the final app design.]

Do you have any advice for designers who are trying to move beyond the standard application styles?

There is a set of styles within Apple’s iPhone software development kit (SDK)—basic interface elements like sliders, rollers, scrollable lists, and so on. MUSE doesn’t use those elements in its main interface; instead, it’s what’s called an “Immersive” app. Immersive apps by Apple’s definition are “wall-to-wall” interfaces that employ a specialized way of navigating.

To the extent that SDK styles support familiar behavior that’s appropriate to the content type, the designer should stick to the existing styles. A simple rule of thumb is: Don’t stray from standard styles until it becomes very clear that you need to. But sometimes SDK styles don’t support the interaction you want to enable. Immersive apps make sense when your content doesn’t lend itself to the options afforded by Apple’s tools.

Ultimately, you have to do what the content demands. That’s what happened with MUSE—we wanted to create an Immersive form of visual navigation that just didn’t fit with Apple’s basic toolkit. Immersive applications are not always the right answer, but in the case of MUSE, we needed to create a new style to “let the brick be a brick.”

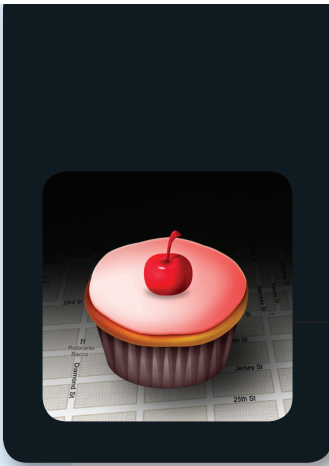


FIGURE CS5.3 Final MUSE design

What’s next for MUSE?

We didn’t start with the idea of creating an iPhone app; the idea started as IP around navigating any music library on any gestural touch platform—the iPhone was the most convenient way to demonstrate the power of that idea. Originally, we envisioned MUSE as a stand-alone touch remote, something like the Sonos device, or implemented in a wall-mounted touchscreen like Control4 provides. We still plan to develop the MUSE touch interface for navigation of home-media music libraries, in-vehicle audio, or in digital jukeboxes. So for us, the next step will be to put MUSE in any environment where you listen to music. ■

(MUSE icon, images, and application screenshots courtesy of MOTO Development Group)



DAN4 is a design practice dedicated to creating clear and engaging software applications, device interfaces, and multichannel services.

CASE STUDY 6

Prototyping at Dan4, Inc.

How do you prototype at Dan4?

We use prototyping essentially three ways at Dan4. First, we see prototyping as a natural part of the design process, allowing us to capture, communicate, and manipulate our ideas—quickly and fluently. In a way, prototyping is designing. It provides an agility that enables rapid iteration and fluid progression of a concept, as the design becomes more and more fleshed out. For us, creating prototypes is not a tangential task or a project luxury. It is simply good design practice.

Second, prototypes make excellent props during two key parts in the design process. During user research, the early days of a project when we are seeking inspiration, prototypes are great for stimulating responses that may reveal opportunities or risks about a concept. After the research has been gathered and synthesized, we use prototypes as material to support user testing, helping us identify design problems, validate designs, or gauge customers' response to a design.

Last, we like to find opportunities to reuse our prototypes and often do. In one case we illustrated a walk-through of a complicated user workflow in PowerPoint. After user testing and resulting modifications, it became the primary design specification document during the development process. Prototypes can also be adapted to add value to marketing efforts, product demos, and investor presentations, to name a few.

How do you choose your prototyping approach?

We factor in the usual constraints—time, budget, and scope—but also how the wider development team works and how the prototypes could be reused. For example, we will consider the tools they are already using, their development approach, workflows, and the degree of formality

with which they operate. From there, we choose the fidelity and the technology for the prototypes. The thoughtful consideration of the client team and what’s important to them is a key aspect of meaningfully communicating our design intent and maintaining that connection throughout the development process. In other words, before it can be respected and accepted, it must resonate.

We once worked on a location-based messaging platform for small retailers and franchisees. We wanted to help the shopkeepers envision the richness of an iPhone interface and felt that static, low-fidelity prototypes and mock-ups would not describe the user experience clearly. We opted to create a more experiential prototype, using Keynote [FIGURE CS6.1]. One of the useful things about Keynote for prototyping is that it offers many of the animations and transitions you see on the iPhone through Build Effects. It enables you to mimic the default UIKit transitions and animations and create more sophisticated behaviors involving fades, flips, zooms, translations, ease-ins, and ease-outs that can be developed using Core Animation.

It took a couple of days to build the prototype, using hand sketches of the rough interaction as the starting point. We used the prototype as a prop, along with mock-ups of other aspects of the platform, and spent three days conducting one-on-one user research sessions—paying informal, guerrilla-style visits with retailers in various neighborhoods.



FIGURE CS6.1 Keynote prototype for a messaging platform. A video of the prototype and “how-to” information can be found at www.dan4.com/prototyping.

But sometimes low fidelity is the way to go. During an innovation workshop with a network security systems provider, we spent a half-day creating a very “quick and dirty” prototype. We wanted to communicate the overall product concept but also examine a hunch we had about the practicality of the proposition.

Using photos of sketches, a crude stop-frame-style animation, an ambient soundtrack, and sounds sourced from the Internet, we created a demonstration that helped the attendees, mostly software developers and managers, quickly gain a common understanding of the concept and an appreciation of the relevance of context of use [FIGURE CS6.2].

Any other advice on iPhone prototyping?

In our experience, it is best to try and start prototyping app concepts as soon as possible. We have found that prototypes are most effective when used to probe the underlying ideas and assumptions around the concept and elicit user insights that help teams figure out where to apply their effort.

Getting early input from others, especially from the people whom you hope will be using and buying the product, provides you with information to support the early

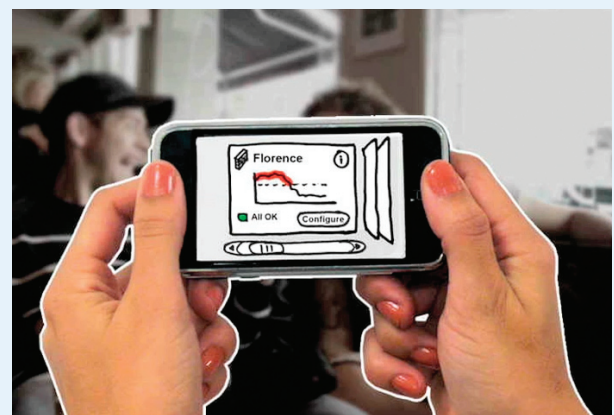


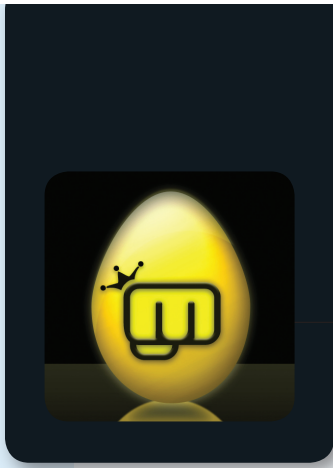
FIGURE CS6.2 Sketch and video prototype for a network security app. A video of the prototype and “how-to” information can be found at www.dan4.com/prototyping.

strategic decisions that set the project trajectory and strongly influence how the end product is conveyed.

Often it's better to create several simple prototypes that probe separate aspects of the product. For instance, the essential functionality and overall architecture could be prototyped and tested using paper wireframes or a simple interactive prototype. But the branding, look and feel, and interface behaviors may be better tested using static visual mock-ups or an animated walk-through.

Prototyping at its best is about creating tools that probe the right questions and enlighten the design—as long as it doesn't knock you off schedule or become a distraction from other project tasks. Keep in mind, it's just as important to know what to exclude from the prototype as it is to know what to keep in, always striving toward “as simple as possible, but no simpler.” ■

(Images courtesy of Dan4, Inc.)



CASE STUDY 7

What's Shakin'

Upon graduating from Caltech with a B.S. in engineering and applied science, **MATT PAUL** moved to Silicon Valley to join Adobe's Advanced Technology Group (ATG) in 2000. In the decade following, he became a founding member of start-ups big and small, such as price comparison innovator StreetPrices, merchandiser to the stars SeenON!, and the veritable TiVo of the web, StumbleUpon, which was acquired by eBay in May of 2007. Nowadays, as founder of mopimp productions, Matt is focused on the intersections of real time meets rhythm, and location-based services meet game mechanics, but he freely admits that by the time this book is published, he might well be working on something else entirely.

How did you get started doing iPhone development?

I first got my feet wet developing for the iPhone in the summer of 2008 at iPhone Dev Camp 2 in which my hack-a-thon team's app, Fwerps, won best app by a group of new/first-time Cocoa and iPhone SDK developers. Soon after, I began prototyping an app for StumbleUpon, my employer at the time, that would allow users to stumble on and rate news articles, photos, and web sites from the iPhone. I had a lot of fun building for the iPhone platform, but that prototype wasn't destined to make it to production, and I desperately wanted to lay claim to being part of the iPhone developer revolution by having an app of my own for sale on the App Store.

What inspired you to build What's Shakin'?

Fast-forward to summer 2009. My friend Hunter Peress, an Android developer, and I thought it would be fun to collaborate on a cross-platform mobile app together. He had already released an app called Wifi Hunter to the Android market but was looking for a change of focus. I still had yet to make my App Store debut, but having just attended Apple's World Wide Developer Conference (WWDC), I had grown ever more motivated to do so.

Since both Hunter and I are musicians, working together to create a musical instrument was an obvious choice. We started brainstorming around what we might find fun and would conceivably want to use ourselves. I had been known to dabble in drumming on and off over the last decade, whereas Hunter regularly performs as a hand percussionist; hence percussion was a natural area for us to explore.

The question remained: Could we make a realistic musical instrument that was played via dance and motion? We asked ourselves what instrument would lend itself best

to our collaboration. We surmised it would be one that you could hold in your hands like a clave, a wood block, or an *egg shaker*—perfect! Finally, we just needed a catchy name; once Hunter suggested *What’s Shakin’*, we were all set to get started.

What kind of up-front research did you do?

I decided to put myself in the user’s seat and get hands-on experience with other apps that were already available. I looked around the App Store myself and asked friends to recommend apps to check out as well. A couple of friends recommended *Drum Kit*, a popular app in which the user taps on the different parts of a photo-realistic drum kit to play the drums. That was along the lines of what I had in mind—intuitive, realistic, and fun to play—but how was I going to differentiate my instrument from the others?

At the time, it was far more common for musical instrument apps to be played via tapping on the device’s multi-touch screen. There was an alternative that but a few had employed—making use of the accelerometer. Nowadays there is no shortage of shaker apps. Over the course of development, I must have tried those of at least ten competitors. Some had nice visuals, some came with a good selection of instruments to pick from, but none of the lot did justice to the experience of playing an acoustic musical instrument; they simply lacked the responsiveness required. There was definitely an opportunity here to improve the state of the art.

How did you start the design?

Our initial approach was to emulate the sound created when you play an acoustic egg shaker by modeling the individual beads inside and their interactions with the eggshell and one another. For instance, let’s imagine we could peer inside an egg shaker and observe what exactly goes on in there when you play the instrument by shaking it.

Before you begin playing the instrument, all of the beads are at rest in a lump at the bottom of the eggshell. While you’re playing the instrument, each shake accelerates

the eggshell in the direction of motion while the beads are effectively thrust against the back of the eggshell as it carries them forward. Then as one shake ends and the next begins, the direction of the eggshell changes and the beads come into contact with the opposite side of the eggshell. Every time one of the beads hits the shell, or comes into contact with another bead, a tiny sound is made. In aggregate, all of these tiny sounds combine to yield the slushy egg shaker sound with which we’re all familiar.

In theory, we could replicate what goes on inside an acoustic egg shaker via a physics simulation (e.g., modeling each bead’s position in 3D egg space) and calculating when, where, and how the beads would collide with the eggshell and one another. Every time a collision should occur, we would play an audio sample of a bead hitting the eggshell, or one bead hitting another, accordingly. Repeat ad eggium.

We knew that this approach would prove challenging. Sure, we could make some simplifying assumptions and disregard bead-bead interactions, but it would likely take a lot of time to get things right and a lot of computation to pull off in a realistic manner. Our operation was bootstrapped after all; we wanted to get to market quickly or fail fast and iterate from there.

Though we ultimately shelved the approach of modeling each bead/collision independently, it was through the consideration of how we might make use of OpenAL to position the sound made by each collision in 3D space that led us to the winning formula.

What did you try next?

Soon enough I realized that a hybrid approach leveraging OpenAL in conjunction with the device’s accelerometer would be sufficient for our purposes. OpenAL is a cross-platform 3D audio API that allows developers to easily position sounds in 3D space and create sound effects such as the Doppler effect (the perceived change in pitch of an emergency vehicle’s siren as it races toward, passes, and finally races away from an observer).

OpenAL afforded us plenty of control over the shaker's sound and gave us the ability to modulate it according to the user's style of play. We were even able to expose a parameter on the Settings page that allows users to vary the number of beads in their egg shaker and produce a more staccato or "slushy" sound accordingly.

Were you able to get user feedback before launching?

One of our goals from the beginning was to build an instrument that could be used to play along with music. First we tested it with our own music—Hunter used it with his Brazilian drums and I tried it while practicing my DJ set. Then we had lots of our friends test the app. I live in a household of DJs and musicians of different musical styles so there was a nice range of input. On several occasions I would shake along to rehearsals of my roommates' band, The New Up, who practice in the room adjacent to my home office.

All of the feedback we received in house was great, but I wanted to know how the app would work in the real world—could people hear it in a noisy bar? So I brought an early beta version along with me down to North Beach to try out What's Shakin' at Giordano Bros.

How did the people in the bar respond?

I didn't really know what to expect, but my fellow bar patrons were rather enthusiastic. Actually, one in particular grabbed the shaker out of my hand so she could play along to the band! People wanted to buy the app right then and there, but it didn't even have its look and feel finalized yet; at that point I believe the app was still using a placeholder image that I didn't have the rights to.

The most concrete takeaway from our real-world testing was that the app's parameters needed to be configured a certain way to enable users to show off the app to their friends in a loud environment. Such a configuration might not sound as good or feel as realistic in a quiet room; hence the app's Settings page would need to offer a fair amount of flexibility to accommodate both ends of the spectrum.



FIGURE CS7.1 What's Shakin' app

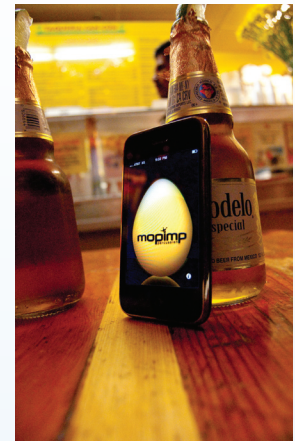


FIGURE CS7.2 What's Shakin' app in context

How did you know the app was done?

Throughout the development process, I would play my acoustic egg shaker and compare its sound to that of the latest and greatest version of the app. All summer long, my roommates were constantly subjected to hearing this comparison of plastic- versus silicon-based egg shaker technologies; I'm sure it drove them mad. One day, after numerous iterations, they could no longer tell if I was playing acoustic or using the app from the other room without running in to see for themselves—that was when I knew we had our emulation down pat and were ready to launch. [FIGURES CS7.1–CS7.2 show the final app.]

How have users responded to your app?

User response has been great. Almost without fail, people respond with encouragement when they meet someone who's building an iPhone app. Of course, compliments come cheap, but this reaction comes across as genuine, I think because at its core, the allure of quitting your day job to become an indie developer really harkens back to, and is just the latest variant of, the American Dream ("life, liberty, and the pursuit of happiness").

Not to suggest that this gamble is any less risky or more lucrative than any other—no one hails the career move as a particularly sound business decision, for instance—but people can tell you're going for it, shooting the moon, and there exists a near-tangible socioeconomic undercurrent of support for such attempts, akin to whatever it is that drives our compulsion for, and satisfaction derived from, rooting for the underdog.

What's next for What's Shakin'?

We're very excited to continue building upon What's Shakin's launch and have made plans to add a greater selection of sounds for users to unlock, the ability for users to record and share their performances with an online community of fellow shakers, and leaderboard rankings and scoreboards so users can boast and brag about their brilliant shaking ability/past performances. The pie in the sky for us would be to create a Shaker Hero—esque franchise of games and levels for users to play. ■

(What's Shakin' icon and application screenshots courtesy of Evan Dudley)



CASE STUDY 8

REALTOR.com

CLIFF WILLIAMS has been designing and building user experiences for the web and desktop for the last 15 years. He's currently the interaction design lead for mobile user experiences at Move, Inc., operator of Move.com, Moving.com, SeniorHousingNet.com, Top Producer, and REALTOR.com, the number-one site for home sales.

What inspired REALTOR.com to build an iPhone app?

A big part of searching for a home is being out and about—checking out neighborhoods, going to open houses, touring with an agent. A REALTOR.com iPhone app felt like a natural and useful extension of our web experience.

One of our developers created the spark with a proof-of-concept app. That pulled in me and a couple of others at the company. From there we spent whatever extra cycles we could find—between projects, nights, weekends—refining that proof of concept. Eventually it developed enough interest and became an official project (great apps from our competitors helped, too).

How did you approach the project?

Early on it was a mix of open brainstorming and continuing refinement of the proof-of-concept app. This was a great approach for us because it let big new ideas flow in while we were learning the bounds of the API and the capabilities of the platform.

Once it became an official project, we took a step back to evaluate where we were with the proof-of-concept app, the pool of ideas we had developed, and past user research that might be applicable. This helped us really hone in on a feature set focused on the mobile user.

What were some of the challenges you faced?

As an experienced web designer but first-time app designer, it took a while for me to really get comfortable applying the patterns and principles of the platform. My first attempts were essentially 320 × 480 web sites, not iPhone apps. Data was displayed in long sections that required lots of scrolling. The architecture focused on presenting a breadth of choices at the expense of a crisp, simple workflow. As the app evolved, it became more and more iPhone-y.

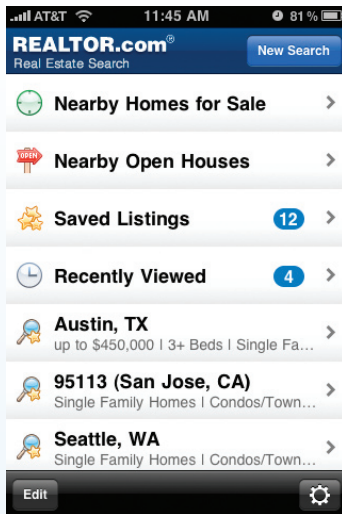


FIGURE CS8.1 REALTOR.com home screen

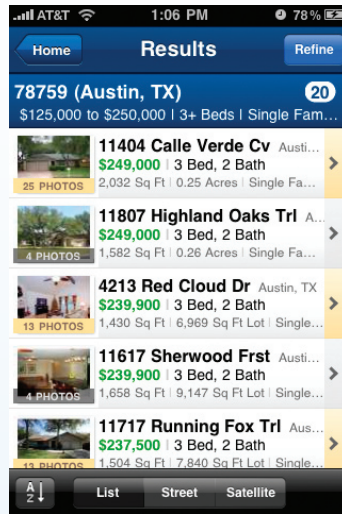


FIGURE CS8.2 REALTOR.com search results

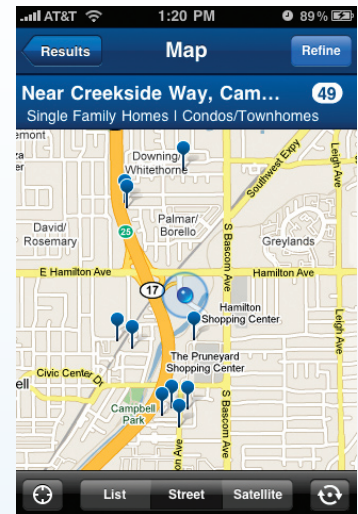


FIGURE CS8.3 REALTOR.com map view

Were you able to conduct usability tests?

After our first big round of design on the official app, we started thinking about usability testing. Getting a live prototype in time wasn't possible, so I started experimenting with simple HTML prototypes that we could test on a working iPhone. We could get close to what we wanted with these but not all the way (scrolling with fixed-position elements was the trickiest part).

We ended up going with a paper prototype that included all the screens in the app, several variations of the key screens, and a handful of bits and pieces that we would overlay at different points. These were created with Adobe Fireworks at 72DPI. Next time I would create them at 300DPI since the printouts weren't that sharp. I acted as the computer and one of our researchers facilitated the sessions.

Testing netted one big issue (saving your favorite items was confusing) and several smaller ones (clarity of icons and copy). These results pushed us even further down the reduce-simplify-streamline path our designs had been evolving along. [FIGURES CS8.1–CS8.3 show the final designs.]

It was really interesting to compare the behavior of our iPhone users with folks we bring in to test our web site. The iPhone users explored without hesitation—in fact, several stated early on, “First I'd go around tapping on everything to see what it does.” We always see foraging in our web users but rarely this kind of proactive exploration.

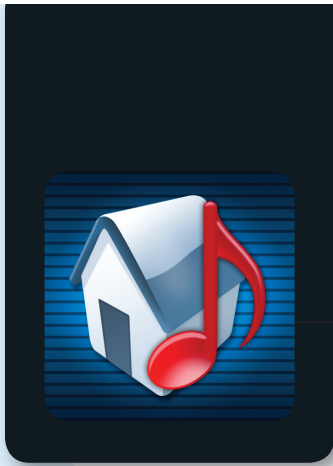
Can you describe the usability-testing environment?

Evonne Shea, our project's user researcher, would start each session with a brief interview that gathered high-level information about the user's current real estate search needs and experiences. At the end of that interview she'd introduce me (the computer) and would move to the adjacent observation room. From there she could observe the participant through a one-way mirror as well as a pair of video cameras in the room—one focused on the prototype, the other on the participant's face. Video was recorded for later analysis and also displayed on a projector in the observation room so team members could watch. We ran eight participants. They were all iPhone users who had some experience downloading apps and were currently involved in searching for real estate.

Any additional advice for iPhone app designers?

If you're not developing the app yourself, learn to speak the same language as the folks who are. Read Apple's *Human Interface Guidelines* to start. It will help get the ball rolling on your designs, and it's invaluable as a Rosetta Stone. Check out the online API documentation, too. For purposes of specifications, it's critical to know what the API provides "for free" and what it doesn't.

Another thing that helped me a ton was creating a giant library of screenshots. Download every app that you hear good things about and screenshot anything remotely interesting you see. Perhaps even more important, screenshot all the built-in Apple apps (even the boring stuff, like Settings, is a gold mine). I constantly referred back to this library for best practices on some of the more mundane things and inspiration for big new ones. ■



CASE STUDY 9

Sonos

ROB LAMBOURNE is the director of user experience at Sonos. He designed the Sonos iPhone app with a team of designers and developers at Sonos (www.sonos.com) and with Oliver Bayley, an independent UX consultant based in San Francisco. Rob earned his master's at London's Royal College of Art. He started his career as an interaction designer at Philips Design in the Netherlands and California.

Sonos is the leading developer of wireless multi-room music systems for the home. The Sonos Multi-Room Music System is the first wireless multi-room music system that lets you play all the music you want all over your house and control it all from the palm of your hand. Sonos liberates the music stored on a computer so it can be enjoyed all over the house.

What inspired Sonos to build an iPhone app?

Sonos has had a handheld controller on the market since early 2005. While this product was terrific for its time, it lacked the speed of a touchscreen for entering artist and track searches that are central to the Sonos experience. When Apple released the iPhone, we were already part-way through designing our own touchscreen controller. By creating the iPhone app, we were able to provide our customers with a touchscreen experience several months before our own controller was due for release. In addition, it was clear that the iPhone was going to be a huge hit, and since Sonos customers and iPhone users overlap demographically, our strategy team saw an opportunity to appeal to the growing number of iPhone customers by providing a Sonos Controller app for them.

Can you describe the design process at Sonos?

Once we have a brief from our product management and strategy teams, we work very iteratively to design our products. We start by identifying and understanding our customers. This can take the form of contextual interviews, online surveys, and focus groups. This process leads to user personas and the description of typical day-in-the-life scenarios that we use as frameworks for idea generation. Our user experience design team then sketches designs and makes prototypes as quickly as possible. For instance, before the iPhone app became a project for us, we built a quick wireframe UI prototype in Adobe Flash, which we ran on a Nokia touchscreen device. This allowed us to get a sense of the flow of the interface and identify our main design challenges. However, because the iPhone is such a sturdy development tool, we found that we could work with our software engineers to create working prototypes in less time than it took to build a comprehensive Flash demo.

Was the iPhone app design process different from your typical process?

We were able to work much more quickly with this project than is typical. We were not making a hardware controller from the ground up, so this project became all about software—saving us many months of development time. Apple also provides very strong API and user interface guidelines that we used effectively. But the biggest difference was our ability to test very thorough prototypes quickly on the real platform (not a platform mock-up).

What were some challenges you faced? Do you have any advice for app creators facing similar design challenges?

Our biggest challenge with UI design at Sonos is to ensure that while we offer very powerful functionality, it must be presented in a way that everyone in a home (not just the tech-savvy people) should be able to understand and use. We aim to never let our UI get in the way of what people are trying to achieve, which at its simplest level is to enjoy music throughout their homes. To this end, we often find ourselves deciding not to build additional features, or limiting the scope of some designs in order to protect the simplicity of the experience. I think this is a challenge for all app designers—to focus on what is really important. Don't be afraid to cut a feature or simplify a design if you are trying to appeal to a broad audience.

For Sonos, it was also important not to get caught up in the throwaway nature of iPhone apps. Generally speaking, iPhone apps are available for little or no cost and there are thousands of them, so it might be tempting to approach their creation with a little less rigor and thought than one might ordinarily apply to a design project. In our case, to follow this temptation would have been a big mistake. I think the success of our app is largely due to the fact that we treated this product as seriously as we treat our other products that sell for much more.

Were you able to conduct usability tests?

We ran usability tests in three stages. First, we tested an early iPhone prototype with 8 to 10 nontechnical family



FIGURE CS9.1 Sonos Zone Menu

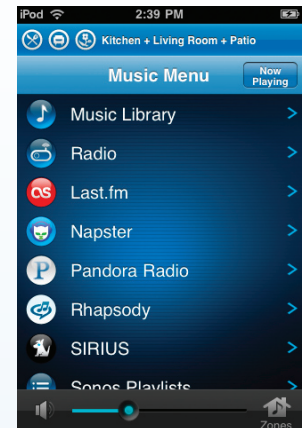


FIGURE CS9.2 Sonos Player Menu

members of Sonos employees. We improved the design based on their feedback and then conducted further tests in a usability lab with current Sonos customers, as well as iPhone users who had never heard of Sonos. We ran this test with about 15 people. Last, once we had a more complete app, we ran an alpha test in which 30 existing Sonos customers used the iPhone app to control their Sonos systems in their own homes. In between these tests, the design team iterated the visual and navigation design and created functional specs that were used to implement the final application.

How did usability tests impact the design?

In addition to testing the general appeal of the app, we were able to test the navigation model, the visual design, and any number of details like legibility of text and icons. In our early prototypes we found that the hit areas on the touchscreen were too small for some people, so we made them larger, rearranged some of the UI components to accommodate the changes, and increased text sizes throughout the app. We also found that users were not understanding one or two state transitions in the UI, so we made use of Apple's built-in animations to help convey navigation more clearly (but beware of overusing these animations).

How have your users responded to the app?

Our customers have responded really well on two levels: They are happy they can experience Sonos at a lower cost, which is ultimately good for the sales of our other products, and our users just love the idea of controlling their music simply by taking out their iPhone and launching the Sonos app. [FIGURES CS9.1–CS9.4 show some of the app screens.]

Any other advice for iPhone developers and designers?

Make use of Apple’s guidelines and all the widgets they provide for app creators. Look at other apps for inspiration to find out what works and what doesn’t. Hire good UX designers to create your app. Focus on simplicity and always talk to potential users of your product—they won’t tell you how to design your product (nor should they), but you will always learn something useful that you can apply to your next design iteration. ■

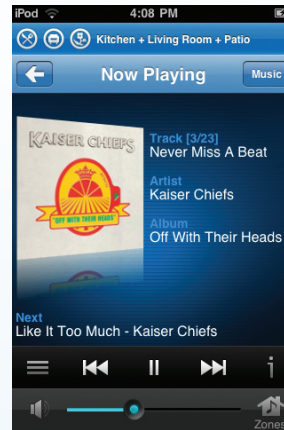


FIGURE CS9.3 Sonos Music Menu

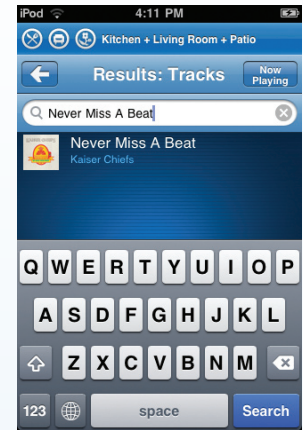


FIGURE CS9.4 Sonos Search



CASE STUDY 10

FlightTrack

MOBIATA creates best-selling mobile travel applications. Mobiata's FlightTrack app has topped the iPhone travel best-seller list ever since the product's launch in November 2008. Benjamin Kazez founded Mobiata in December 2008 and currently serves as president. He has spearheaded the development of Mobiata mobile travel applications such as FlightTrack, HotelPal, and TripDeck.

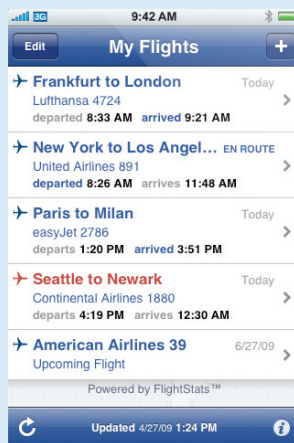


FIGURE CS10.1 My Flights screen

What inspired you to create FlightTrack?

The idea came to me when I was traveling in November 2008. I was at the airport and needed to look up my flight number. After digging through my backpack, I finally unearthed my boarding pass, which was already crinkled and falling apart. At that moment I thought, "Why isn't this on my iPhone?" My iPhone is always in my pocket and easy to access; it would make much more sense to access the info from there. I could use the built-in Notes app, but it would be much better if there was an app that knew my flight number and could automatically update the info if there were any delays. In addition to travelers, I thought this app could be useful for people who are picking others up at the airport.

How did you approach the project?

At first I did a lot of pencil sketches of various user interface possibilities. I spent a lot of time trying to figure out what was the bare minimum needed to find flights. There's not much room, so it was essential to come up with a flight summary that would fit on the screen and not overwhelm users. To this end, I considered what information we could omit from the flight list view based on what the user already knows. For example, I decided to place baggage and aircraft information in the detail view since this was lower priority for most users.

What were some of the design challenges you faced with the detail view?

Based on my own experience, I knew users would be walking around the airport and periodically checking for time and gate number; thus I decided to make this information the most prominent. And around that information are the annotations to the data, such as the terminal and miscellaneous gate details. We went through several iterations of this view. It was a tough challenge since there's a lot

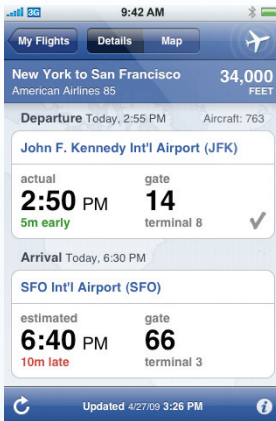


FIGURE CS10.2 Flight detail screen



FIGURE CS10.3 Map view for selected flight

of flight data available—scheduled time, estimated time, runway time, gate time, and much more.

What made you decide to allow multiple flights?

From the beginning I knew that I wanted people to be able to track multiple flights at once. On most web sites you can track only one flight, but it's much more useful to allow multiple ones. This way, if you have connecting flights you can see them all at once. Our professional users—taxi drivers and limousine drivers—really value this feature. [FIGURES CS10.1–CS10.3 show a few of the app screens.]

Any advice for designers and developers working on similar products?

Getting external content to work with the iPhone can be a real challenge. We spent a lot of time with our content provider, Flight View, to make sure their data fit well within our application. Many of our competitors weren't taking the time to work through these kinds of details, but they are really important for the user experience. Even though it was a lot of work, going the extra mile really paid off.

How do you stay in touch with your customers?

For the first three to four months I answered all of the support emails and tracked how many users were suggesting

certain improvements. When I was ready to plan the next release, I used that as a way to prioritize features. Now we have a dedicated support person—it's crucial to have a connection to your users. Support regularly summarizes user feedback, and we make sure everything is accounted for in our bug tracker. We are constantly asking ourselves why users are requesting a particular feature and how we can improve the design.

What's next for FlightTrack?

A few months after FlightTrack, we released a professional version called FlightTrack Pro that offers more detailed airport delay and closure warnings and synchronization with third-party itineraries. FlightTrack Pro has been localized into English, German, French, and Spanish, and the app works worldwide.

Our HotelPal app enables users to browse and search hotels worldwide with live availability and rates, then secure a room with easy in-app reservations, powered by the Travelocity Partner Network. HotelPal is perfect for last-minute plans, canceled flights, road trips, or just exploring.

Our latest app, TripDeck, offers complete itinerary management for virtually every component of a business or personal trip, including flight information and live flight status, car rentals and driving directions, hotels, meeting times and locations, restaurants, trains, and other related travel details. ■



CASE STUDY 11

USA TODAY

MERCURY INTERMEDIA collaborated with USA TODAY on the design and development of its iPhone app. Mercury has been helping major brands reach their customers through new and emerging technologies for over 15 years. Mercury started as a traditional web development company, transitioned into desktop application development, and recently refocused exclusively on high-end best-of-breed mobile development. Mercury Intermedia provides a full range of services for its clients, including programming, graphic and UI design, content hosting, and analytics.

How did Mercury Intermedia get involved in the USA TODAY app?

It is a bit unbelievable looking back at it, but the USA TODAY app was our first iPhone app and our first venture into the mobile space. Before we shifted our focus to mobile development, Mercury Intermedia had been working with companies such as *Sports Illustrated*, *Golf* magazine, and several NFL teams creating proprietary, network-enabled desktop applications. What we had been working on in the personal computing space translated incredibly well to mobile. We were very fortunate that USA TODAY and its VP of mobile strategy and operations, Matt Jones, saw our potential and gave us a chance to prove ourselves on the iPhone with the USA TODAY iPhone app.

How did you approach the project?

It is vital when first starting out with a new device such as the iPhone to study the existing UI and maintain consistency when at all possible. Usability should always trump uniqueness when employing nonstandard approaches to presentation and navigation. If the nonstandard approach doesn't make the application easier to use or desired functionality quicker to access, it's best to stick to the *Human Interface Guidelines*. In USA TODAY's case, where new UI approaches were necessary, such as maintaining category access via the slider bar in Headlines, we looked to educate the user in ways that were quick and unobtrusive. When users enter the Headlines section of the USA TODAY app, the sections in the slider navigation bar slide into place from the right of the screen each time the section is opened until users slide the bar for themselves. Once users slide the bar once, we store this action and never auto-animate the bar again. This is a great way to train those users who are having trouble recognizing the control without having to add a dialog window or interfere

with those users who do understand how the slider control works.

How did the USA TODAY brand impact the iPhone app design?

USA TODAY has always been known for large color photos, graphics, and diagrams, along with short summarized news articles, all of which translated perfectly to the iPhone. Snapshots proved especially successful, partially because of the unique ability the iPhone gave us to sort voter responses by location through the built-in location services; we often saw more user voting from the iPhone app than from usatoday.com.

Color-coded sections are also a vital part of the USA TODAY brand identity, and it was critical that we incorporate them into the USA TODAY app gracefully. As in the paper and the web site, we wanted viewers to know immediately when they saw red in the app to associate it with sports and green with money, and so on.

How did you evaluate your initial designs?

No one knows USA TODAY's readers or the brand as well as they do, so we felt it was important to keep them involved in all stages of the design process. We were fortunate to have a great team to work with at USA TODAY. Within USA TODAY's design group, Chuck Rose and William Couch were invaluable. William was USA TODAY's creative point person on the project and we worked closely together throughout the project, bouncing ideas off of one another while wrangling over nuance.

Early on we pulled the full project team together to lock down features and the general structure and presentation of the app. Once that was established, William and I did much of the back-and-forth design tweaking and revising until we had a beta version of the application ready to hand back over to the full team to test.

Once we were in the beta phase, we rolled out the early builds internally within Mercury first and then expanded the beta to the USA TODAY team. When we had a build both groups were comfortable with, we expanded the beta to include people within USA TODAY who were outside the iPhone team as well as a select group of external beta testers.

How did the designs evolve over the course of the project?

The Headlines section saw the most evolution throughout the design process [FIGURE CS11.1]. In our early designs, the Headlines article list was presented using a similar accordion scheme to what we used in the Weather and Pictures sections. The accordion approach looked and worked great for Headlines. It was an interesting departure from the standard list view that almost all other news applications use, and we were really looking for ways to help USA TODAY stand out and differentiate itself. The seven Headlines categories fit perfectly on screen, leaving room for one headline in the selected category that the user could easily page through to view the next or previous headline. We put together a working demo, and it looked and felt unique. However, as unique as it was,



FIGURE CS11.1 Progression of the Headlines section

we quickly realized that it just wasn't easy enough or fast enough to consume a group of headlines quickly, which is what most mobile users demand, so we returned to a more standard headline list view in that section.

Once we decided to move to a list view in Headlines, we had to figure out how we were going to deal with news categorization. Since USA TODAY is a multipurpose application, simply creating a tab bar section for each news category wasn't a viable option. Lumping seven news categories in with four non-news features wasn't ideal, and we noticed in our own iPhone use that anytime content or features were placed behind the tab bar More button, they generally went unnoticed and unused. We weren't willing to sacrifice any of the main sections within the application by burying them behind the More button.

At this point, USA TODAY suggested we try the section slider bar approach. We had seen how successfully the slider bar approach worked for the original Facebook app developed by Joe Hewitt. Once we locked in on the slider bar approach, we wanted to keep the look and feel of the slider control similar to Facebook's approach. We hoped that by modeling our slider control closely with Facebook's, it would encourage a standard that other applications would adopt and would make it more familiar. To make sure we weren't offending anyone at Facebook, we even emailed Joe Hewitt and got his blessing before moving forward. [FIGURES CS11.2–CS11.4 show some of the final screen designs.]

Since the USA TODAY application launched, news apps from CNN, *Time*, CBS News, and the *Financial Times* among others have all followed suit with slider bar section navigation. (It is worth noting that the Facebook application has since abandoned the slider bar with its second version release.)

When you look back at the overall process, what was the biggest challenge?

The biggest challenge of the USA TODAY app was finding ways to present so much and such varying content to the user in a way that wasn't overwhelming. We strove to keep

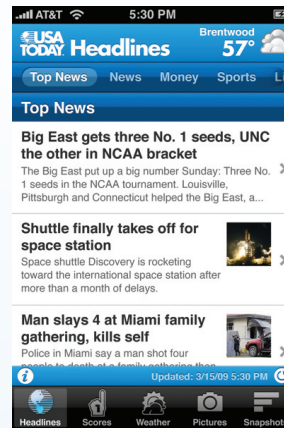


FIGURE CS11.2 Headlines view



FIGURE CS11.3 Photos view

the content presentation and organizational approach unique and fresh while also keeping the content accessible and clear to users. The goal was to keep primary content and features in each section a single tap away and secondary content no more than two taps away.

It is important to limit the number of steps to access primary content and to keep access to that content visible and easily accessible when at all possible. Whenever we can streamline a process and eliminate steps, we feel it is essential to do so. Mobile users are often looking to fill small gaps in their day. Although they may not always know how to verbalize it, users demand that load times be quick and that the app's UI be efficient.

How have users responded to the app?

User response to the USA TODAY app has been overwhelmingly positive. We've been fortunate to hear from a number of users who have said that they became daily USA TODAY readers after using the iPhone app. Although we have been blown away by the well over two million installs of the application, we are even more pleased with the substantial user retention. Install numbers are sexy and tend to get all of the attention, but the number of those users who continue to use the application on a

daily basis is much more important for an ad-supported application such as USA TODAY.

Along with positive user response, industry reviews have also been very favorable. *Macworld* rated the app four or five mice, saying, “USA TODAY for the iPhone is fast. Very fast. The app is graphically pleasing, and makes reading the day’s top stories easy and pleasurable.” O’Reilly featured the application as its pick for best general news app in “Best iPhone Apps,” claiming, “... no other app beats USA TODAY at quickly scanning what’s happening in the world. The app’s fast, efficient display lets you hop quickly among sections, articles, sports scores, and daily photo galleries.” *Communication Arts* featured the USA TODAY app in its juried 2009 Interactive Annual, stating, “This app packs an impossible amount of fresh information into an elegant and intuitive experience.”

What’s next?

We are constantly reevaluating and looking for better and more effective ways to present information and have several new features and improvements planned for the USA TODAY app. Two of our most requested features, font resizing and offline content access, will be added soon. Users can also expect to see improved subcategory navigation in Headlines as well as a more robust Scores section in the coming year. ■

(USA TODAY icon and application screenshots courtesy of Mercury Intermedia)



FIGURE CS11.4 Snapshots



CASE STUDY 12

Voices

VOICES was created by Taptivate and tap tap tap, one of the leading iPhone app development companies. tap tap tap had apps available the day the App Store launched and has sold well over a million copies of the apps the company has been involved in. They value simplicity and beautiful design, and it shows in their work.

John Casasanta is the founder of tap tap tap. He is a longtime Mac developer and the creator of MacHeist, the number-one Mac software promotion site. Although born a programmer, John's well past his coding prime now and can't be trusted around a compiler these days without the proper supervision.

What inspired tap tap tap to build the Voices app?

Voices was a collaboration between tap tap tap and Taptivate. Taptivate, an iPhone development house, had an early version of Voices and was looking for a publishing partner. They were familiar with our successful app launches and thought we'd be able to make a similar contribution to Voices. In the end, we were very satisfied with the result. Voices quickly became our most popular app, selling over 450,000 copies within its first two months in the App Store.

How did you approach the project?

As with all of the apps tap tap tap has created, we work with people from all around the world. We don't have one office or one central location, so we don't have the limit of having to work with people who live in close proximity to us. As a result, we're able to pool together talented individuals from virtually anywhere.

For Voices, our goal was to try to take a *good* app and make it *great*. This involved assessing any usability issues and reworking many of the graphics in the app. We also redesigned some areas of the app to make them fit with the playful theme. For instance, the playback mechanism was originally a simple progress indicator, but we changed it to have more of a real-world tape deck interface. [FIGURES CS12.1–CS12.3 show how the start screen evolved.]

How have users responded to the app?

They seem to really love it. One of the reasons it's done so well is that it's the kind of app that people are quick to show their friends and family. If you're looking to create a hit app, this viral quality is a major factor that can't be underestimated.

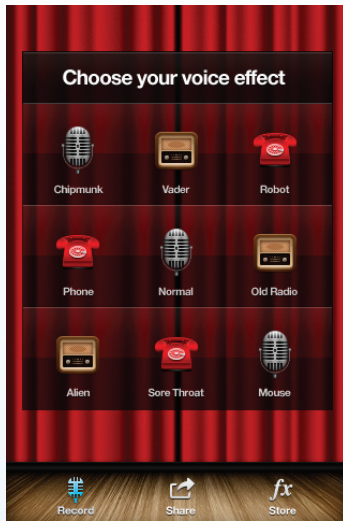


FIGURE CS12.1 Early design of the start screen



FIGURE CS12.2 Later design of the start screen

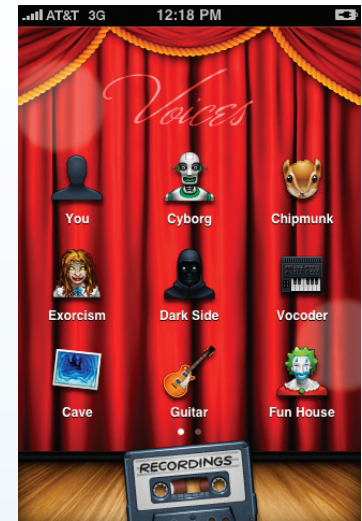


FIGURE CS12.3 Final design of the start screen

We put a lot of time into polishing the UI, and this didn't go unnoticed, thankfully. There are several other voice-changing apps in the App Store, but most of them have hardly made an impression. I strongly feel that it takes a combination of many things to make for a successful app. You need a solid idea, great execution, and strong marketing, among other qualities. Ignore any of these at your peril.

Any advice for developers embarking on similar projects?

One thing I've always stressed over the years is to collaborate with others as much as you can. It's extremely rare that someone has the talents to handle all aspects of creating software, whether it's for iPhone, Mac, Windows, the web, or other platforms.

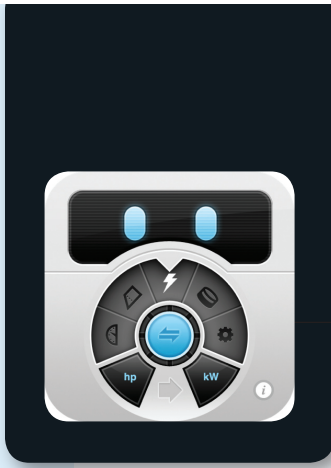
Realize your strengths and weaknesses and choose exceptional people to work with and you'll go *much* further than you'd ever be able to on your own.

What's next for tap tap tap?

We plan to continue along the same track that we've been on since the App Store opened. We care deeply about making the highest-quality apps and providing great user experiences. Our portfolio consists of a combination of both functional apps and fun apps, and it brings us pleasure to develop both types.

Our "next big thing" is our take on the ideal iPhone camera app; the name of the app will be Camera+. We've been working hard on it for the past several months and it's our most ambitious project to date. We also have several other apps in various stages of development. It's been an exciting ride for us since the iPhone came out, and we're really looking forward to seeing what the future brings. ■

(Voices icon and application screenshots courtesy of Taptivate)



CASE STUDY 13

Convertbot

MARK JARDINE is the designer and cofounder of Tapbots, the maker of Weightbot, Convertbot, and Pastebot. Mark has been designing for over 10 years. He loves to draw, take photos, and play on computers. His partner, Paul Haddad, has been developing software for the Mac platform for over 17 years.

What inspired Tapbots to build Convertbot?

Convertbot was based on our first app, Weightbot, which launched in late 2008. Around that time, the movie *WALL-E* came out in theaters. The robots in that movie fascinated me, and they inspired us to create an app with a robot-like aesthetic and interaction. You may notice that Weightbot resembles Eve from the movie. When it was time to embark on our second application, we wanted to build an application that was really simple and evolved the user experience through design. Converters seemed like a good area since they were a popular category with limited innovation.

How did you approach the project?

Before I start sketching, I try to break apps down into their most simple functionality. What are the interaction points that are required for the app to function? For example, Convertbot had four functions: choose a category, choose what to convert from, choose what to convert to, and enter the value. If I know the main points, then the little details are not as important. [FIGURE CS13.1 illustrates some of the initial directions.]

How did the design evolve along the way?

Early on we considered using something similar to the wheel on the iPod. We knew it wouldn't necessarily be the most efficient but that wasn't our goal—we wanted to provide a fun experience that people would enjoy. Once I had the design worked out on paper, I immediately switched to the computer. I always start with flat shapes, then gradually build depth into the designs. Depth was really important—there were many iterations to make the app feel like a real robot [FIGURE CS13.2].

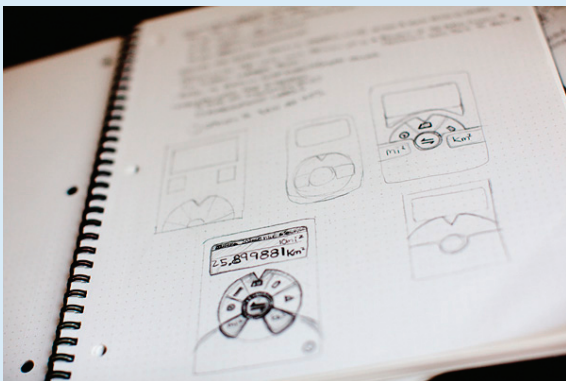


FIGURE CS13.1 Early Convertbot sketches

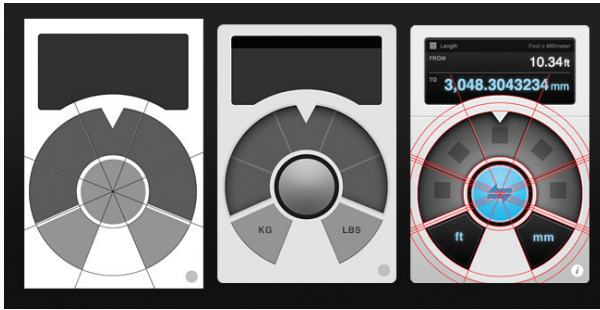


FIGURE CS13.2 Progressive refinement of the Convertbot design

How did you know when the design was done?

It's typically done when Paul and I can look at it and I say, "Okay, this works." Often I'll put a design aside for a few days and then return to it later. With a fresh set of eyes, I'll notice things I didn't see before. For example, the LED screen in an earlier sketch felt dated [FIGURE CS13.3]—the style was originally from Weightbot. After not looking at it for a few days, I returned to the design and started experimenting with the border. Several iterations later, I removed the thick border and added a much narrower one. This subtle change made the design fresh and gave the LED more room [FIGURE CS13.4].

How did the Convertbot decisions affect Weightbot?

Right after Convertbot was done, we went back and changed the Weightbot borders. Whenever we come out with a new application, we always go back and evolve our other app designs. Basically, the more design work you do, the better you get at it. With every new application we feel our designs improve.

How did you make the Convertbot video demo?

We like to do everything in-house—it's cost-effective and always a good learning experience. For the video demo, I had a decent camera, microphone, tripod, and Final Cut Express. When I took the first shot, the camera was too close to the phone so you could see scan lines. To prevent this from happening, I had to place the camera farther

away and zoom in to the iPhone. The phone was in a dock and taped to the table so it wouldn't shift during the demo. Afterward I added sound and a voice-over via Final Cut Express.

What's next for Tapbots?

We are always trying to push our designs. Our latest app, Pastebot, has user-generated content which is very different from our other apps. When we designed this app, we incorporated much more of Apple's guidelines since they were the most effective way to achieve our design goals.

What kind of advice would you give to designers who want to create Immersive apps but are afraid to diverge from the HIG?

First, the designer should be capable of creating something that's visually compelling. Second, diverging from the *Human Interface Guidelines* just for the sake of it isn't necessarily going to make your app better. The bottom line is you have to consider your core audience and your goals. We did something completely different since we had specific design goals and a concept behind them. Our applications are part of our story. ■



FIGURE CS13.3 Convertbot with thick border



FIGURE CS13.4 Convertbot with narrow border

(Convertbot icon, photographs, and application screenshots courtesy of Mark Jardine)