

Mobile Design Patterns

A large, bold, orange letter 'M' is positioned in the bottom right corner of the cover. It is set against a white circular background that is part of a series of overlapping circles in shades of orange and white. The entire cover has a network of white lines connecting circular nodes, creating a geometric pattern.

Imprint

Copyright 2012 Smashing Media GmbH, Freiburg, Germany

Version 1: September 2012

ISBN: 978-3-943075-43-4

Cover Design: Ricardo Gimenes

PR & Press: Stephan Poppe

eBook Strategy: Talita Telma Stöckle

Technical Editing: Talita Telma Stöckle, Andrew Rogerson

Idea & Concept: Smashing Media GmbH

ABOUT SMASHING MAGAZINE

[Smashing Magazine](#) is an online magazine dedicated to Web designers and developers worldwide. Its rigorous quality control and thorough editorial work has gathered a devoted community exceeding half a million subscribers, followers and fans. Each and every published article is carefully prepared, edited, reviewed and curated according to the high quality standards set in Smashing Magazine's own publishing policy. Smashing Magazine publishes articles on a daily basis with topics ranging from business, visual design, typography, front-end as well as back-end development, all the way to usability and user experience design. The magazine is — and always has been — a professional and independent online publication neither controlled nor influenced by any third parties, delivering content in the best interest of its readers. These guidelines are continually revised and updated to assure that the quality of the published content is never compromised.

ABOUT SMASHING MEDIA GMBH

[Smashing Media GmbH](#) is one of the world's leading online publishing companies in the field of Web design. Founded in 2009 by Sven Lennartz and Vitaly Friedman, the company's headquarters is situated in southern Germany, in the sunny city of Freiburg im Breisgau. Smashing Media's lead publication, Smashing Magazine, has gained worldwide attention since its emergence back in 2006, and is supported by the vast, global Smashing community and readership. Smashing Magazine had proven to be a trustworthy online source containing high quality articles on progressive design and coding techniques as well as recent developments in the Web design industry.

About this eBook

Since the appearance of the smartphone and tablet computers, we are experiencing a boom in Mobile Design. And, as it happens with the development of new gadgets and technologies, some trends and patterns are already established. With this eBook "Mobile Design Patterns", you will learn the most important guidelines and strategies for Mobile Design, including solutions for forms, tap-ahead patterns and UX approaches.

Table of Contents

[The Elements Of The Mobile User Experience](#)

[Picking A Mobile Support Strategy For Your Website](#)

[Essential Design Patterns For Mobile Banking](#)

[Seven Guidelines For Designing High-Performance Mobile User Experiences](#)

[Useful Design Tips For Your iPad App](#)

[A User-Centered Approach To Web Design For Mobile Devices](#)

[Mobile Auto-Suggest on Steroids: Tap-Ahead Design Pattern](#)

The Elements Of The Mobile User Experience

Lyndon Cerejo

Mobile users and mobile usage are growing. With [more users doing more on mobile](#), the spotlight is on how to improve the individual elements that together create the mobile user experience.

The mobile user experience encompasses the user's perceptions and feelings before, during and after their interaction with your mobile presence — be it through a browser or an app — using a mobile device that could lie anywhere on the continuum from low-end feature phone to high-definition tablet.

Creating mobile user experiences that delight users forces us to rethink a lot of what we have taken for granted so far with desktop design. It is complicated in part by mobile-specific considerations that go hand in hand with small screens, wide variations in device features, constraints in usage and connectivity, and the hard-to-identify-but-ever-changing mobile context.

Dissecting the mobile user experience into its key components gives us a conceptual framework for building and evaluating good mobile experiences, within the context of a [user-centered approach to designing for mobile](#).

These components shape the mobile user experience — including functionality, context, user input, content and marketing, among others.

The relevance of these elements will change depending on the type of device (feature phone versus smartphone versus tablet) and the

presentation interface (app versus Web). This article briefly describes each of these elements and elaborates on each with selected guidelines.

FUNCTIONALITY

This has to do with tools and features that enable users to complete tasks and achieve their goals.

Guidelines

- Prioritize and present core features from other channels that have especial relevance in a mobile environment. For an airline, this includes flight statuses and flight check-ins. For cosmetic chain Sephora, it includes supporting in-store shopping via easy access to product reviews on mobile devices.
- Offer relevant mobile-only functionality (like barcode scanning and image recognition), and enhance functionality using the capabilities of mobile devices where possible to engage and delight users. Old Navy's app serves up surprise games or savings when users snap the logo in a store.
- Ensure that fundamental features and content are optimized for mobile. For example, make sure the store locator shows the nearest stores based on the device's location, and make the phone numbers click-to-call.
- Include features that are relevant to the business category. For retail websites and apps, this would include product search, order status and shopping cart.

- Offer key capabilities across all channels. Users who sign in should see their personalized settings, irrespective of the device or channel being used. If certain functionality is not offered on mobile, then direct users to the appropriate channel, as Triplt does to set up a personal network.

INFORMATION ARCHITECTURE

This has to do with arranging the functionality and content into a logical structure to help users find information and complete tasks. This includes navigation, search and labeling.

Guidelines

- Present links to the main features and content on the landing page, prioritized according to the user's needs. [Mobile Design Pattern Gallery](#) has examples of primary and secondary navigation patterns for mobile, many of which are vertical instead of horizontal as on desktop websites.
- Enable mobile users to navigate to the most important content and functionality in as few taps or key presses as possible. Navigation optimized for small screens is usually broad and shallow instead of deep. While [three clicks \(or taps\) is not the magic number](#), users need to be able to recognize that each tap is helping them complete their task. Every additional level also means more taps, more waiting for a page to load and more bandwidth consumed.
- Address the navigation needs of both touchscreen and non-touchscreen users. When designing for touch, make sure the tap size of the navigation item is at least 30 pixels wide or tall. Provide keypad shortcuts for feature phones, so that users can enter, say, a number (0 to 9) to quickly access a link:

Cater to feature phone users, as CNN does with access keys (left), not

as Delta does by making the first action to be nine key presses downs (middle and right).

- Provide navigational cues to let users know where they are, how to get back and how to jump back to the start. Mobile breadcrumbs are often implemented by replacing the “Back” button with a label showing users the section or category that they came from. For mobile websites, use standard conventions, such as a home icon that links back to the start screen, especially when navigation is not repeated on every screen.
- Use concise, clear, consistent and descriptive labels for navigation items and links. While always a good practice, it becomes even more important on tiny mobile devices.

CONTENT

Otherwise known as “the stuff on your website” (as Lou Rosenfeld and Peter Morville refer to it in *Information Architecture for the World Wide Web*), content is the various types of material in different formats, such as text, images and video, that provide information to the user.

Guidelines

- Present an appropriate and balanced mix of content to users (product information, social content, instructional and support content, marketing content).
- Use multimedia when it supports the user’s tasks in a mobile context, adds value to the content or supports the goals of the website. Most of the time, multimedia content is best provided when the user is looking for distraction or entertainment (such as news or funny clips) or when it has instructional value (for example, how to use an app or new feature).

- Always give the user control over multimedia content by not auto-starting video or sound, by allowing the user to skip or stop multimedia content and by being mindful of the bandwidth it takes up.
- Ensure that content is mobile appropriate. Just as we had chunking guidelines when going from print to Web, copy should be written for shorter attention spans on mobile devices. Optimize images and media for the device; this means scaling down for smaller devices and making sure images are [sharp enough for the new iPad](#).
- Ensure that primary content is presented in a format supported on the target device. Even now, websites such as Volkswagen's ask iOS users to download Flash.

DESIGN

This has to do with the visual presentation and interactive experience of mobile, including graphic design, branding and layout.

Guidelines

- Remember the sayings “Mobilize, don’t miniaturize” (popularized by Barbara Ballard) and “Don’t shrink, rethink” (of Nokia). Both make the point that mobile design should not just rehash the desktop design.
- Design for glanceability and quick scanning. Glanceability refers to how quickly and easily the visual design conveys information.
- Maintain visual consistency with other touchpoints and experiences (mobile, app, Web, print and real world) through the use of color, typography and personality. Identifying Amazon in the stack below is easy even though the brand name is not visible.

- Guide users from the initial and most prominent element of the design to other elements to help them complete their tasks. This is known as visual flow. A good design brings together visual elements as well as information architecture, content and functionality to convey the brand's identity and guide the user.
- Consider both portrait and landscape orientations in the design process. Devices increasingly support multiple orientations and automatically adjust to match their physical orientation. Maintain the user's location on the page when they change orientation. Indicate additional or different functionality in the new orientation if applicable, as shown by ING:

USER INPUT

This has to do with the effort required to enter data, which should be minimized on mobile devices and not require the use of both hands.

Guidelines

- Limit input to essential fields. Or, as Luke Wroblewski says in his book *Mobile First*, “When it comes to mobile forms, be brutally efficient and trim, trim, trim.” Limit registration forms to the minimum fields required, and use shorter alternatives where possible, such as a ZIP code instead of city and state. My favorite offender of this guideline is Volkswagen's form to schedule a test drive; the mobile form has more required fields than the desktop version (the extra fields are highlighted below):
- Display default values wherever possible. This could be the last item selected by the user (such as an airport or train station) or the most frequently selected item (such as today's date when checking a flight's status):

- Offer alternate input mechanisms based on the device's capabilities where possible. Apps take advantage of quite a few input mechanisms built into devices, including motion, camera, gyroscope and voice, but mobile websites are just starting to use some of these features, particularly geolocation.
- Use the appropriate input mechanism and display the appropriate touch keyboard to save users from having to navigate their keyboard screens to enter data. Keep in mind that inputting data is more tedious on feature phones that have only a numeric keypad. For non-sensitive applications, allow users to stay signed in on their mobile device; and save information such as email address and user name because mobile phones tend to be personal devices, unlike tablets, which tend to be shared between multiple people.
- Consider offering auto-completion, spellcheck suggestions and prediction technology to reduce the effort required to input data and to reduce errors — with the ability to revert as needed. Disable features such as CAPTCHA where not appropriate.

MOBILE CONTEXT

A mobile device can be used at anytime, anywhere. The mobile context is about the environment and circumstances of usage — anything that affects the interaction between the user and the interface, which is especially important for mobile because the context can change constantly and rapidly. While we often focus on distractions, multitasking, motion, low lighting conditions and poor connectivity, it also includes the other extreme — think using a tablet in a relaxed setting over a fast Wi-Fi connection.

[“The Context of Mobile Interaction,”](#) Nadav Savio

Guidelines

- Use device features and capabilities to anticipate and support the user's context of use. The iCookbook app allows users to walk through a recipe using voice commands — a nice feature when your hands are covered in batter!
- Accommodate for changes in context based on the time of day and when the user is using the app. The Navfree GPS app automatically switches from day to night mode, showing low-glare maps for safer nighttime driving.
- Use location to identify where the user is and to display relevant nearby content and offers. A Google search for “movies” on a mobile device brings up movies playing nearby and that day's showtimes, with links to buy tickets online if available.
- Leverage information that the user has provided, and respect their preferences and settings. After the first leg of a multi-leg flight, Triplt showed me the flight and gate information for my next flight, as well as how much time I had to kill. United's app did no such thing, even though it knew much more about me. It could have shown me how to get from my current plane to the connecting flight and highlighted the location of the United Club along the way, where I could comfortably spend my two-hour wait, since it knew I was a member.
- Default to the user experience most appropriate for the device (i.e. a mobile experience for small screens, and perhaps a desktop-like experience for tablets), but give users the option to have enhanced features. A big discussion on how to present this to the user recently took place, with [Jakob Nielsen recommending a separate mobile website](#) and [Josh Clark arguing instead for a responsive design](#); yet others [believe that Nielsen and Clark are both wrong](#).

USABILITY

This is the overall measure of how well the information architecture, design, content and other elements work together to enable users to accomplish their goals.

Guidelines

- Make it clear to the user what can be selected, tapped or swiped (this is known as affordance), especially on touchscreen devices. One of the big findings of [Nielsen Norman Group's usability studies of the iPad](#) was that users didn't know what was touchable or tappable. Another issue was swipe ambiguity: when the same swipe gesture means different things in different areas of a screen. Ensure that touchability is clear and that items such as links, icons and buttons are visibly tappable.
- For touchscreen devices, ensure that touch targets are appropriately sized and well spaced to avoid selection errors. Also, place touch targets in the appropriate screen zones; for example, put destructive actions such as those for deletion in the "Reach" zone, as shown by Luke Wroblewski in his book *Mobile First*:
- Follow conventions and patterns to reduce the learning curve for users and to make the mobile experience more intuitive. Dedicated apps should follow platform-specific standards and guidelines. A comprehensive collection of links to official UI and UX guidelines is available in the article "[UI Guidelines for Mobile and Tablet Web App Design](#)" on Breaking the Mobile Web.
- Ensure usability in variable conditions, including for daylight glare and changed angle of viewing and orientation, by paying attention to design elements like contrast, color, typography and font size.

- Do not rely on technology that is not universally supported by your audience's devices, including Java, JavaScript, cookies, Flash, frames, pop-ups and auto-refreshing. When opening new windows or transitioning from an app to the browser, warn users to avoid overwriting already open tabs.

TRUSTWORTHINESS

This relates to the level of confidence, trust and comfort that users feel when using a mobile website or app. According to a [2011 study by Truste and Harris Interactive](#), privacy and security are the top two concerns among smartphone users:

Guidelines

- Do not collect or use personal information (such as location and contact list) from mobile devices without the explicit permission of the user. The first few months of this year have seen numerous reports of apps secretly copying smartphone address books, with [watchdogs up in arms](#) and [users retaliating](#).
- Make it easy for users to control how their personal information is shared in a mobile app by asking before collecting their location data and by allowing them to opt out of targeted advertising.
- Clearly state your business practices (including for privacy, security and returns), and present them contextually (such as by displaying links to your privacy and security policies on the registration screen). The policies themselves should be accessible in a secondary section of the mobile user experience (such as the footer or a “More” tab). Reinforce credibility by displaying trusted badges, especially when users need to trust you with their personal or financial information.

- Present policies appropriately on mobile devices by offering a concise summary and an option to email the entire policy. Privacy and security policies tend to be notoriously long and full of boring legalese that users often blindly click through to continue what they really want to do, so make it easy for users who are interested in the fine print.
- Don't break the user's workflow when displaying legalese. [Take them back to where they were](#) before being interrupted, instead of making them start all over.

FEEDBACK

This has to do with the methods for attracting the user's attention and displaying important information.

Guidelines

- Minimize the number of alerts the app displays, and ensure that each alert offers critical information and useful choices. For a smile, look at Chris Crutchfield's [video on notification and alert overload](#).
- Keep alerts brief and clear, explaining what caused the alert and what the user can do, along with clearly labeled buttons.
- Notifications should be brief and informative, not interfere with anything the user is doing, and be easy to act on or dismiss.
- Provide feedback and confirmation on screen without disrupting the user's workflow.

- If your app displays badges and status bar notifications, keep the badges updated and clear them only when the user has attended to the new information. Chase clears the notifications badge for its mobile app the moment the user visits the notification section, even before the user has seen which of their multiple accounts triggered the badge, forcing them to hunt through each account to see what triggered it.

HELP

This relates to the options, products and services that are available to assist the user in using the website or app.

Guidelines

- Make it easy for users to access help and support options. Users commonly look for help in the footer of a mobile website and in the toolbar or tab bar of an app.
- Offer multiple ways to get support, including options relevant in a mobile context, such as self-serve FAQs, live support via click-to-call, and near-real-time Direct Message tweets. Two financial service companies that actively offer support via Twitter are American Express and Citibank.
- Present a quick introduction and short tutorial on using the app when it first launches, with options for the user to skip and view later.
- When introducing new or unique functionality (such as when check depositing via mobile apps was first introduced), offer contextual help and tips to guide users the first time, and as a refresher for infrequently used functionality.

- Offer help videos when appropriate, but allow the user to start, pause, stop and control the volume as they wish, and keep in mind the multimedia guidelines mentioned in the “Content” section above.

SOCIAL

This relates to content and features that create a sense of social participation, that enable user interaction and that facilitate sharing on established social networks.

Guidelines

- Create and maintain a presence on social networks (for example, a Facebook page) and local services (for example, a profile page on services such as Google Places, Bing Business Portal and Yahoo Local). These will be highlighted in search results and on location-based social networking services. In addition to your business’ name, include your physical address, phone number, URL and hours of operation.
- Incorporate your social presence and activity into your website’s mobile experience by showing your recent activity and offering an easy way to follow or like you on these networks.
- Integrate social networking features into your website’s mobile experience to make it easy for users to connect with their own social networks. This could be as simple as [using APIs](#) to enable social sharing, bookmarking, tagging, liking and commenting.
- Invite users to generate content featuring your brand, product or service from their mobile device, offering some incentive in return. For example, the burger chain Red Robin could invite the user to share a picture of their child reading a school book at one of its locations to get a free milkshake.

- Provide mobile offers that can be shared and go viral. American Express currently offers savings and discounts to users who [sync their profiles](#) on networks such as Facebook, Twitter and Foursquare to their credit card.
- Apps that rely on social contributions from users should look at ways to seed content in a way that is useful and, eventually, self-sustaining. For example, the My TSA app has a user-contributed feature that shows the wait times at security checkpoints, but it often shows outdated information, even though airport staff post physical signs of wait times at some airports.

MARKETING

This has to do with the methods by which a user finds a website or app and the factors that encourage repeated usage.

Guidelines

- Ensure findability by optimizing for mobile search and discovery, such as by keeping URLs short. If you have a separate mobile website, follow URL naming conventions (**m.site.com** or **mobile.site.com**). In mobile search results, provide quick access to location-based content (e.g. directions from one's current location) and device-formatted options (e.g. click to call).

Mobile-formatted information is optimized for UPS (left), but partially missing for FedEx (right).

- “Quick response” (QR) codes should lead to a mobile-optimized landing page, instead of a traditional page that requires zooming or, worse still, to the website’s home page, from where the user has to hunt for information. As a side note, [QR codes painted on buildings](#) should be big and clear enough to be recognized and deciphered by mobile devices.
- Email campaigns should include a link to view the message in a mobile-friendly format, which itself links to the relevant offer page formatted for mobile — unlike CVS/pharmacy, which takes users to its mobile home page.
- Promote your app in other channels where possible (TV, print and in-store advertising), and offer incentives to download and use the app, usually in the form of discounts and savings. If your app has a price tag, attract users to buy it in an overcrowded market by offering a limited-time promotional price. Another option is to promote the app through the Free App A Day marketplace.
- Prompt users to rate and review your app or to share it on social networks after they have used it, but give them the option to postpone or stop these prompts. This will not only generate word of mouth, but give you insight into what users like and don’t like about the app. [“Taking Control of Your Reviews”](#) by smalltech discusses the strategy of encouraging happy customers to post reviews and unhappy customers to email you feedback.

Conclusion

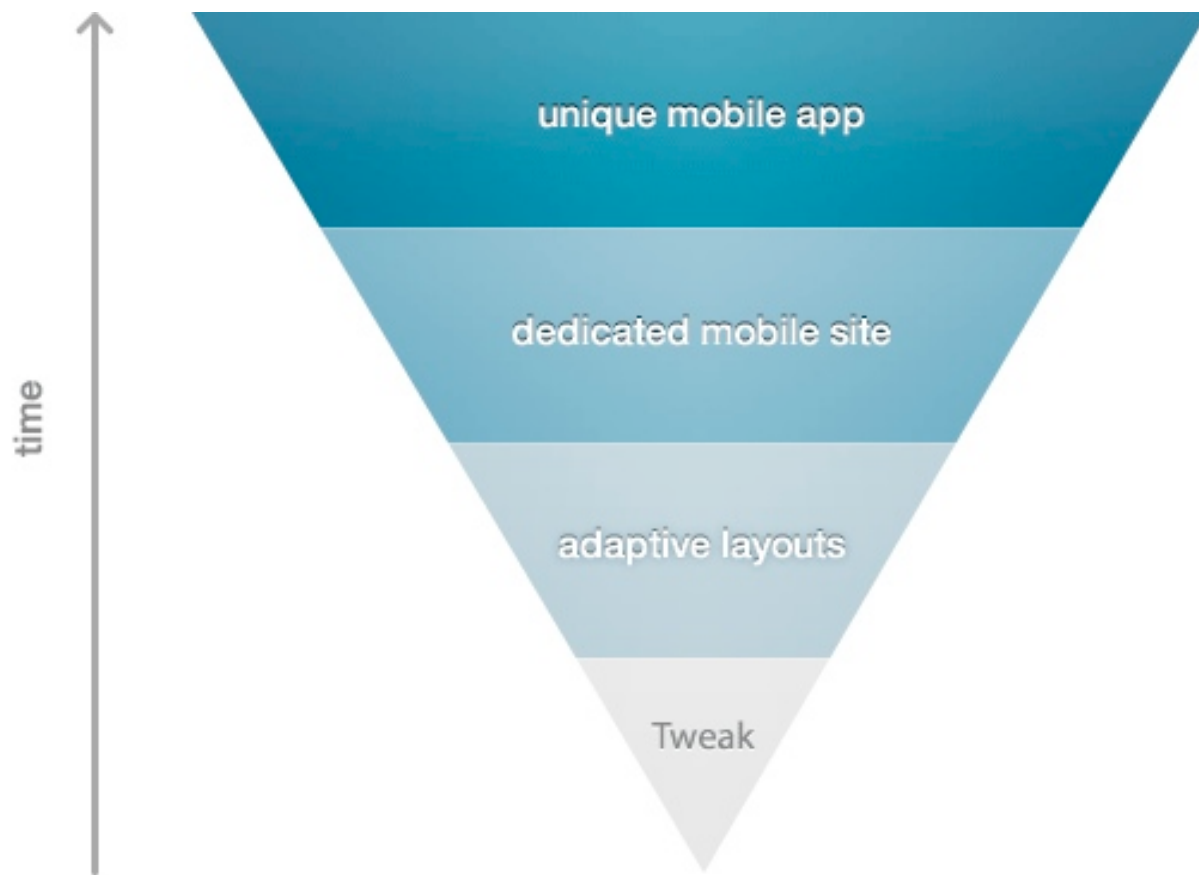
Mobile user experience is still a developing field, and opportunities for improvement continue to emerge. We've presented an overview of the key elements of the mobile user experience, along with some guidelines to get started in each. Focusing on these individual elements will help us create great overall mobile user experiences for our users.

Picking A Mobile Support Strategy For Your Website

Matt Lawson

The number of people browsing the Web from a mobile device has [more than tripled since 2009](#), and it is sure to continue growing, with browser platforms such as iOS and Android offering mobile browser support that is almost identical to what we have come to expect from a desktop experience. As the mobile consumer market continues to grow, so will the aspirations of individuals and companies who look to embrace what the mobile Web has to offer.

With this in mind, many website owners have begun to develop a strategy for providing information and services to their mobile visitors. However, mobile strategies can vary massively from website to website, depending on what the company wants to offer visitors. For example, eBay's strategy will be very different from an individual's strategy for a portfolio website, which might simply be to improve readability for those viewing on a mobile device.



Increasing mobile support could lead to a better experience, but at what cost?

So, as website owners define the level of support they aim to provide, a scale of support for mobile devices emerges. Picking where on the scale your website should sit can be quite tricky; each level of support is not without its pros and cons. Let's take a look at some of the more common approaches:

Approach A: Tweak What You Have

The most basic and, thus, quickest option is to do only what is required to get the website to work on mobile devices. I use the word “work” loosely here because it can be very subjective, but the main goal is to ensure that the website displays and functions properly on mobile devices and perhaps similarly to the desktop experience.

Sure, delivering a desktop experience on a mobile device is not ideal by any stretch of the imagination, but this option simply offers the minimum required to get the website to function and display OK. With modern mobile devices offering good CSS support and zooming functionality, visitors should at least be able to access the information they need.

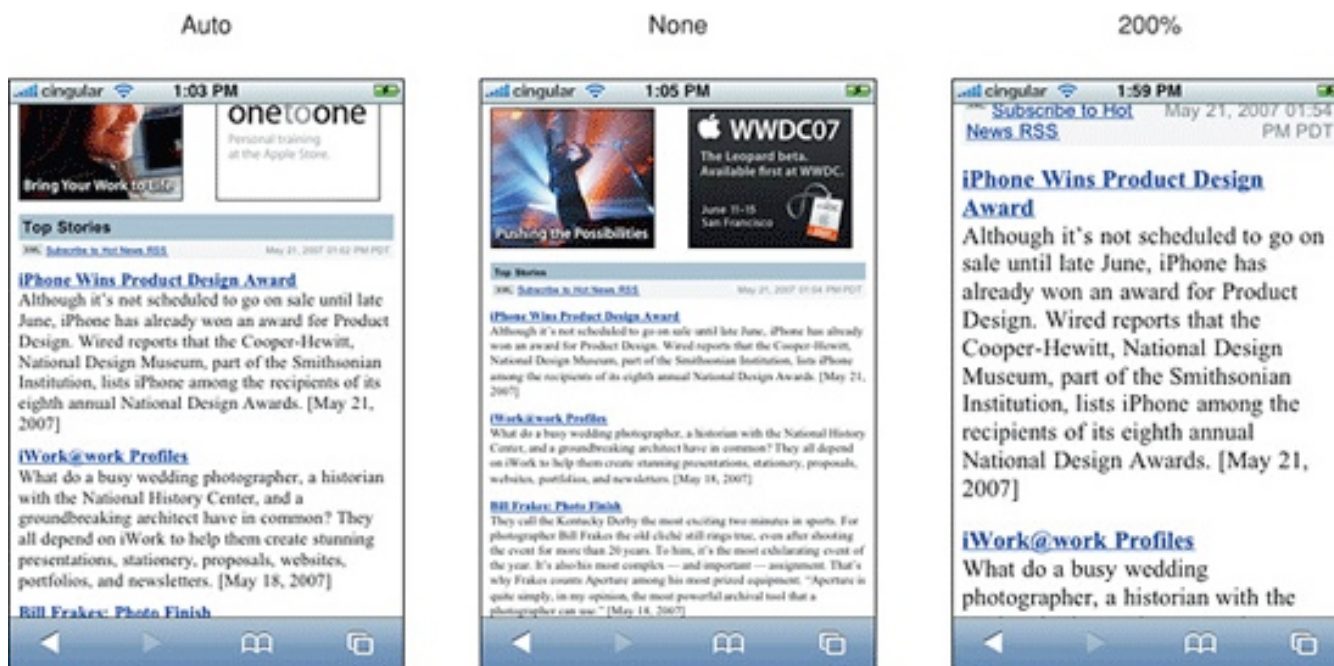
HOW TO IMPLEMENT THIS APPROACH?

Simple tweaks could include adjusting the viewport and text size, which will affect the way the website displays on a mobile device. The default viewport dimensions should work well for most layouts, but we can make adjustments using the `meta` element:

```
<meta name="viewport" content="width=device-width" />
```

Text size can also be adjusted for some mobile devices using the CSS **text-size-adjust** property which specifies a size adjustment for displaying text content:

```
html {  
  -webkit-text-size-adjust: auto; /* Automatically adjusted for  
  Safari on iPhone. */  
  -ms-text-size-adjust: auto; }
```



Different text-size-adjust values demonstrated on the iPhone.

More information on the **text-size-adjust** property is available in the [Safari Developer Library](#). With a small number of tweaks, you should be able to optimize your website to appear as usable as the desktop experience.

Be careful when making any adjustments to the CSS for mobile visitors: you do not want desktop users ending up with a 200% font size by default! If you think this might happen or you want to further improve the experience, consider putting the CSS in a separate file:

```
<link rel="stylesheet" href="..." media="handheld, only screen and (max-device width: 480px)" />
```

Pros:

- Quick to implement;
- Minimal work required to replicate the desktop design;
- Strong brand identification with basic consideration for mobile visitors.

Cons:

- Mobile users could suffer from a poor experience;
- Slow due to users downloading styles and large assets;
- Content and navigation path are not optimized for mobile visitors.

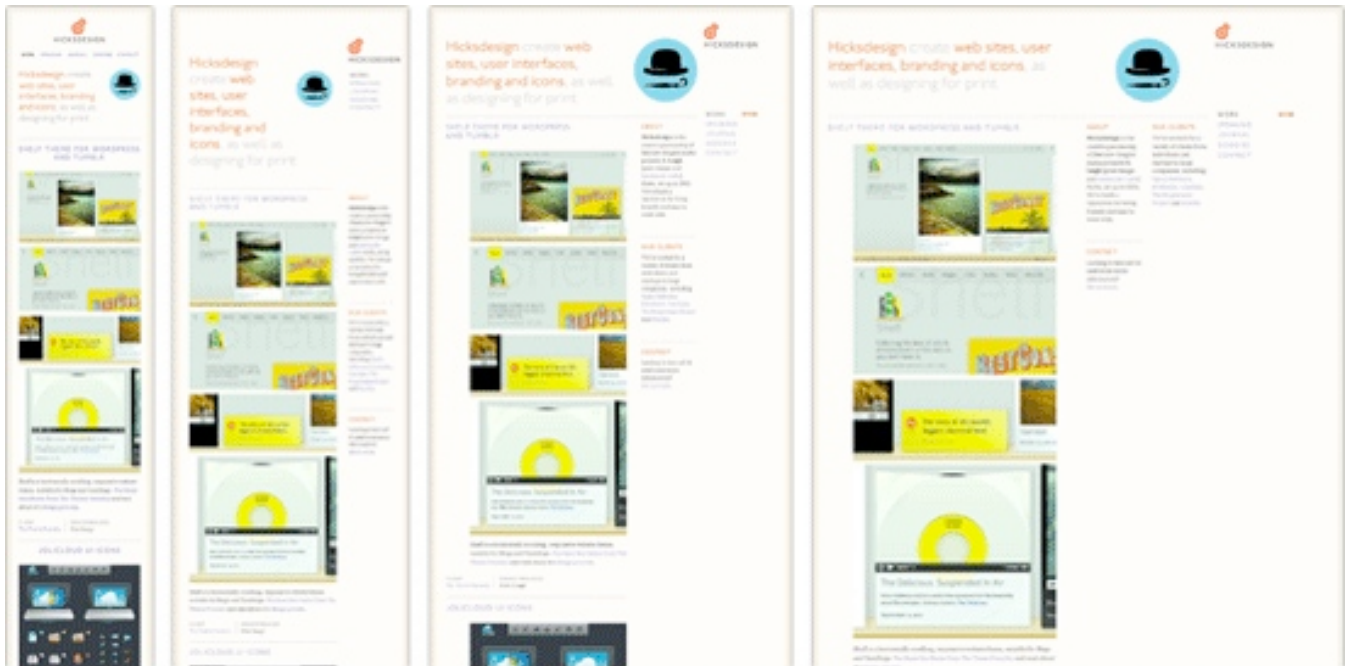
Approach B: Adaptive Layout (Media Queries)

Media-dependent styling has been around for a long time; you will almost certainly have used “media types” before:

```
<link rel="stylesheet" href="..." media="print" />
```

Media queries, on the other hand, have really started to gain popularity since browser vendors began to support the [W3C's CSS3 “Media Queries” specification](#).

Most modern browsers, including mobile ones, should now be able to query such things as width, height, device width and height, orientation and more. This has led to more people using media queries to provide responsive designs to their visitors:



Hicksdesign demonstrates adaptive layouts using media queries.

For older browsers, including Internet Explorer 6 to 8, several solutions are available that add some level of support for media queries, such as [Respond.js](#) by Scott Jehl.

HOW TO IMPLEMENT THIS APPROACH?

We can target specific resolutions and device sizes. For example, we could target mobile devices with a maximum device width of 480 pixels, such as the iPhone:

```
<link rel="stylesheet" media="only screen and (max-device-width: 480px)" href="mobile.css" />
```

Or we could put the same media query in our CSS file:

```
@media only screen and (max-device-width: 480px) {  
    // insert styling here...  
}
```

Adaptive layouts need to work with the content already available on your website. This means that the source order and mark-up can play a vital role in providing a logical order to content when linearized for narrow layouts. You will also need to take into account that images will need to scale to fit as their containing elements adapt to different layouts. One way to achieve this is to specify a maximum width:

```
img { max-width: 100%; }
```

You could consider providing the mobile experience as the default and the desktop experience through media queries, an idea discussed by both [Luke Wroblewski](#) and [Peter Gasston](#). Combining this approach with something like [Adapt.js](#) or [320 and up](#) could improve performance for mobile visitors.

However, making the mobile experience the default isn't without its own problems. Always consider your audience, and review visitor data before finalizing your approach.

Pros:

- Quick to develop, especially when considered from the start;

- Cheap to produce because minimal additional design is required;
- Can result in improved readability and experience for mobile visitors.

Cons:

- Older mobile and desktop browsers, including Internet Explorer 8, do not natively support media queries;
- Visitors could face a short learning curve if the navigation and layout are altered;
- Rendering could potentially be slower as images and non-critical content in the HTML are being downloaded.

Both approach A and approach B beautifully embrace the “[One Web](#)” philosophy which sees the Web as one universal medium that should adjust itself to the different environment of its users. Using mobile tweaks and media queries can help to keep the website a standalone, universal entity optimized for both mobile and desktop user experiences. As Jeremy Keith writes in his article,

“Recent developments in areas like [performance](#) and [responsive design](#) means that we can realistically pursue that vision of serving up content at a URL to everyone to the best ability of their device. At the same time, the opposite approach—creating multiple, tailored URLs—is currently a popular technique.”

— Jeremy Keith, [One Web](#)

We will discuss the latter approach in the next sections of this article.

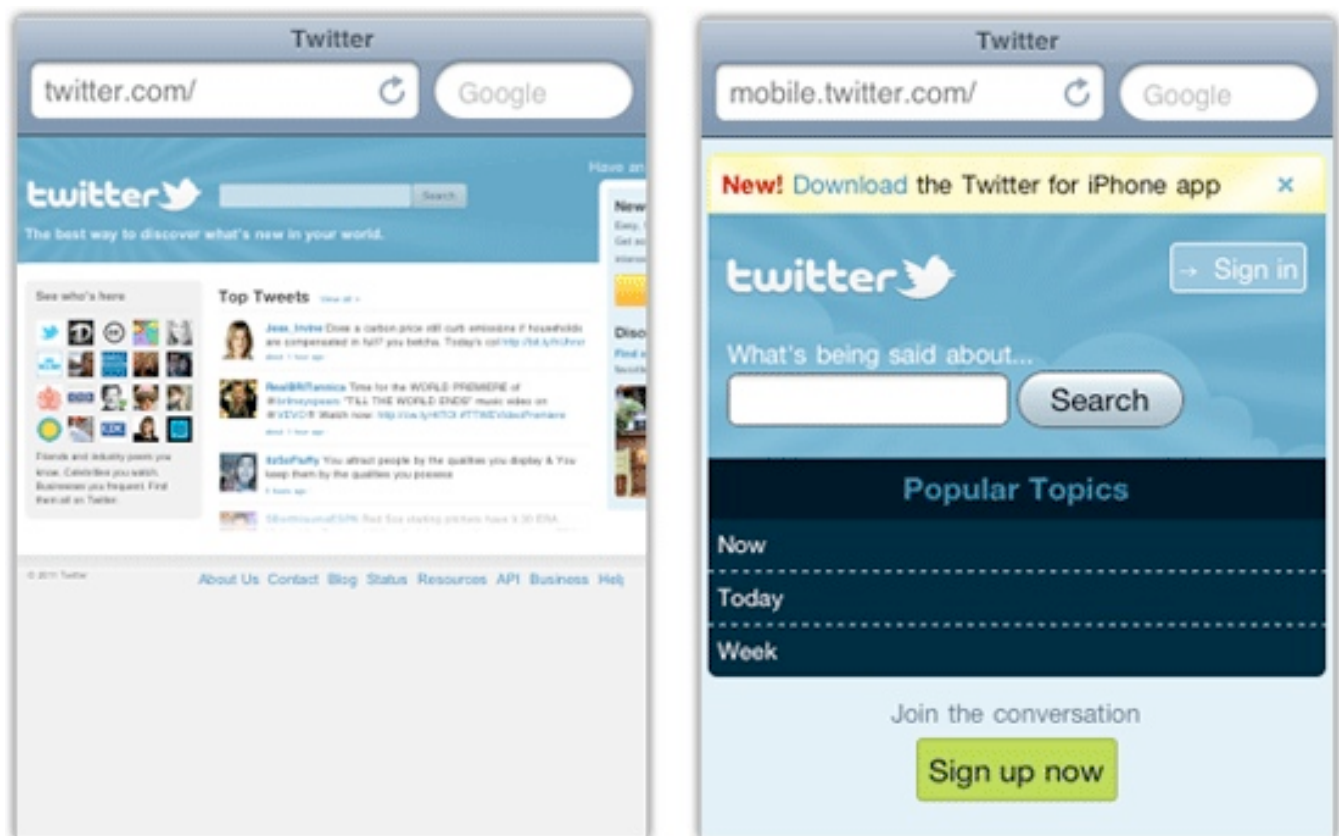
Approach C: A Dedicated Mobile Website

A website dedicated to mobile users aims to deliver an optimized, and often very different, experience to visitors. These micro or mobile websites can take on a life of their own and often require a lot of research and analysis in order to prioritize and deliver the most important content to users.

Mobile websites from the likes of eBay and Amazon show a very different strategy than their desktop equivalents because screen space and file sizes are at a premium.

HOW TO IMPLEMENT THIS APPROACH?

A dedicated mobile website will normally reside on its own domain or sub-domain, such as mobile.twitter.com:



The main Twitter home page and the mobile version on an iPhone.

Redirecting mobile traffic to a dedicated website ensures that visitors arrive in the right place. But if you do this, provide a link to allow visitors to access to the full version! Also make sure that mobile users are redirected to the correct page when deep linking from another source.

Assets such as images should be kept to a minimum. And popular content, common tasks and key navigational paths should be highlighted to give users exactly what they want. More often than not, there is no room for advertisements in mobile versions.

Despite the extra work, the result can be a faster, more streamlined experience that puts the most important features and content at the user's fingertips.

Pros:

- Greatly improved performance;
- Optimized paths make it easy and fast for users;
- Enhances your support of and appeal to growing mobile consumer market.

Cons:

- Relatively expensive to build and maintain;
- Time-consuming because assets must be optimized and content prioritized;
- Higher learning curve if the layout and content are very different from the desktop versions.

Approach X: Native Apps

Finally, another option to consider is a native app. Apps can be the ultimate in an optimized, streamlined journey for visitors, and they often have native controls. Several properties, such as eBay, Twitter and Amazon, have clear user goals and have therefore invested time and effort into creating native apps that provide the best possible experience on a wide range of devices.

HOW TO IMPLEMENT THIS APPROACH?

A native app should provide the best possible experience for users on the go, while taking full advantage of device-specific features and controls. This approach is very different from the others described, and the project could be considered “ad hoc” development, correlating more closely to the user’s goals than the content or features on your website.

If this appeals to you, consider using an SDK, such as the ones available from [PhoneGap](#) and [Appcelerator](#). These SDKs enable developers with a Web background to create applications and tap into native APIs that are not always available in the browser. Native app development can be quite bespoke and is sometimes undertaken parallel to the main website.

Facebook, which offers a native app, is a good example of how a integrated approach can ensure that content is accessible through full, mobile, touch and app versions, each optimized for the best possible experience.



Facebook can be accessed through full, mobile, touch and native app interfaces.

Pros:

- Streamlined journeys;
- Device controls are native and optimized for platform in terms of speed and performance;
- Incredibly lightweight, with minimal bandwidth usage.

Cons:

- Requires bespoke development;
- Creating and maintaining apps for a range of devices is time-consuming;
- Third-party approval is required before the app is available in stores.

Which Approach Is Right For You?

Approach A and Approach B offer varying levels of support and often could be considered as a “quick win” strategy. Consider them if you want to improve the experience and optimize readability for mobile visitors at minimal cost.

A strongly related design strategy would be to start with a mobile layout of the website first, having a strong focus on content and navigation and then extend the mobile experience to larger desktop experiences. You can find some interesting workflow techniques for this strategy in Luke Wroblewski’s conference notes of Ethan Marcotte’s talk [The Responsive Designer’s Workflow](#). For instance, you might need to consider using server-side media queries as well as fluid images in the development process.

Approach C requires considerably more time to develop and maintain but results in a faster, more streamlined website for task-oriented visitors. Approach X requires significantly more time to develop and approval from third-party app stores. But it might be ideal if your brand has many mobile users who expect a flawless experience. The main problem with these two approaches is that they aren’t scalable as new mobile devices might require new versions of the websites which increases costs and makes maintenance more difficult.

Ultimately, your approach should be guided by your content, objectives and visitors. What might work in theory doesn’t necessarily work in practice. A bit of digging in your analytics might show that a large proportion of visitors are on mobile devices, and so the extra time spent improving their experience would be worth it. Once you have all of the data, you can make an informed decision about which approach will benefit you — and more importantly — your visitors.

Essential Design Patterns For Mobile Banking

Greg Nudelman

Despite a great deal of mobile innovation, many creators of financial apps still copy their interface patterns from the desktop Web, even though these patterns are not as well suited to the mobile space. Small screens, custom controls, divided attention and fat fingers demand different thinking when designing for mobile.

I previously covered mobile wallet UX considerations in my article [“Ultimate Guide to Designing NFC Mobile Apps You Won’t be Ashamed Of.”](#) In this article, we will look specifically at simple mobile transfers of funds from checking to savings accounts, taking what works on the Web and converting it into authentically mobile flows using simple, effective design patterns. Similar analyses and design strategies can be applied to many other areas that involve complex forms, such as mobile e-commerce checkout and social network registration.

We will not name any companies in this article. That is deliberate. If you are looking for company bashing, you won’t find it here. But if you want to know how to make mobile banking work, read on.

Let’s begin with the basic building block: selecting an account. It can be accomplished in two primary ways: via a “picker” (called “spinner” in Android) and via a dedicated selection page (also called “table view”).

Picker/Spinner

For system interactions, many app and mobile website designers start by looking at the desktop Web interface pattern: a form with drop-down menus. Here is a common pattern for me-to-me transfers (i.e. transfers between two of your own accounts, such as checking and savings):

From Account:

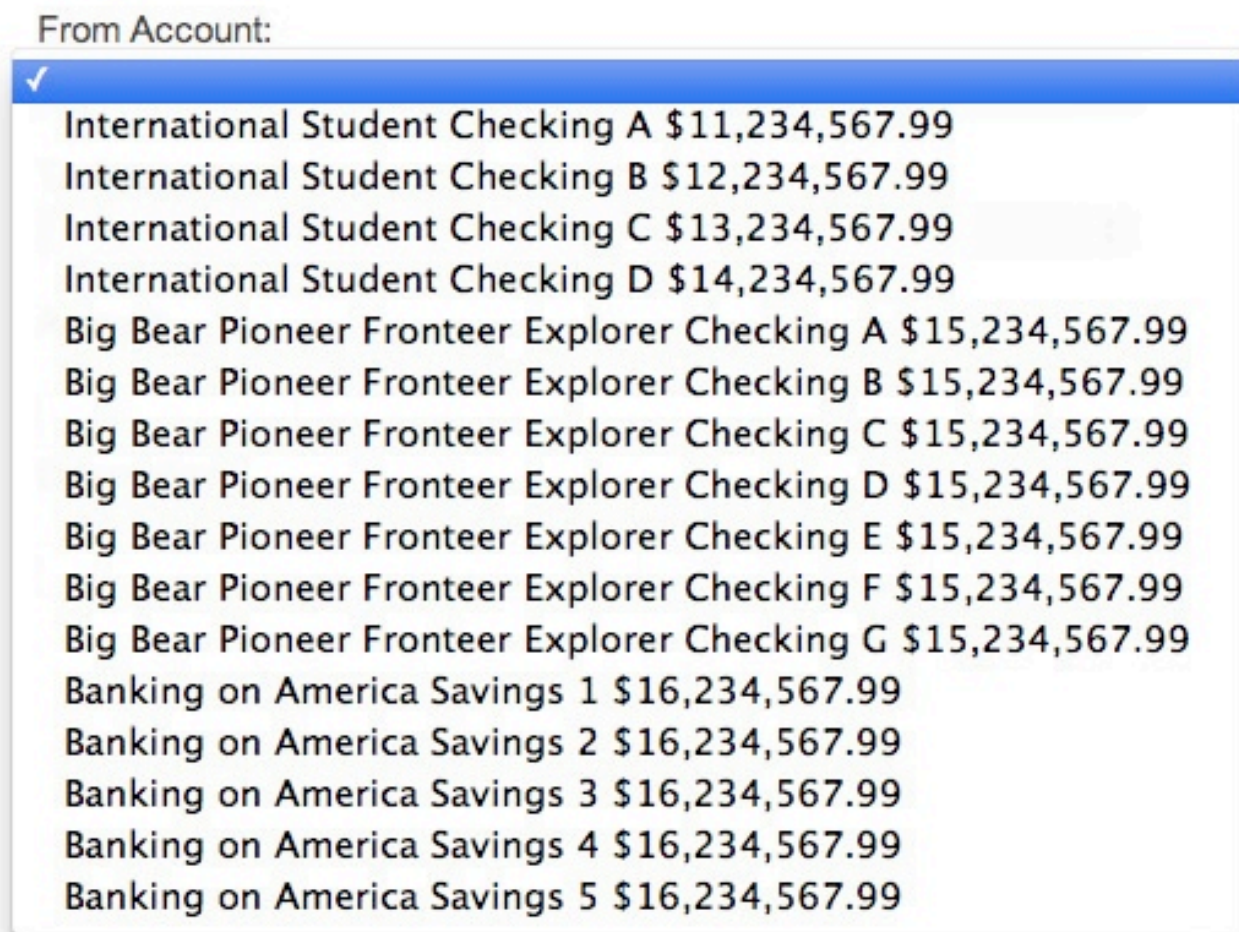
To Account:

Amount:

Description:

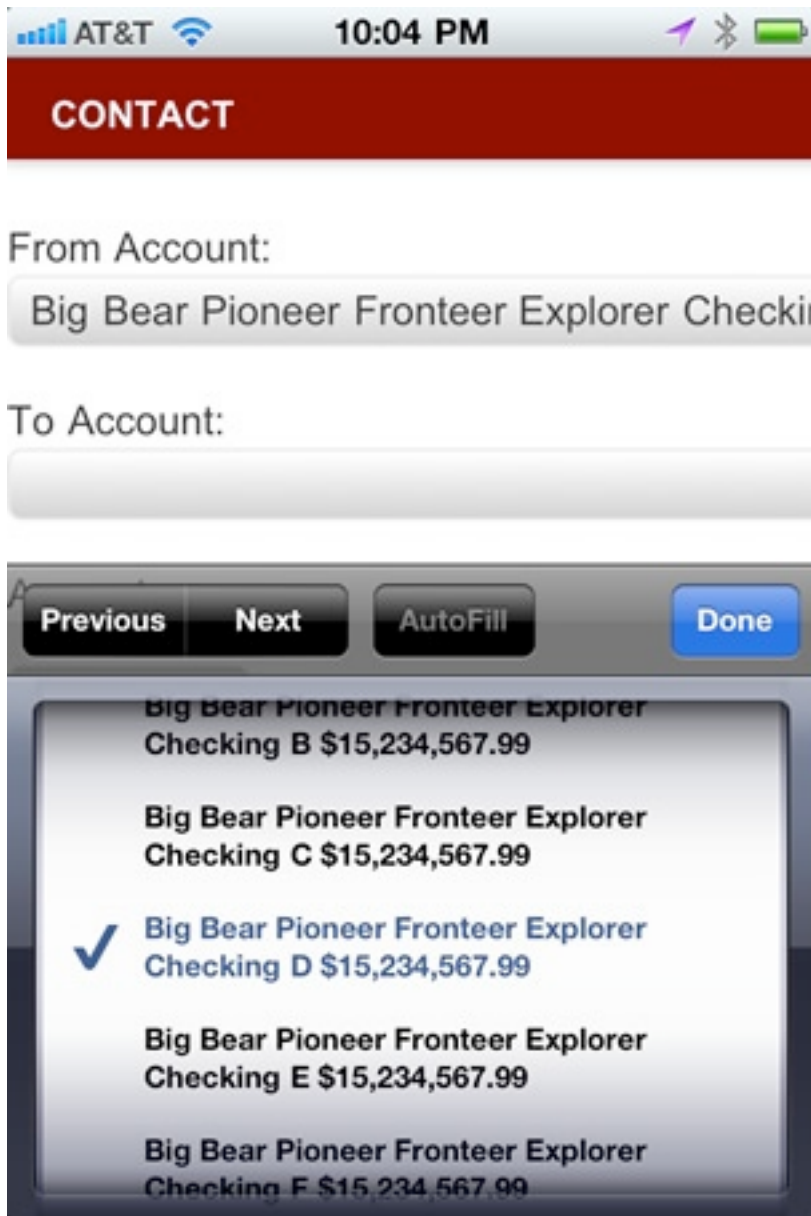
Typical me-to-me transfer via Web form.

The drop-down menu works reasonably well on the desktop Web, assuming the customer has between 1 and about 20 or 30 accounts. Each account can be listed in the drop-down menu by its full name, along with the account balance:



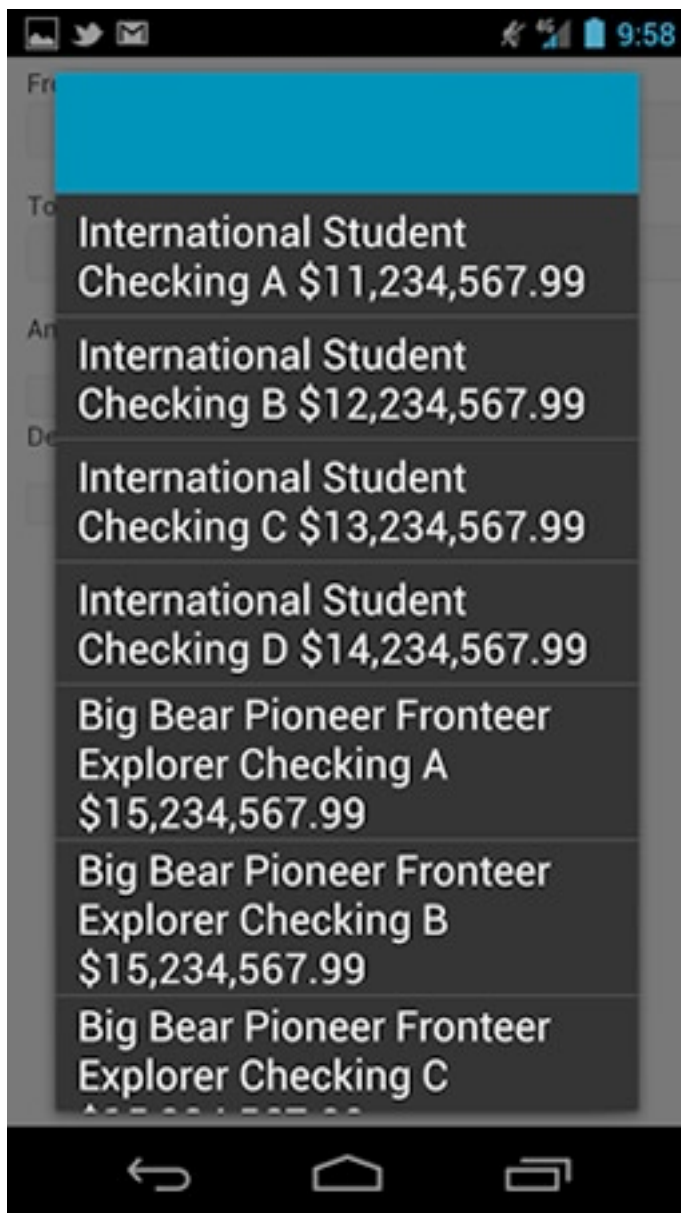
Selecting an account via the Web form's select control.

How does this translate to mobile? Not very well. Blindly copying the desktop Web is a knee-jerk reaction, and it turns out that it's mostly unsuccessful, resulting in a subpar experience. Here is why. Instead of the 20 or 30 selections that can be displayed in the drop-down menu, the iPhone's standard picker control shows only 3 full and 2 partial choices:



Selecting an account via iOS' picker control.

In Android 4.0 (the latest public Android version, named Ice Cream Sandwich), the situation is slightly better. Instead of the picker, the Android OS uses the spinner overlay, which shows 8 options. Unfortunately, the formatting options are quite limited, and the text area in the overlay is about 20% narrower than the main screen because the spinner is not using the full width of the device. This leads to confusing double and triple wrapping of text and numbers:



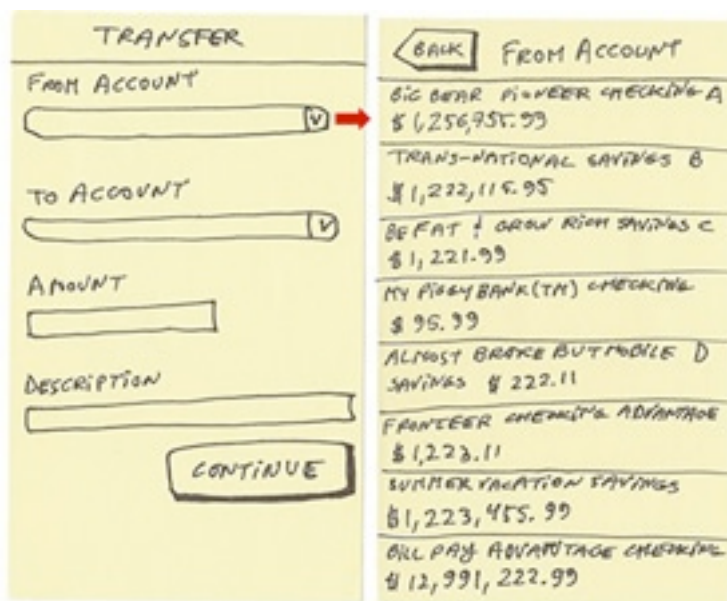
Selecting an account via Android's spinner control.

Interestingly, some online banks use this pattern to display a list of accounts. However, by necessity (and to avoid the confusion of wrapping and truncation on older and smaller low-end devices), they use short codes for account names, such as CHK, SAV and CC1. These abbreviations work reasonably well for text banking, where the short-and-sweet mental model (“C U L8R”) reigns supreme. However, code abbreviations are far from the slick world-class UI elements that consumers have come to expect from their smartphones. Rather, they smack of “dim phones,” BlackBerrys, DOS and enterprise software. Having to remember codes to do mobile banking is a far cry from the experience of playing Angry Birds or shopping on Amazon or Gilt. To create a better experience on mobile, we need another design pattern: a dedicated selection page.

Dedicated Selection Page

A slicker and more usable mobile design pattern for listing accounts than pickers and spinners would be a dedicated selection page (also called “table view”) in which 10 or more account options could be listed comfortably. As Apple’s [iOS developer guidelines](#) state, “Consider using a table view, instead of a picker, if you need to display a very large number of values. This is because the greater height of a table view makes scrolling faster.”

This is how it looks wireframed using the agile, lightweight, sticky-note methodology (see the “References” at the end of this article):



Selecting an account via a dedicated selection page (wireframe).

The advantages of using a dedicated selection page over a picker or spinner include the following:

- Any font and branding you like;
- A platform-independent experience;
- Use the entire width of the page;
- Text wraps as needed, so multiple device profiles can use the page comfortably;
- Display 10 or more options at a time, with comfortable scrolling.

The bottom line is that, with a dedicated selection page, you can easily display the account's full name and balance.

How could this pattern be used with a form? One popular pattern is a form with dedicated selection pages. Unfortunately, this often creates very long flows.

Form With Dedicated Selection Pages

The idea behind this pattern is simple: copy the standard desktop Web form but use dedicated selection pages instead of pickers or spinners.

Using this mobile design pattern, our me-to-me transfer flow would look like this:

1. Blank form;
2. Dedicated page to select the “From” account;
3. Back to the form (with the “From” field now filled in);
4. Dedicated page to select the “To” account;
5. Back to the form (with both the “To” and “From” fields now filled in);
6. Fill in the amount, etc., and hit “Continue”;
7. Verification page.

Here is the flow wireframed using the agile lightweight sticky-note methodology:

The wireframe consists of seven sticky-note pages connected by red arrows, illustrating a 'Me-to-me' transfer process:

- Page 1: TRANSFER**
 - FROM ACCOUNT: [Input field] (V)
 - TO ACCOUNT: [Input field] (V)
 - AMOUNT: [Input field]
 - DESCRIPTION: [Input field]
 - [CONTINUE]
- Page 2: FROM ACCOUNT**
 - BACK
 - BIG BEAR PIONEER CHECKING A \$1,256,955.99
 - TRANS-NATIONAL SAVINGS B \$1,222,115.95
 - BE FAT & GROW RICH SAVINGS C \$1,221.99
 - MY PIGGY BANK (TM) CHECKING \$95.99
 - ALMOST BRAKE BUTTMOBILE D SAVINGS \$222.11
 - FRANKEER CHEMIST'S ADVANTAGE \$1,223.11
 - SUMMER VACATION SAVINGS \$1,223,455.99
 - BILL PAY ADVANTAGE CHECKING \$12,991,222.99
- Page 3: TRANSFER**
 - FROM ACCOUNT: BIG BEAR PIONEER CHECK (V)
 - TO ACCOUNT: [Input field] (V)
 - AMOUNT: [Input field]
 - DESCRIPTION: [Input field]
 - [CONTINUE]
- Page 4: TO ACCOUNT**
 - BACK
 - TRANS-NATIONAL SAVINGS B \$1,222,115.95
 - BE FAT & GROW RICH SAVINGS C \$1,221.99
 - MY PIGGY BANK (TM) CHECKING \$95.99
 - ALMOST BRAKE BUTTMOBILE D SAVINGS \$222.11
 - FRANKEER CHEMIST'S ADVANTAGE \$1,223.11
 - SUMMER VACATION SAVINGS \$1,223,455.99
 - BILL PAY ADVANTAGE CHECKING \$12,991,222.99
 - KIDS COLLEGE FUND \$99,999,999.12
- Page 5: TRANSFER**
 - FROM ACCOUNT: BIG BEAR PIONEER CHECK (V)
 - TO ACCOUNT: TRANS-NATIONAL SAVINGS (V)
 - AMOUNT: [Input field] →
 - DESCRIPTION: [Input field]
 - [CONTINUE]
- Page 6: TRANSFER**
 - FROM ACCOUNT: BIG BEAR PIONEER CHECK (V)
 - TO ACCOUNT: TRANS-NATIONAL SAVINGS (V)
 - AMOUNT: 50.00
 - DESCRIPTION: [Input field]
 - [CONTINUE]
- Page 7: TRANSFER**
 - ONE MORE STEP: PLEASE VERIFY INFORMATION BELOW AND HIT "TRANSFER":
 - FROM ACCOUNT: BIG BEAR PIONEER CHECKING
 - TO ACCOUNT: TRANS-NATIONAL SAVINGS
 - AMOUNT: \$50.00
 - [EDIT] [TRANSFER]

Me-to-me transfer via a form with dedicated selection pages (wireframe).

While this pattern works, it makes the flow quite long: seven pages. Could it be shortened? Absolutely. One excellent mobile-first pattern is the dedicated wizard flow.

Dedicated Wizard Flow

This is an extreme adaptation of the desktop Web form. This pattern works extremely well for short forms because it dispenses with desktop forms entirely, using a dedicated page for each attribute of the form.

Using this pattern, our me-to-me transfer flow would look like this:

1. Dedicated page to select the “From” account;
2. Dedicated page to select the “To” account;
3. Dedicated page to enter the amount, with a numeric keyboard;
4. Verification page.

And here is the flow using the agile methodology:

The wireframe illustrates a four-page wizard flow for a me-to-me transfer:

- Page 1: TRANSFER**

FROM ACCOUNT	TO ACCOUNT	AMOUNT
BIG BEAR PIONEER CHECKING A		\$ 6,256,955.99
TRANS-NATIONAL SAVINGS B		\$ 1,222,115.95
BE FAT + GROW RICH SAVINGS C		\$ 1,221.99
MY PINKY BANK (TM) CHECKING		\$ 95.99
ALMOST BROKE BUT MOBILE D		SAVINGS \$ 222.11
FRONTIER CHEERLEADER ADVANTAGE		\$ 1,223.11
SUMMER VACATION SAVINGS		\$ 1,223,455.99
BILL PAY ADVANTAGE CHECKING		\$ 12,991,222.99
KIDS COLLEGE FUND		
- Page 2: TRANSFER**

FROM ACCOUNT	TO ACCOUNT	AMOUNT
TRANS-NATIONAL SAVINGS B		\$ 1,222,115.95
BE FAT + GROW RICH SAVINGS C		\$ 1,221.99
MY PINKY BANK (TM) CHECKING		\$ 95.99
ALMOST BROKE BUT MOBILE D		SAVINGS \$ 222.11
FRONTIER CHEERLEADER ADVANTAGE		\$ 1,223.11
SUMMER VACATION SAVINGS		\$ 1,223,455.99
BILL PAY ADVANTAGE CHECKING		\$ 12,991,222.99
KIDS COLLEGE FUND		
- Page 3: TRANSFER**

FROM ACCOUNT	TO ACCOUNT	AMOUNT
		AMOUNT
		1 2 3
		4 5 6
		7 8 9
		+ * # 0 <input type="button" value="X"/>
		<input type="button" value="CONTINUE"/>
- Page 4: TRANSFER**

ONE MORE STEP: PLEASE VERIFY INFORMATION BELOW AND HIT "TRANSFER":

FROM ACCOUNT	TO ACCOUNT	AMOUNT
BIG BEAR PIONEER CHECKING	TRANS-NATIONAL SAVINGS	\$ 50.00

Me-to-me transfer via a dedicated wizard flow (wireframe).

This pattern works very well for short forms, and it is a great example of Luke Wroblewski's [mobile-first](#) design thinking. The entire flow is accomplished in only four steps. Note that a verification page (allowing the customer to review the entire transaction before tapping the final “Transfer” button) is recommended with this pattern. Note also the use of the breadcrumb pattern, which shows the customer which step of the workflow they are on and how many steps remain. The breadcrumb enhances this design pattern nicely.

Are we done? Should you create a dedicated wizard for every flow on your mobile banking website? Not so fast.

In mobile, nothing comes for free. That includes the dedicated wizard flow, which completely breaks down in longer forms. The basic idea is to have a dedicated page for each element of the form. But if the form has five or more elements, then the flow starts to get too long. Another issue is the inability of this pattern to distinguish between optional elements (such as “Memo”) and required elements (such as “Amount”). With this pattern, each element gets its own entry page with the appropriate keyboard and is likely to be treated as “required”. Even if the customer understands that they don't need to enter anything, each element requires the customer to at least look at the page and click “Continue.”

So, is there another pattern for a page with five or more elements and many optional fields? I'm glad you asked. One of the most versatile yet underused patterns is the wizard flow with form. And as a bonus, this pattern dispenses with the need for a separate verification page.

Wizard Flow With Form

The idea here is very simple. Start with a dedicated page to select each account, and then show the rest of the fields in a form.

Using this pattern, our me-to-me flow would look like this:

1. Dedicated page to select the “From” account;
2. Dedicated page to select the “To” account;
3. Continue to the form (with both the “To” and “From” fields now filled in);
4. Fill in the amount, etc., and hit “Continue”.

Here is the wireframe using the agile methodology:

The wireframe illustrates a four-screen process for a me-to-me transfer:

- FROM ACCOUNT:** A list of accounts with their balances. A red arrow points from the top account, "BIG BEAR PIONEER CHECKING-A", with a balance of \$1,256,755.99.
- TO ACCOUNT:** A list of accounts. A red arrow points from the top account, "TRANS-NATIONAL SAVINGS B", with a balance of \$1,222,115.95.
- TRANSFER (Left):** The "FROM ACCOUNT" field is populated with "BIG BEAR PIONEER CHECKING-A" and "\$1,256,755.99". The "TO ACCOUNT" field is populated with "TRANS-NATIONAL SAVINGS B" and "\$1,222,115.95". Below these are input fields for "AMOUNT" (with a red arrow pointing to it) and "DESCRIPTION". At the bottom are "CANCEL" and "TRANSFER" buttons. A note states "TRANSFER WILL BE AVAILABLE IMMEDIATELY".
- TRANSFER (Right):** Similar to the previous screen, but the "AMOUNT" field is pre-filled with "50.00".

Me-to-me transfer via the wizard flow with form.

This mobile design pattern combines the best features of Web forms, such as the flexibility of having optional fields and multiple input fields, with the vastly improved usability of dedicated, mobile-optimized selection pages. Another boon is the option to dispense completely with the verification page, because the form page itself acts as an editable verification page. Of course, you could always append a separate verification page if you really must.

Editing is also much easier than with most other patterns. Instead of having to go through the entire flow again, the customer only has to edit the fields that need correction. For example, to edit the “To” account, the customer would tap the corresponding field in the form and be taken to the dedicated “To” account page, and then immediately back to the form, without having to go through the entire **“From” account → “To” account → Amount** flow again.

Mobile: Thinking Differently

As we’ve seen now, mobile design itself is usually not complicated. In fact, because people will be using your app with fat meaty pointers (commonly called “fingers”) in a stressful multitasking environment (commonly called “life”), a less complicated design is virtually guaranteed to perform better. However, designing for mobile is one of the most sophisticated exercises any of us is likely to encounter. Simply copying successful desktop patterns is usually the worst choice, yet it is the one many designers naturally gravitate to.

Designing for mobile requires thinking differently. Remember that in mobile, each form field requires at least an extra tap to bring up the keyboard, picker, dedicated page or other element to enter data. Instead of a vertical flow to guide the person to the next field, consider using a horizontal flow instead. Look for options and inputs to eliminate. Whenever possible, minimize the number of pages the person has to go through in order to complete the workflow; this will reduce input errors and increase customer satisfaction. Last but not least, make user testing the core of your mobile design process, and be sure to user test everything you design as early as possible.

Seven Guidelines For Designing High-Performance Mobile User Experiences

Ivo Weevers

A positive first impression is essential to relationships. People look for trust and integrity, and they expect subsequent encounters to reflect and reinforce their first impression. The same principles apply to brands and their products. Design plays an important role in building lasting relationships with end users and, thus, in supporting the brand's promise.

Users expect mobile services to be relevant and user-friendly and to perform well. The limitations of the medium, however, impose significant challenges to designing products that meet all of those expectations. While often underestimated, performance is a crucial contributor to a trustworthy mobile user experience. Therefore, it should be considered a key driver in the design process.

In this article, we'll discuss performance in relation to design and present seven guidelines that can help shape design decisions related to performance while accounting for the needs of end users and businesses. These guidelines are based on the experiences of our teams in designing native mobile apps for a broad product portfolio and on multiple mobile platforms.

Performance For Mobile

People use their mobiles to enhance productivity, comfort and pleasure, everywhere and at anytime: waiting for the bus, walking on the sidewalk, checking which platform their train leaves from. Mobile applications need to focus on a core utility, and they need to be fast and reliable in order to be valuable in those environments.

Paradoxically, we've noticed that many mobile design cycles start with requests for great aesthetics first: "It has to look amazing!" Of course, stunning visuals will attract customers by making a great first impression. However, a long-lasting relationship depends on the quality of each encounter, which is also heavily influenced by how the app performs. An application that looks stunning but performs poorly can damage integrity. Users quickly notice software that is slow or likely to break (whether because of downtime, crashes, etc.), and this impairs both usage and brand perception. Users expect an app to be fast and responsive. If it's not, it will get poor reviews, low ratings and low adoption numbers.

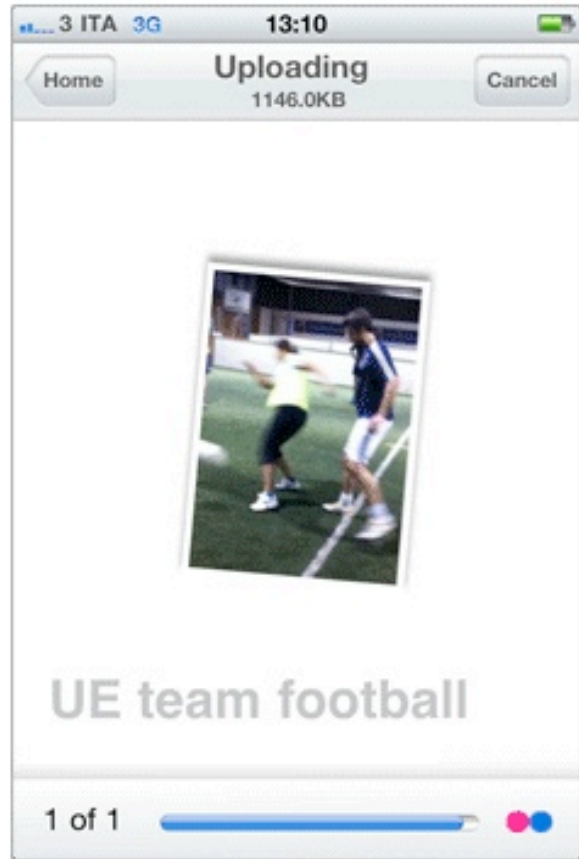


The visuals in the Twitter app are not as rich as Cookmate's (which are stunning), but reviews for the Twitter app in the App Store are much better.

Performance Supports Brand Differentiation

Every product encounter affects the brand perception. Users are looking for the best tools to enhance their lives. To attract a significant user base in a landscape that is becoming more populated each day, a mobile app has to stand out. It should do something no other app does, or do it better than others do. These benefits reinforce the brand. Because performance is an essential element in the user experience, it directly helps to differentiate the brand. Taking this one step further: making a technically challenging feature perform smoothly will give the product a unique selling point, one that will be difficult to imitate.

A good example is the Flickr iPhone app. Flickr states on its website that it has two main goals: to help people to make their photos available, and to enable new ways to organize photo and video. Guess what? That is exactly what its app does well, thus fulfilling its brand's promise.



Flickr does a good job of optimizing the key brand encounters.

So, when planning an app, it is worth analyzing the market and trying to answer two questions. (1) What should it do differently from its competitors? (2) What should it do better than its competitors? Then, focus your design and development efforts on the resulting top three goals.

A Key Design Exercise

Crafting products of any kind requires an appreciation of the way they are built. Well-established design and engineering disciplines have recognized this for a long time. A car's design influences its aerodynamics. A beautiful eye-catching bridge has to cope with wind and with traffic passing over and beneath it. The same applies to interactive mobile services. Design choices affect features, content, interactivity, graphics and, therefore, performance.

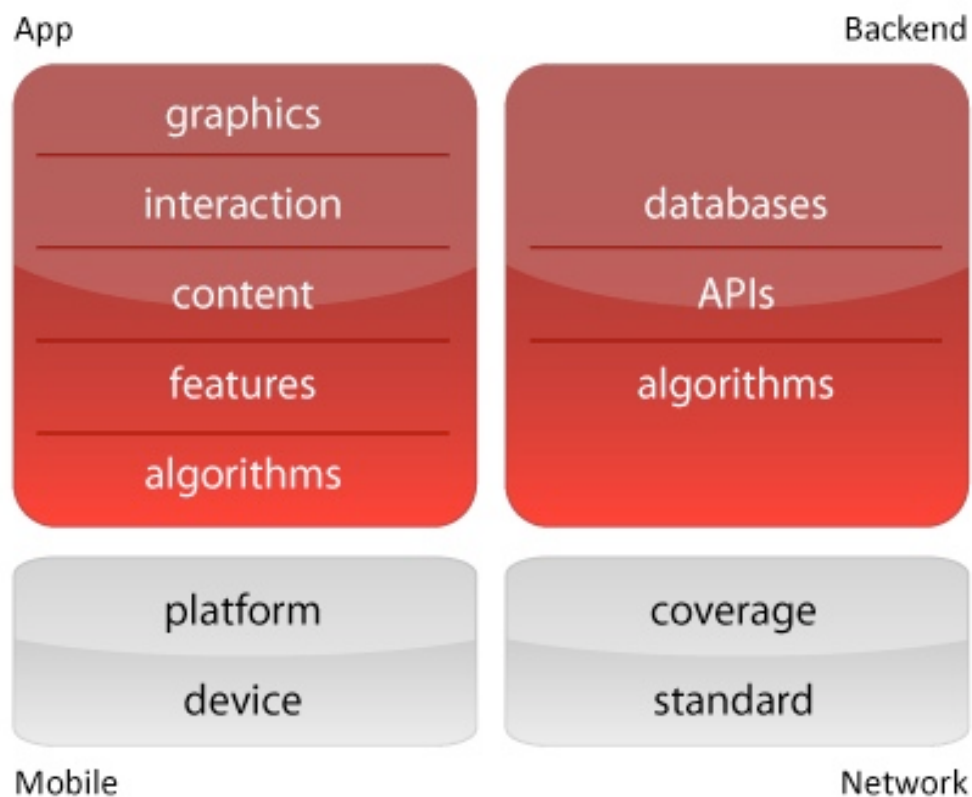
We've noticed on several product teams that performance is believed to be exclusively the responsibility of developers, and therefore it is considered too late in the design process. But in order to assess feasibility, development needs to be considered during the creative process. Applying the seven guidelines below and considering relevant factors for mobile UI (see the next section) up front will help to incorporate the topic of performance into design discussions. Achieving a high-performance experience is not just a coding exercise. It is a key design exercise.



The Erasmus bridge in Rotterdam, the Netherlands. (Image: [Wikipedia](#))

Relevant Factors In Mobile UI Performance

The perception of performance is based on start-up time, page-loading behavior, smoothness of transitions and animations, errors, and waiting times. The diagram below illustrates these factors: the “app” (with its graphics, interaction, content, features and code) runs on a “mobile” device with certain technical capabilities (CPU, screen size, etc.) and a platform (Android, iOS, etc.). In many cases, the app connects over a “network,” with a set coverage and standard (LTE, 3G, 2G), to a “back end.” The two factors at the top are heavily influenced by the decisions of the design and development team. The bottom two factors are constraints that need to be taken into account.



Factors that influence performance.

Decisions for each factor will affect performance. Any combination, enhancement (such as advanced visuals) or limitation (such as poor network

coverage) could increase complexity. For example, content being loaded from the back end in addition to advanced graphics traveling over a slow network are a combination of factors that will reduce performance.

Seven Guidelines

The last years have seen dramatic changes in the mobile platform landscape. New UI paradigms have emerged, screens and processors are becoming as advanced as desktop computers, and input mechanisms have been revolutionized.

Within these shifting constraints, designers should always try to create a look and feel that is cutting-edge, memorable and high-performing. This is not just a matter of reducing image sizes. Decisions made at various levels of the design and the design process will have a significant impact. Let's look at seven guidelines that have proven to be helpful tools at all design levels to achieve high-performing mobile user experiences.

1. DEFINE UI BRAND SIGNATURES

Each user interaction with an app should reflect the story of the brand and should increase recognition, loyalty and satisfaction. Identifying which elements contribute most to the brand's identity is essential. Examples are features, visuals, wording, fonts and animations. Our design teams work on many different products on different product teams. This could easily lead to several design and implementation variations of similar UI elements.

Defining the core building blocks encourages reuse and discourages reinvention and, therefore, optimizes the design and implementation of a set of components.

One approach is to define the UI elements that form the core building blocks of the user interface and, together, to create the interface's unique character. In the concept phase, identify those elements that do the following:

- Differentiate the app (for example, the photo-viewing feature in the Path app);
- Represent key functions (for example, a check-out feature for a store);
- Set the pattern of the design language (for example, the header in the screenshot below).



Windows Phone 7's Metro UI is a great example of how fonts, layout and interaction can establish a unique design DNA. (Image: [Wikipedia](#))

The core signature elements need to be the most responsive. They will be seen by users over and over and will be reused in different product features. By focusing the design and implementation on this set of elements, each optimization will pay off multiple times.

2. FOCUS THE PORTFOLIO OF PRODUCTS

Whether a company wants to launch a product quickly, or develop a product portfolio (i.e. multiple products on one platform, the same product on multiple platforms, or both), or if facing limited time and resources, hard choices have to be made. Design and optimization efforts should be targeted at those products in the portfolio that matter most. A design priority matrix helps us understand where design efforts will pay off the most.

Product	A		B	
Platform				
Current customer base	200k	300k	100k	50k
Expected customer base	500k	350k	350k	75k
Competitors	C1, C2, C3	C2	C4, C5	-
Design priority	★★★★		★★★	★★

Example of a design priority matrix.

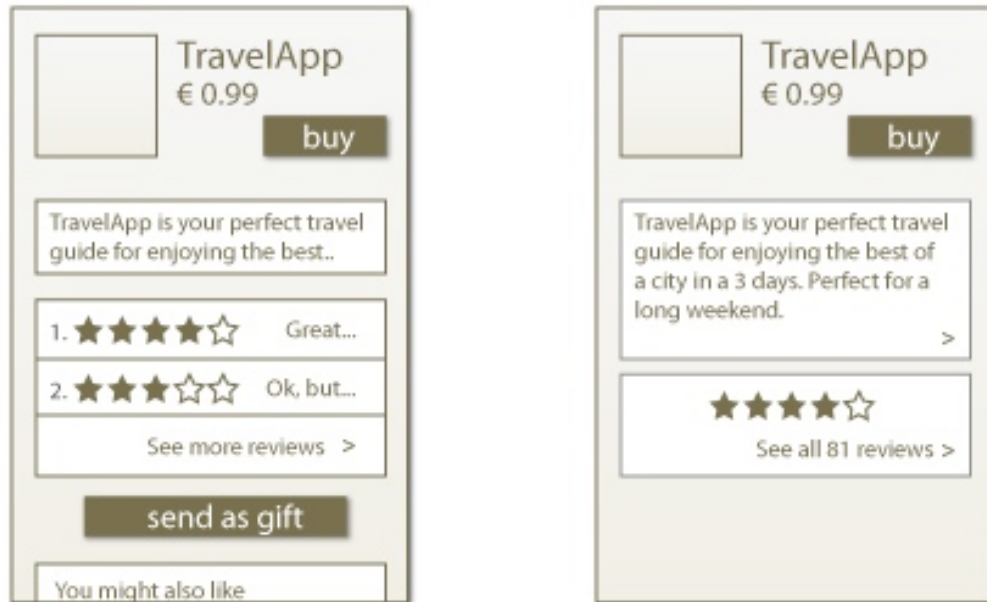
Focusing design efforts helps to optimize performance in the most rewarding areas. For example, if most of your anticipated customers are using Android phones, and competitors are also targeting them, dedicating more design effort to creating an elegant and fast Android app would be more valuable than dividing your efforts equally across all platforms.

3. IDENTIFY THE CORE USER STORIES

Our teams have faced several project kick-offs in which the initial list of requested features was lengthy, unfocused and impossible to build within the requested timeline. When dreaming up what a product should do, companies often lose sight of the fact that customers look for solutions that help them with very particular needs.

For example, one main shopping goal (besides socializing, inspiration, etc.) is to find and purchase a product. Whether in a small city, on Oxford Street in London or on the Internet, it's about finding and buying what you're looking for. The experience could be enriched to make shopping more fun, but the core goal — finding and purchasing — should never be lost. The same applies to the design of a shopping app (whether for games, music, vouchers). The user needs to be able to find and purchase quickly, regardless of whatever other functions that enrich the overall experience.

The illustrations below show two designs for a product detail page in a store. The left screen has advanced shopping features, such as gifting, related products and detailed reviews. The right screen is more focused on purchasing. Implementing the right screen will optimize the company's story, and the team won't get distracted by designing and implementing side features. Only when the basic core functionality is optimized can enrichment features be added, as long as they do not hinder the core user stories.



Two kinds of purchasing screens.

During the product definition process, the core user stories should be identified in order to focus the design and development efforts.

4. OPTIMIZE UI FLOWS AND ELEMENTS

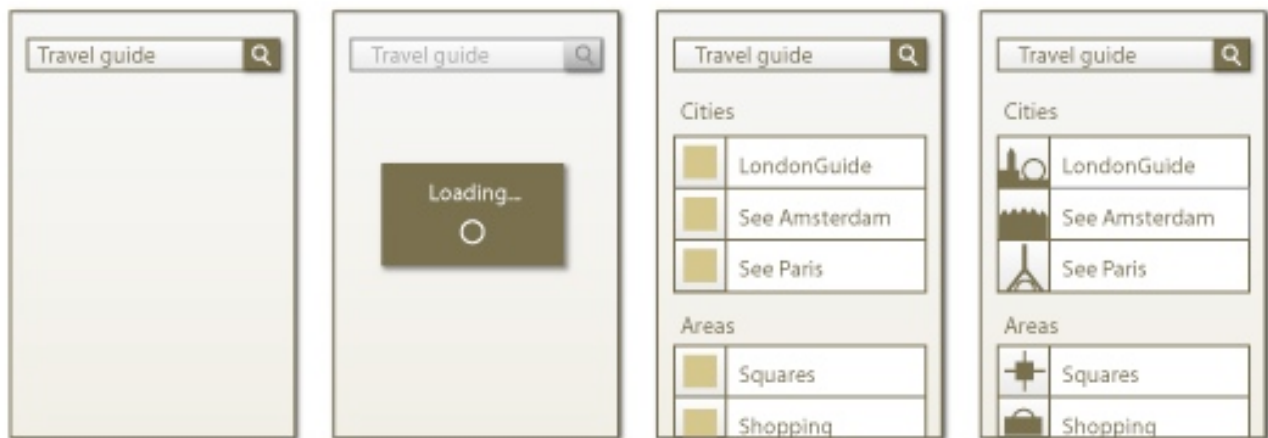
Users don't like to wait. (Google puts “[Every millisecond counts](#)” as the second principle of its user experience.) Optimizing individual screens, flows and UI elements will reduce waiting times and keep users from thinking that they're wasting their time.

A. Speed up perceived performance

The designer cannot control performance all of the time. The network might be slow; the device might be running other tasks in the background; certain operations might require a lot of calculation. If the user at least perceives that they are not losing time, then the app will make a solid impression. Design can help communicate this, even during unexpected delays.

The first step is to identify flows that will likely have delays (fetching back-end data, performing a lot of calculations, etc.). The second step is to guide users through these delays by introducing additional steps that they would perceive as being necessary (showing loading animations, displaying useful tips, etc.).

The following set of images shows possible steps in a content search:



The user here experiences four steps:

1. Hits the search button.
2. Sees a loading animation.

3. Sees the first part of the list, with textual content and placeholder images (which could be stored in the app itself).
4. Sees the actual thumbnail images appear.

The user experiences short steps, rather than jumping directly from step one to four, and so perceives progress rather than delay.

Another example is when an app starts loading up. By first displaying a picture that matches the application's layout, the user gets the impression that the app is loading more quickly. The screenshot below illustrates this; however, the perceived performance could be sped up even more by adding a simple progress notification in the blank space of the first screen. This would avoid the impression that something is waiting to be loaded. (In case of a slow connection the app does show a loading notification, thereby communicating progress to the user for that situation).



The Facebook app for the iPhone loading up.

B. Optimize individual UI elements

Every UI element affects performance. And because every optimization contributes to overall performance, all UI elements should be considered. Key aspects to look at are:

- **Elements on screen**

The number and type of UI elements on the screen will affect the performance of that screen. For example, media items (audio, video, maps) will affect performance more than simple elements (static images, etc.).

- **Element characteristics**

The characteristics of an element, such as its resolution or image depth, affects drawing time. For example, on Android, each drawable resource (JPG, PNG) is decoded to bitmap format, so each optimized image will result in fewer kilobytes. Could you reduce the color depth? Or decrease the resolution?

- **Drawing technique**

The way a UI element is drawn by the app affects screen-loading time. For instance, is the entire background of a screen being drawn, even when a big opaque image is laid on top of it? Could a background be broken down into small tiles in order to reduce the size that needs to be uploaded?

5. DEFINE UI SCALING RULES

Building the most appealing design is like navigating a terrain with many hurdles. It is a continual balancing act between functionality, aesthetics, usability and performance. Some platforms demand more UI compromises than others. No matter what the platform's constraints, the brand's key signatures should remain.

A UI scaling toolkit could help by communicating the relative importance of UI elements. Some elements are critical and contribute strongly to the brand's identity, while removing others will have less of an impact. Our team has established the following categories:

- **Essentials**

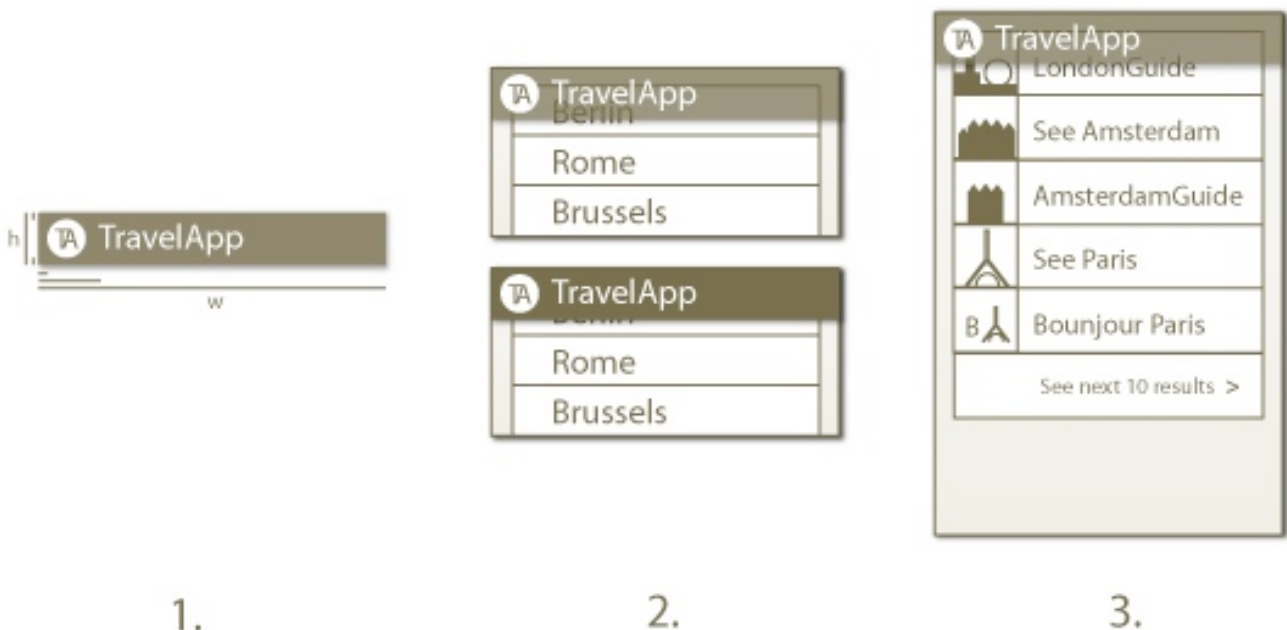
Essentials are the brand's core UI signatures (guideline 1). For example, the application's header.

- **Alternatives**

Alternatives are less optimal, but good for high-end solutions that put a low burden on performance. An example is replacing transparent elements with opaque ones.

- **Options**

These are elements that enhance the experience but could be removed to maintain performance. For example, reducing a list of search results on a page from 25 items to 10.



1. Essentials (the header).

2. Alternatives (opaque instead of transparent).

3. Options (reducing list length).

6. USE A PERFORMANCE DASHBOARD

Clear communication among the team is critical to delivering a great product. We've encountered several situations where expectations of how a product should perform differed between marketers, designers and developers. Because performance is affected by the requirements and constraints of all of these disciplines, performance expectations need to be agreed on. As a solution, we introduced performance dashboards. These help to measure, monitor and set goals for the product's current state. Dashboards effectively communicate the product's state and the team's expectations and areas of focus. The dashboard we've used accounts for the following elements:

- **Core user stories**
Ensures that the dashboard communicates what the user experiences.
- **Benchmark**
Compares the app to a key competitor's.
- **Current measurement**
Shows the performance of the product's current implementation.
- **Goal**
Sets the performance goal for the app.
- **Status**
Indicates the current status of the app against the goal.

User stories	Competitor benchmark	Current measurement	Goal	Status
Application start- up	3.1	16.3	3	
Search results	8.5	3.3	4	
Track preview	4.7	3.7	4	
Purchase track	3.1	3.3	3.5	
Genre page	8.3	7.5	5.5	
Album page	4.3	2.7	2.5	
Top 20	4.3	1.4	3	

Example of a performance dashboard. (Numbers in seconds. For a good 3G connection.)

Several tools can help you measure performance. You can do it subjectively, by manually recording the time for certain tasks, or objectively, with tools such as TraceView for the Android SDK (if you're developing a native app).

7. CHAMPION DEDICATED UI ENGINEERING SKILLS

Design has always gone hand in hand with technology. Being able to code high-performance user experiences is a specialist's skill. It requires strong knowledge of front-end coding and a profound understanding of the design's purpose.

The implementation of layout, graphics, animation and so on will have performance implications. Of the many things that need to be considered, here are two:

- **Smart loading**

Smart-loading mechanisms, such as lazy loading, first load visible content and then move on to content below the fold. This technique reduces the user's waiting time and thus makes for a smoother experience.

- **Background loading**

This is another well-known example. Performance depends on whether the background is one large image, an amalgamation of small tiles (say, to create a texture) or a pure algorithm. The best solution depends on the situation.

In situations where responsibilities are split between the marketing, design and development teams, we've noticed that UI performance tends to fall between the cracks. Each team has its own goals, and so certain shared responsibilities, such as UI performance, lose attention. We've addressed this by including front-end coding specialists on the design team. This encourages focus on optimal UI implementation and performance, and it achieves a more advanced user experience.

Conclusion

We've gone over seven guidelines that address performance relative to different aspects of mobile app design. Design choices affect performance, and so performance should be considered a key factor in the design process. Unfortunately, it tends to be looked at too late in the process, which ends up impairing the user experience significantly.

We have successfully introduced these guidelines into several product streams, which has in turn improved the performance of those products and fostered awareness of the issue among the teams. This has helped to shift the initial request from "It has to look amazing" to "It has to look, feel and work amazing!"

CREDITS

A big thank you goes to my colleagues Daniela Aramu, Mark Howell and Mirja Leinss, who provided invaluable feedback on this article!

Useful Design Tips For Your iPad App

Jen Gordon

With tools like [Appcelerator's Titanium](#) and some JavaScript programming skill, creating native iPhone and iPad apps is simple. The danger is in not being always on the look-out for the kind of design pitfalls that plague many products in the App Store. In this article, we'll consider some design tips that will get you on the road to iPad success.

Design For People

Apps will define the iPad, it's true. But in developing your app idea, which comes first, the idea or the device? Good news: neither. It's people! When brainstorming and researching ideas for your app, step back and consider the context in which the device will be used by real live people. How does the iPad fit into our lives? In what situations would we prefer this device to a laptop or iPod Touch?

WHO ARE YOUR PEOPLE?

Ideally, your target audience includes you, but using this as a reason to decide that "I know what people like me want" could lead you to miss out on opportunities to enrich the product beyond your imagination. Surprises await when you consider the variety of people who might use your product. Your target audience may be different than what you think; or in defining your target audience, you may find that your product is missing important features.

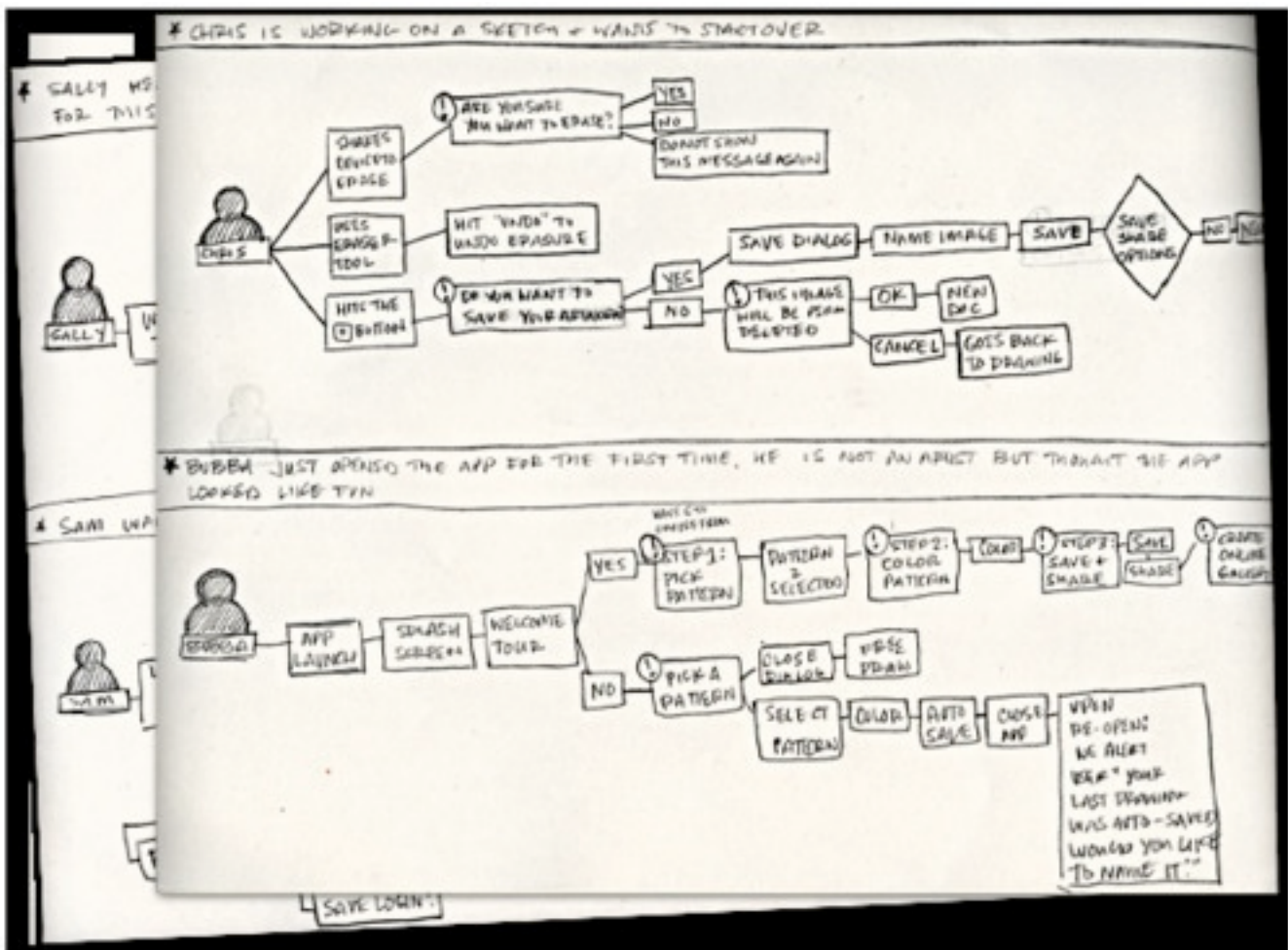
For example, with our application (a drawing tool), our initial target audience included early-adopter techies. But after some analysis, we saw that the interface needed to be greatly simplified so that the children of the techies would also enjoy the app.

Tip: Define your target audience. Who are they? Where do they live, work and play? What challenges do they face? Give one of them a name, a job, a family, a car; then see how your perspective changes.

WHAT IS YOUR PEOPLE'S STORY?

Defining a target audience is only half of the equation. Now you have to put your audience into action! What do they do in their daily life? How will their daily life intersect with your product? Get into their minds. Try this, and I guarantee it will lead you down some expected *and* unexpected paths.

You don't need fancy software to do this. Below is an example of our use case sketches for our application. After writing out a few use cases, we learned that people lose interest in drawing games when they're forced to create original artwork. Many people will say, "I can't draw," but they still have a desire to create beautiful things. Originally, we planned for our app to ship with patterns, similar to the classic Lite-Brite toy, but it didn't occur to us that people would play with the app more if we provided pre-fab patterns and templates for them to color. Pretty important feedback!



In developing the idea for an app, our use cases revealed that the replay value of such an app is low unless you provide patterns for the user to get started.

Tip: Think about the device in terms of lifestyle rather than features and technology. Will the iPad's unique characteristics and environmental and sociological factors make your idea resonate with people?

- It's lightweight = "I'll carry it more places than I would a laptop."
- It's smaller than a laptop = "It's discreet. Would I crack open my laptop to do some work in a waiting room? No. Would I switch on my iPad? Yes."

- It has robust utilities and multimedia capabilities = “I can work and enjoy books, movies and games.”
- Its screen is larger than the iPhone’s = “I can consume more media with less eye strain. My kids will be entertained on a road trip.”
- It has Wi-Fi and 3G Internet connectivity = “I can be online on a plane, train or car.”

Designing for people is critical to weeding out weak (i.e. unprofitable, untargetable or unmarketable) ideas quickly and to developing a product that not only meets but anticipates the target audience’s needs.

Minimalism Works Best on iPad

With robust, portable, location-aware devices like the iPad, the temptation is to throw in everything and the kitchen sink. If you’re an iPhone developer, you’re probably excited about the additional screen real estate. Resist the temptation to fill the space! Keep it simple. Display only the content and controls that are relevant to the user at that moment. This requires you to use the following two things in your interface design.

CONTEXTUAL MENUS

Contextual menus are a great way to keep your UI out of the way. We wanted to keep sharing and community features out of the primary navigation. We used a contextual menu (“Share this thang!”) to present actionable items at the appropriate time.



Example of a contextual sharing dialog, activated when you tap and hold on an image.

MODAL VIEWS (BUT WISELY)

With the increased real estate on the iPad, one can build in robust functionality that is impossible on an iPhone or iPod Touch screen. Modal views give you another way to organize your application; they give the user clear “modes” of operation; and they can be an elegant solution to UI clutter if executed wisely.

For example, “photos” on the iPad could operate similar to iPhoto on the desktop Mac. You have two “modes” of operation:

1. View or edit an individual photo,
2. Manage groups of photos.

In each scenario, packing the viewing, editing and managing functions into one view wouldn't make sense. Think of how you could segment features in your app, while maintaining a smooth connection between the two modes.



iPhoto has two modes of operation: viewing and editing a photo or managing photos.

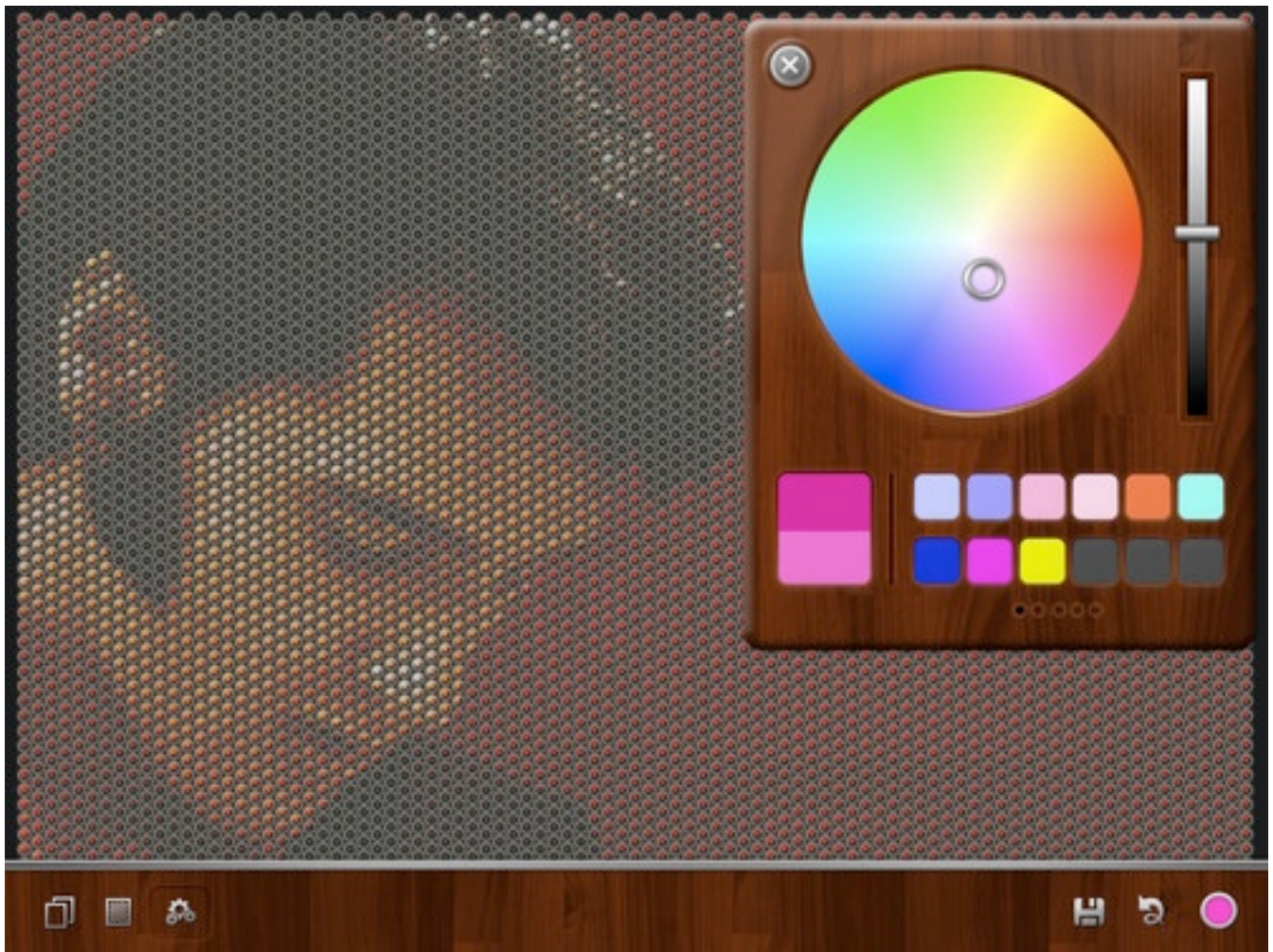
Tip: What is the bare minimum your app could provide and still be useful to users?

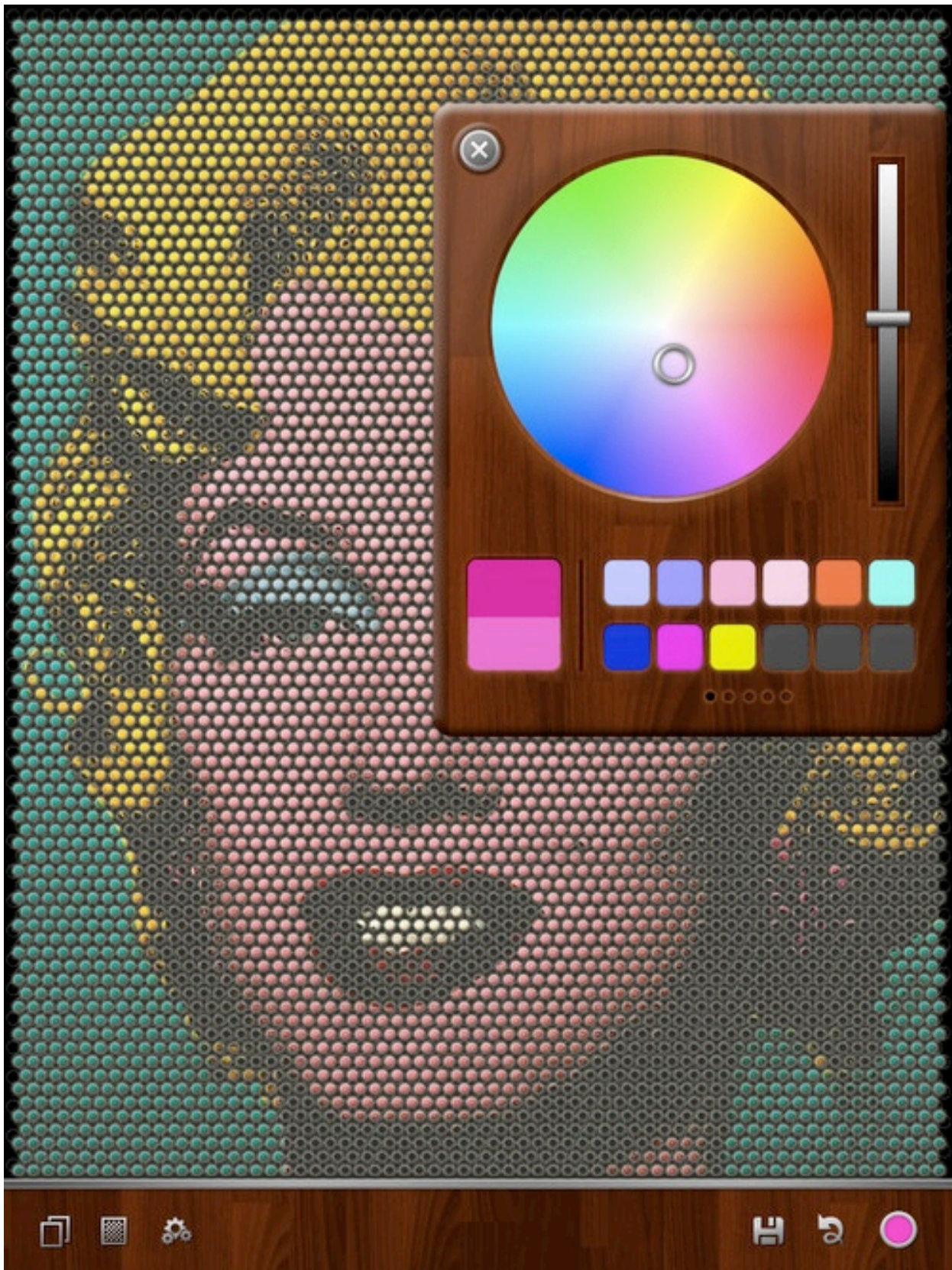
iPad's Two Orientation Are A Big Deal

Being able to switch views—landscape to portrait and back again—is not unique to the iPad, but it's a bigger deal on it. This is where paper prototyping will save you from wasting loads of time in Photoshop.

Having to consider every element of your app in these two sometimes radically different layouts is like designing for two devices... except that you're not designing for two devices. The trick is to keep the experience consistent in each view, allowing for a seamless user experience when switching views.

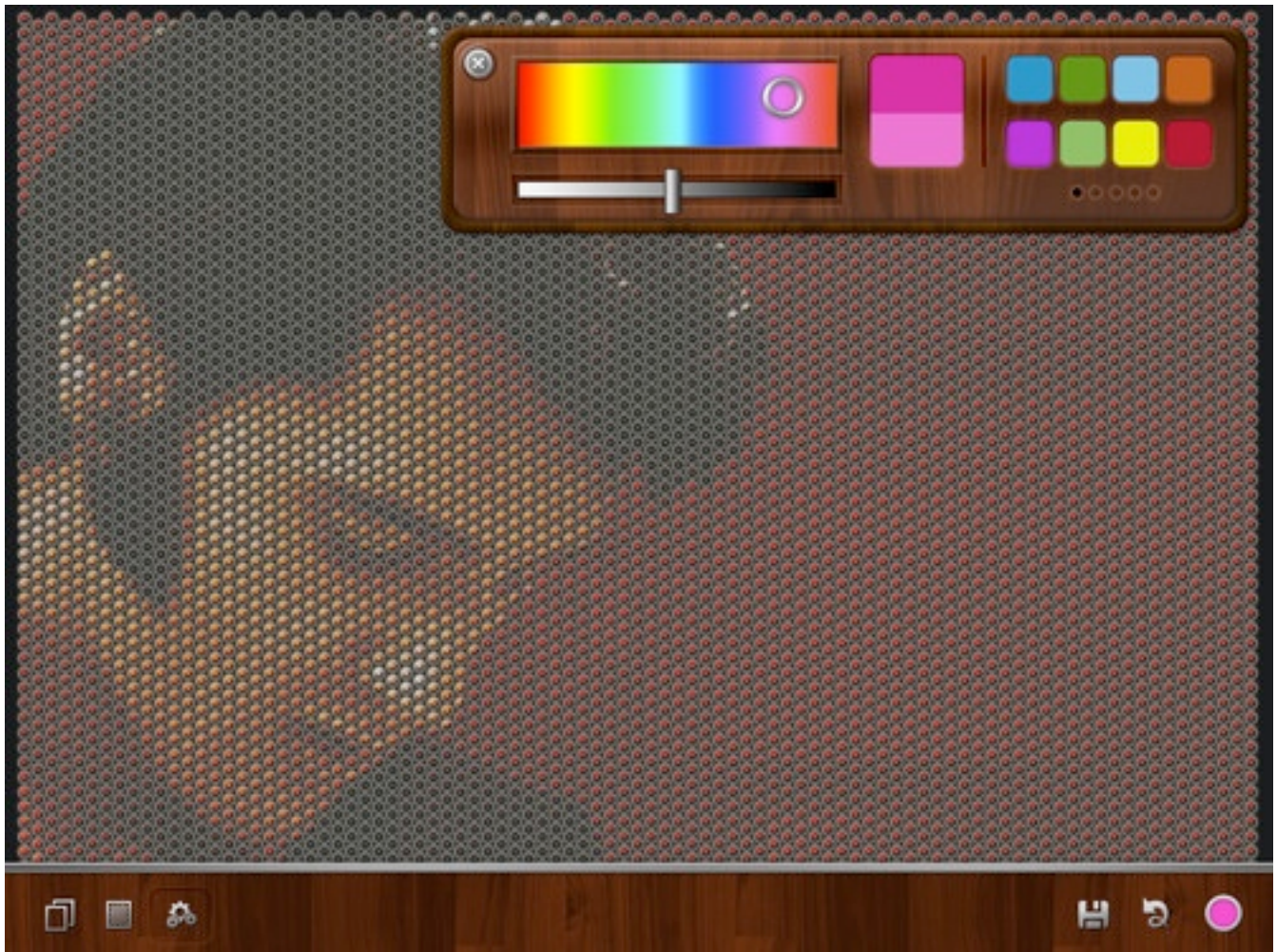
Below is a color palette we tested for our app. The palette looks great in landscape mode but absolutely hogs the screen in portrait mode. Oops.

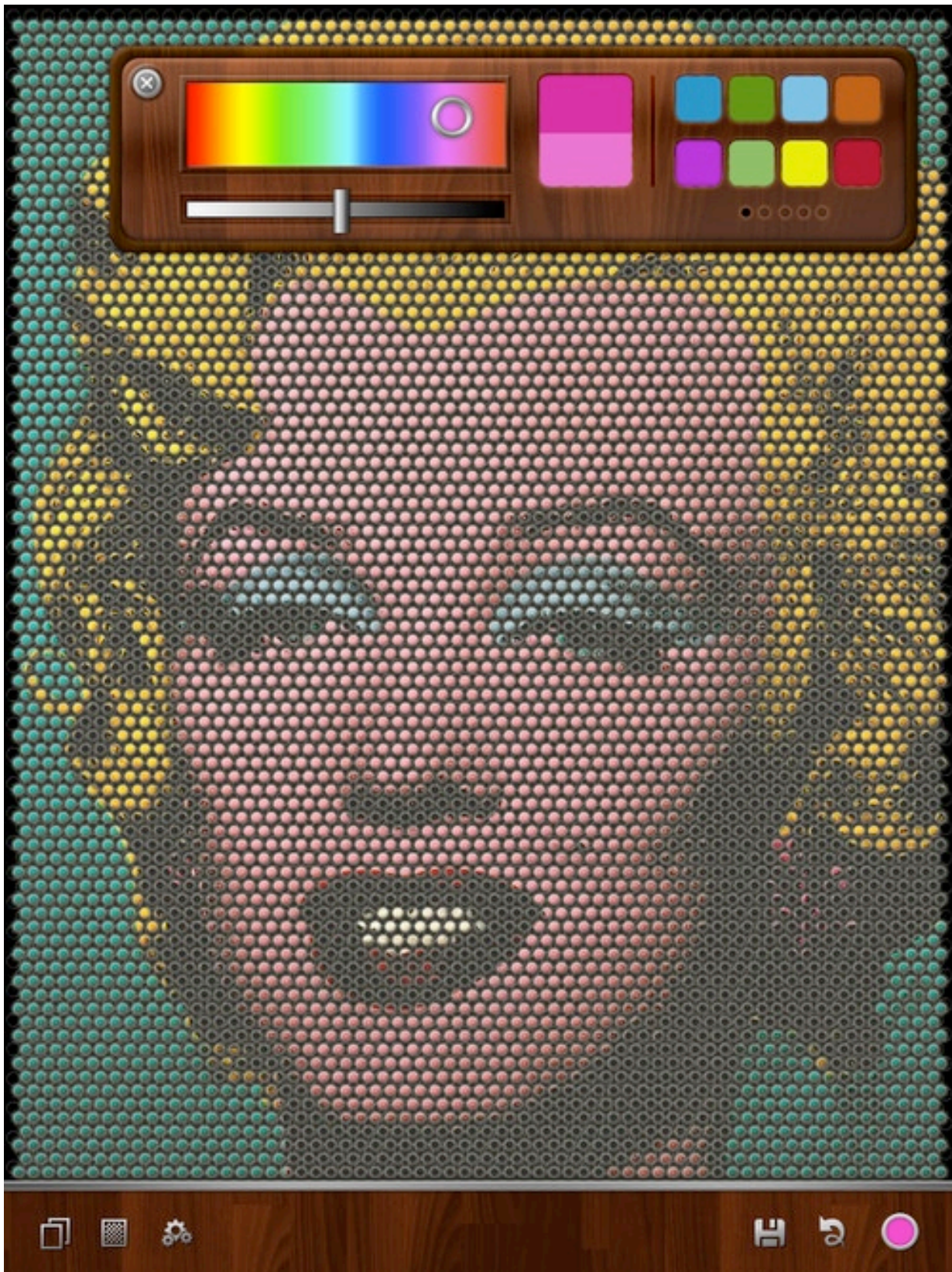




This palette looks okay in landscape mode but gobbles up the interface in portrait.

We reconfigured the color palette to have a smaller footprint in both landscape and portrait modes:





A streamlined color palette allows accessibility while staying out of the way in both landscape and portrait views.

Tip: Paper prototype all of your screens in portrait and landscape modes... a lot.

Use Touch And Real-World Metaphors

Touch changes how we interact with, edit and perceive on-screen elements. With the iPad's larger screen, touch and gestures are turbo-charged. A quick run-down of unique UI elements on the iPad:

- Select and take action on multiple items at once by dragging them to another area of the screen.
- See both a list view and details of items in that list view (e.g. Mail).
- Edit elements in place rather than from a global menu bar.



Spread your fingers over a stack of photos to spread them out for viewing, as you might in the real world.

Tip: Think of how you interact with things in the real world. Think of the on-screen elements as tangible things.

Design For Dynamic Content

The iPad's portability and bigger screen size gives users unlimited possibilities for quickly creating and sharing robust dynamic content on the go. Hybrid apps (i.e. native apps that draw from Web pages or that load real-time Web content) are becoming more common as users demand connectivity to Web-based tools. Designing for dynamic content means working through the challenges and opportunities of dealing with an Internet connection (e.g. slow downloads or lost connection). Think of what visuals you would present to users when they're stuck in East Bum with no connection to be found!

Tip: Plan for problematic situations in your design.

Get Started!

The first step to getting started is downloading the [iPad SDK](#). For Web developers looking to get into iPad development with their current skills, products such as Appcelerator's [Titanium](#) offer a good way to build native iPad, iPhone and Android apps in JavaScript.

A User-Centered Approach To Web Design For Mobile Devices

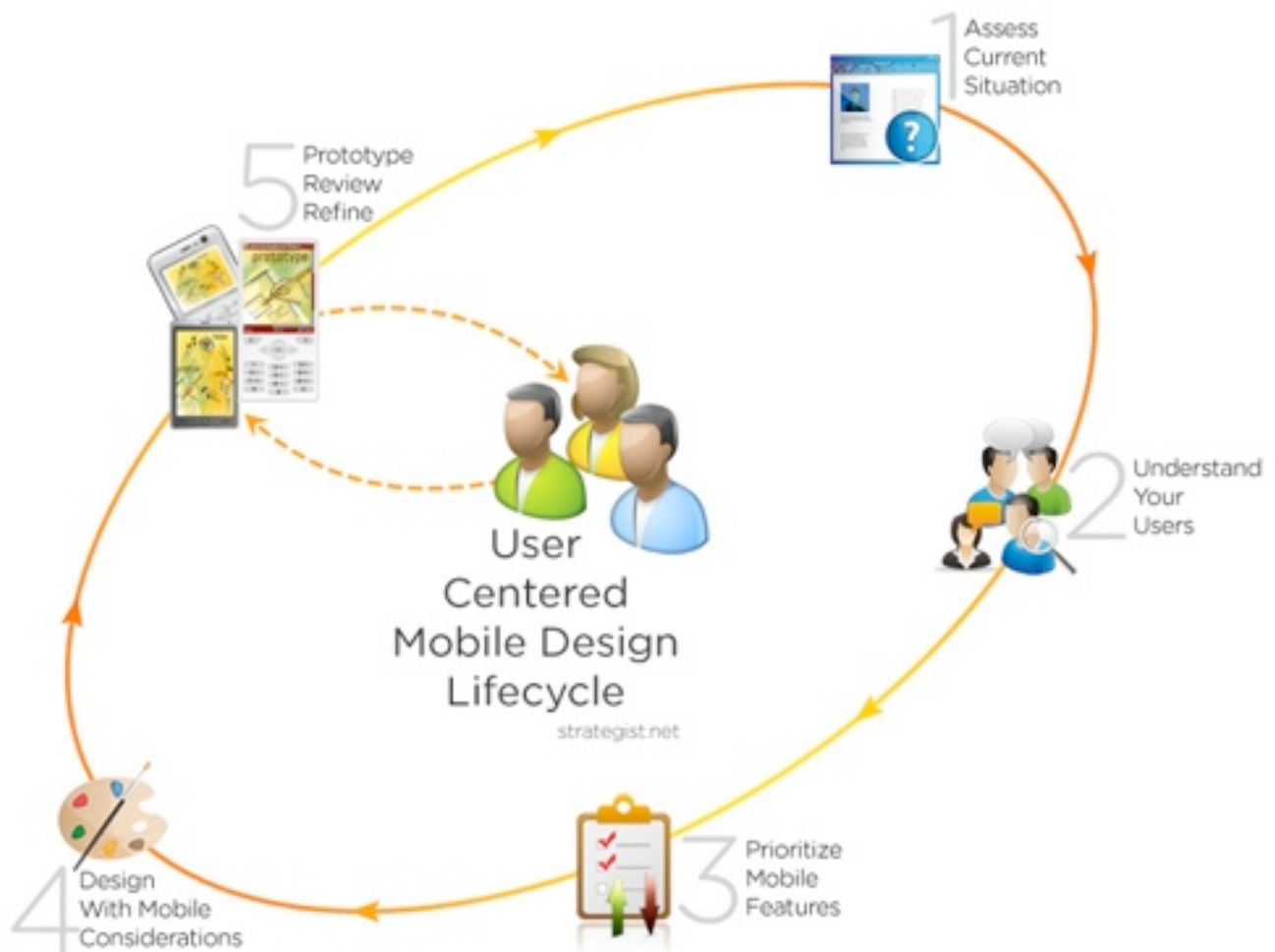
Lyndon Cerejo

For the past few years, we've heard pundits declaring each year as "year of the mobile Web"; each year trying to sound more convincing than the previous. Whether 2011 will be the real "year of the mobile" remains to be seen, but what is indisputable is the fact that the mobile usage of the Web is growing and evolving. As it evolves, so does the mobile user experience, driven by advances in mobile device technology — from better browsers on basic mobile phones (or feature phones — remember the Motorola RAZR?) to the increased adoption of smartphones and tablets.

The term "Mobile Web" (although often [criticized](#)) is commonly used to describe accessing the internet using a mobile device. This definition is broad enough to cover everything from using a browser on a feature phone, to using highly customized apps on smartphones or tablets. "There's an app for that" has made device-specific applications the rage of the day, with some companies starting off backwards with "we need an iPhone app" instead of first understanding *what* their users actually need when they are mobile, the devices that they use, and then deciding the best approach for going mobile, which may not be an app, but could be a mobile website instead. Mobile websites are universally accessible, less expensive to develop and maintain, and can be searched and accessed by most mobile phones.

(The term “Mobile Web” is criticized because it implies that there are “different” Webs which just isn’t true — there is no Desktop Web, for example. It makes more sense to speak of the websites optimized for users accessing those websites through mobile devices. We will be using this perspective in this article. — Smashing Editorial Team)

This article focuses on designing the user experience for mobile websites accessed from mobile phones with small screens, though the process can be applied to building apps as well. As a Web designer, the good news is that the process is similar to designing desktop websites — with some additional mobile-only considerations that go hand-in-hand with small screens, device features and constraints, and connectivity issues. The user-centered mobile design life cycle can be thought of as an ongoing process as shown below:



The ongoing user-centered mobile design life cycle

Let's discuss each phase of this life cycle in more detail.

1. Assess Current Situation: Do You Really Need A Mobile Website Now?

Silly as this may sound in an article about creating mobile website user experiences, it is important to first figure out whether you actually need a mobile website. True, there will be 91.4 million mobile internet users in the US by the end of this year, but how many of them are in your target audience? Mobile commerce sales in the US in 2010 were \$3.4 billion, but if you are not selling anything online, does that matter to you? The more relevant statistic is how many of your users are accessing your website using mobile devices. A quick way to find out is of course by looking at your existing desktop website analytics to identify the percentage of mobile visitors, along with the types of devices and operating systems they use to access your full desktop site. Dig deeper to understand why these users visit your site using a mobile device — what they are trying to do, and the content and functionality they are using.

Now, think about what else would be relevant to these mobile users, and for some competitive insight (and possibly inspiration), take a look at what similar sites are doing with their mobile presence. Your desktop site was created to support some business goals — it may have been a simple goal of creating awareness or a more complex goal of generating revenue through online sales. How will a mobile website complement that existing website? Identify the content and functionality that will be useful for your mobile users while supporting your business goals, including any new goals for mobile. Save these “business requirements” — we’ll need them when deciding what goes on the mobile website.

If, at the end of this exercise, you realize that you have very few mobile users, and they occasionally use just a couple of features (like finding your phone number, or hours of operation), you may be better off for now just optimizing your desktop site to make that information easily accessible by mobile users instead of building and maintaining a separate mobile site. If your website happens to be running on WordPress, there are plugins that can easily mobile-enable that website with little effort.

Not all websites need to go mobile now. Companies that need to extend their core services to their users (like those in travel and banking) certainly do, but a manufacturer of commercial jetliners and military aircraft or a provider of specialized industrial gases would probably not need a separate mobile website now, though that may change in a few years. But if you realize that you need a mobile website, let's focus on the reason you need it: your users!

2. User-Centered Mobile Design Starts With The User

User-centered design relies on user involvement throughout the design process, leading to a solution that users will find useful and want to use. To achieve that, you first need to have a clear understanding of your users, grouped into a prioritized set of user groups whose needs can be thought of individually. For a pharmaceutical company, those groups could be patients, healthcare professionals and caregivers, with the first two groups being the primary user groups, and caregivers being a secondary user group with very similar needs to patients. Identifying your key user groups and creating personas will help you design better for your main users, the way BBC did when building their [future mobile strategy](#).



**Mobile
First**



**Addicted
Devotees**



**Social
Animals**



**Mobile
Lifestyle**

BBC's mobile user groups (taken from their [future mobile strategy](#))

To develop a mobile user experience that aligns to the needs and expectations of your targeted users, you must involve representative users and their feedback throughout the process. Direct input from your users through primary research methods like observing users, contextual interviews and focus groups will give you insights about their mobile behavior, including what they want to do on the mobile Web, and when and how they will use it. Primary research will give you answers to questions like:

- Why do they use your site on a mobile device?
- What features are they using?
- What features are crucial for them when mobile?
- What are some sources of frustration?
- What devices do they use to access the mobile Web?

As you build a detailed picture of your users and their usage patterns, supplement it with secondary research about industry and mobile market trends from [Forrester](#), [eMarketer](#), [Nielsen](#) and [comScore](#), [comScore Data Mine](#), [DeviceAtlas](#) and [Opera's State of the Mobile Web](#). Apart from an understanding of your user and their needs, you will also get a better understanding of the types of mobile devices you have to consider while designing.

3. Prioritize What Your Mobile Website Should Offer

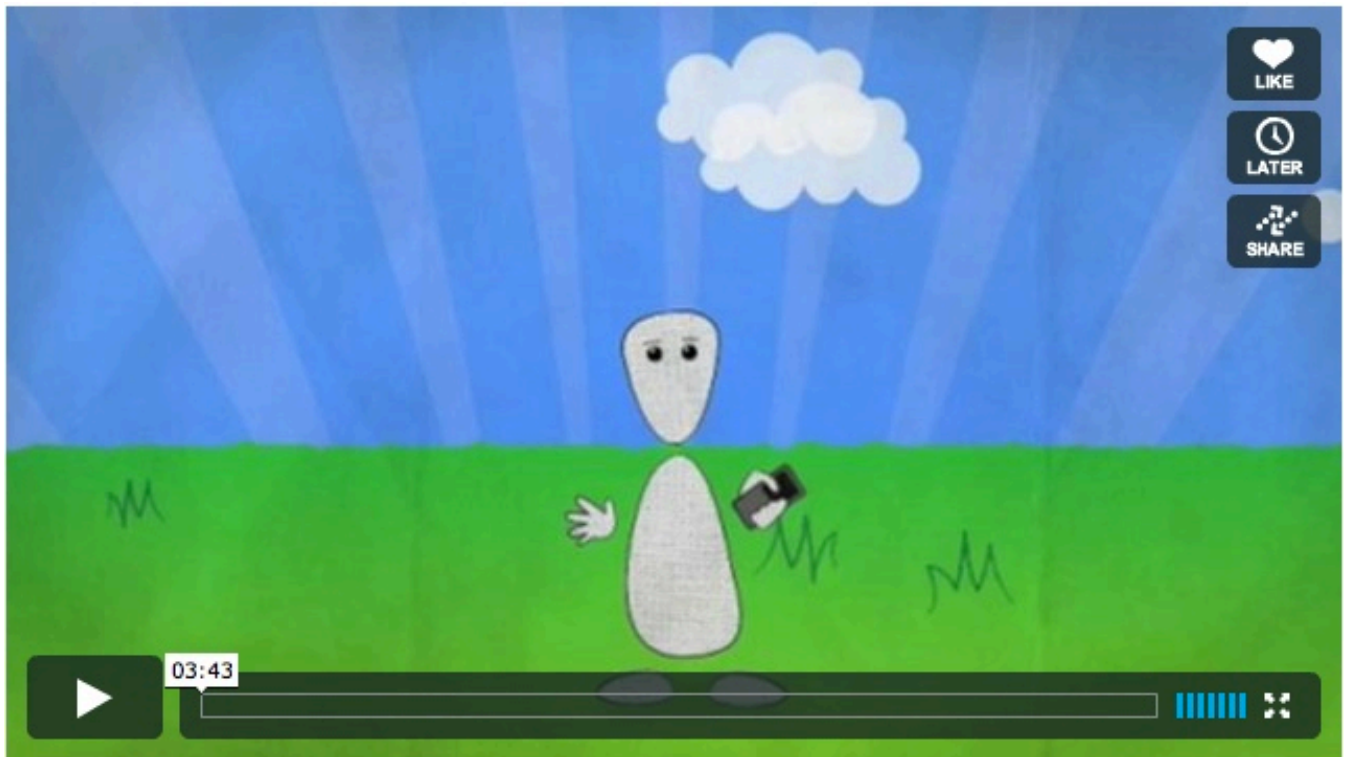
While evaluating the need for a mobile website, you had a list of features you would like to offer on your mobile website. Ideally, these business requirements would align closely with the user requirements identified during user research. If they do not align, look at the requirements asking yourself “what value will this add to the users?” to ensure that your business requirements meet user needs and goals — this is user-centered design after all.

As is often the case, if you end up with more features than you can handle at launch, prioritize what you can initially offer, taking into consideration time, effort, and resources available. Hard as it may be, resist the temptation of trying to build in all the features right from the start. As 37signals so eloquently put it: “[Build half a product, not a half-ass product.](#)”

4. Mobile Design Considerations

Now that the groundwork has been completed, it's time to get to the fun part: actual design! The basic design steps and principles of desktop website design apply to mobile design, with the addition of a few important mobile design considerations. Mobile devices are personal devices with small screens, always turned on, have intermittent slow connections and are mostly used when the user is — you guessed it — mobile.

This brings us to the big “C” of mobile: Context. Mobile users are not captive like computer users are. A user using your desktop website is sitting comfortably, and in a worst case scenario, may be simultaneously listening to music and intermittently tweeting. They are however, not doing all that with one hand, while walking around. Picture a mobile user trying to find directions using a tiny phone with intermittent connectivity, while strap hanging and swaying in a subway train with sub-optimal lighting conditions, deafened by the screeching of wheels on tracks — that gives you some context of use. Simply put, context is about the environment and conditions of usage, including distractions, multitasking, motion, lighting conditions and poor connectivity as you can see in the video below:



Designing for Mobiles: Users & Context from Niek Dekker on Vimeo.

In *Tapworthy – Designing Great iPhone Apps*, author Josh Clark boils down the mobile user’s mindset to one of three possibilities:

- **Microtasking**: Using the phone for short bursts of activity.
- **Local** : Finding out what’s around the user.
- **Bored** : Using the phone for distraction/entertainment.

Keeping all these factors in mind, here are some specific mobile design considerations to pay attention to while designing for the mobile Web:

DESIGN FOR SMALLER SCREEN SIZES



From a designer's big screen to smaller laptop and mobile screens

The most visible difference between a mobile device and a desktop is the screen size. For years, we have been increasing the minimum screen resolution we design for (remember the days of “optimized for 640 x 480”?). Mobile phone screen sizes have also been increasing, but even the gorgeous screen of the iPhone 4 is still small in comparison to a standard 1024×768 desktop design (designed by someone using a 2560×1440 display). This gets more complicated when you factor in the range of screen sizes used by your mobile users. And let’s not forget that some smartphones can change orientation and their users’ expect the website to resize accordingly. Even though many smartphone browsers today can miniaturize desktop websites, they inadvertently break the user experience by making users zoom in and out, which brings us to the traditional approach of creating a separate website for mobile devices.

Bryan Rieger addresses the issue of [designing for multiple screen sizes](#) with a 4-step process:

- Define device groups by grouping screens with similar widths together resulting in a manageable number of groups,
- Create a default reference design that will adapt to smaller and larger screens,
- Define rules for content and design adaptation for it to display well and
- Opt for Web standards and a flexible layout.

While you should ideally be designing for mobile devices used by your users, for a more generic list of browsers to support, follow Peter-Paul Koch’s recommendations of supporting Safari iPhone, Opera Mini, Android WebKit, BlackBerry (WebKit & older for US), Nokia WebKit (Europe).

The other approach advocates a single website that caters to all devices, mobile or not, based on Luke W's [Mobile First](#) principle (see [presentation](#)). Mobile First starts with a design for mobile, which can then be [progressively enhanced](#) for desktop websites. To see it in action, take a look at Yiibu's [proof of concept site](#) based on Mobile First.

Selecting your design approach is not a black and white option. Think about which will work better for your situation. Irrespective of the approach you select, there still are other considerations that go into building a mobile Web.

SIMPLIFY NAVIGATION

Without a mouse to point and click, mobile users have to rely on tiny keypads, trackballs and touch to navigate mobile websites. Add in the small screen with the need to complete tasks quickly and efficiently, and clear and intuitive navigation becomes crucial:

- A common approach that works for the start pages of most sites is a list of links to main features and content, prioritized based on user needs. These often end up being presented vertically instead of the horizontal model used on desktop websites.
- Reduce the number of categories and levels of navigation, and rearrange based on priority, presenting the most important categories first.
- Use clear, concise and consistent labels for navigation across the site.
- Provide access key shortcuts for feature phones, so users can enter a keypad number (0-9) to quickly access a link (AA and CNN examples below).

- When designing for touch, make sure the tap size (width or height) for the navigation item is at least 30 pixels.
- Make links obvious, and provide clear and immediate visual feedback to show the selected link.
- When displaying the main navigation links on internal pages, put them at the bottom instead of the top, to avoid users having to tab through all of them to get to the main content on the screen (CNN example below). If not displaying all the links, Southwest's approach of listing key links (Home, Air Reservations and Flight Check In) on secondary screens works well.
- At the bottom of the mobile homepage, offer a link to the full site so users can switch over to the desktop site, and vice versa.
- Breadcrumbs are not usually not used on mobile sites since navigation is not usually so deep that users need a trail back.



American Airlines, CNN, and Southwest simplify mobile navigation

PRIORITIZE CONTENT

Be succinct. Smaller screen sizes require even more careful attention to the content displayed to the user. Put on your editor's hat and cut unnecessary content, then cut some more. When you're done, prioritize the content and display the most important content first.



desktop site @ 12%



mobile site @ 100%

Amazon's mobile site prioritizes what it offers users on the homepage

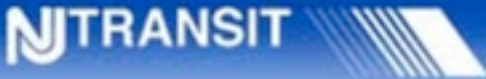
MINIMIZE USER INPUT



Having used tiny numeric keypads and touchscreens on these Casio DataBank watches for over 20 years (yes, I admit it), I know the pain involved with entering data on miniscule screens. Feature phones have tedious numeric keypads for input, while smartphones have tiny keyboards, either real or virtual, which are subject to fat-finger errors.

- Keep your URL as short as possible, keeping in mind, feature phone users have to repeatedly press a key for most alphabets (key presses for “com” are 222-666-6). Follow URL naming conventions that users are used to, like m.site.com or mobile.site.com.
- Use alternate input mechanisms where possible. While apps can use device capabilities for input including motion, camera, gyroscope, and voice, mobile websites are slowly starting to use some of these features, like geolocation.

- Limit input to essential fields. For instance, if registration is required, limit it to the minimum required fields, and use shorter alternatives like a zip code instead of city and state. Volkswagon mobile captures geolocation, but does not use it when trying to schedule a test drive. Not only that, the mobile form is longer and more tedious than the desktop one.
- Select the best mobile input option (e.g. allowing users to select from a list of options is often faster than typing).
- Use smart default values. NJ Transit's mobile website remembers your last selections and defaults the date to today's date.
- When users need to log in, offer the option to stay signed in, especially since mobile devices are personal and not usually shared.



Service Nearby

Enter Location :

OR

© 2010 NJ TRANSIT, All Rights Reserved

Geolocation



Train Schedules & Fares

Origin Station :

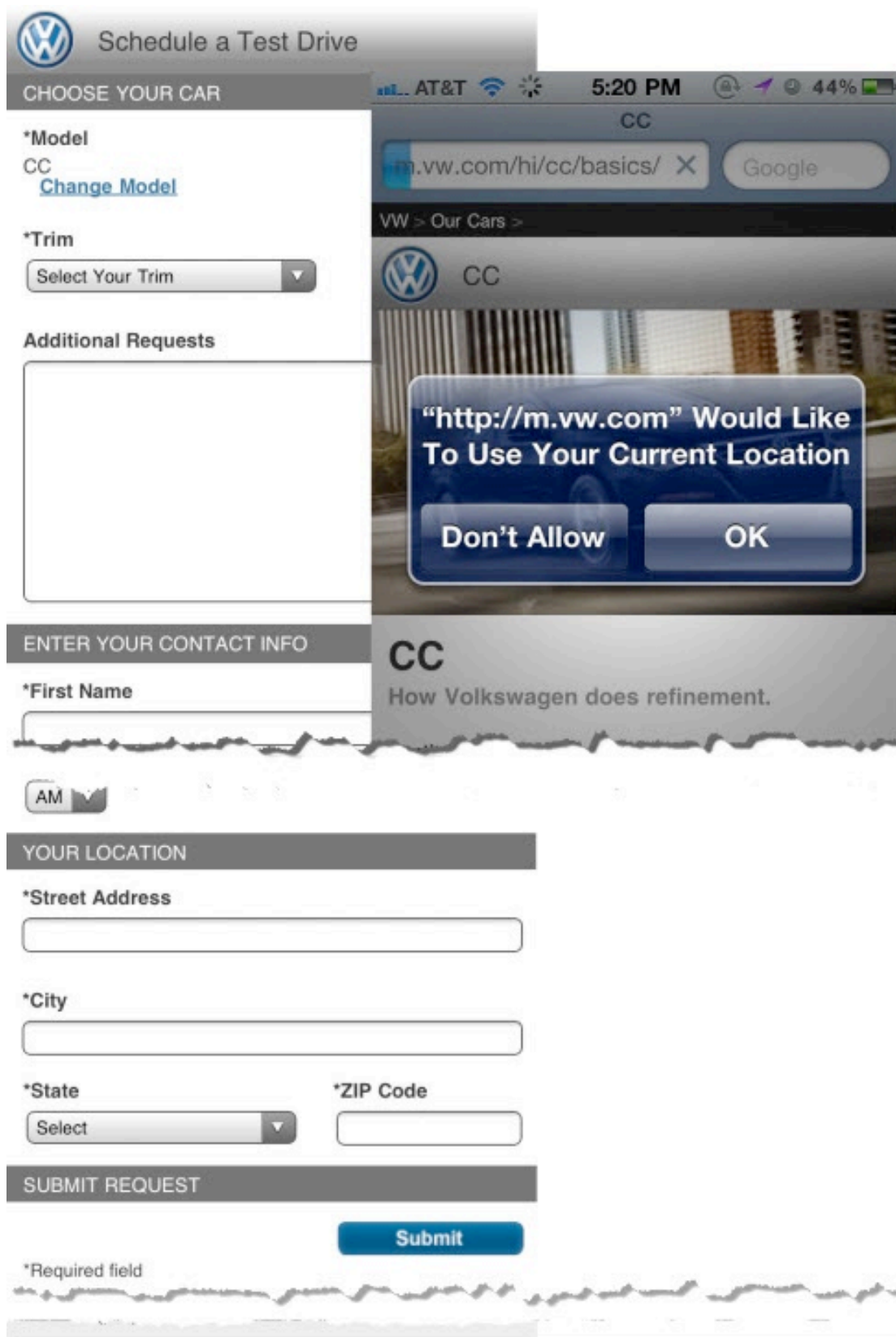
Destination Station :

Date of Travel :

© 2010 NJ TRANSIT, All Rights Reserved

Smart defaults

NJ Transit uses geolocation, remembers selections and defaults to today's date



VW asks for location, but fails to use it (the mobile form is also longer than the desktop version). [See larger image](#)

DESIGN FOR INTERMITTENT CONNECTIVITY

Even with mobile service providers rolling out faster G-speeds, mobile connectivity remains intermittent and does not even approach broadband speeds we are so used to on our wired and wireless connections (not even [Justin Bieber's 6G phone!](#)). Users pay for internet access on their phones, and not everyone has unlimited data plans, so as mobile designers, we should think small when designing for mobile:

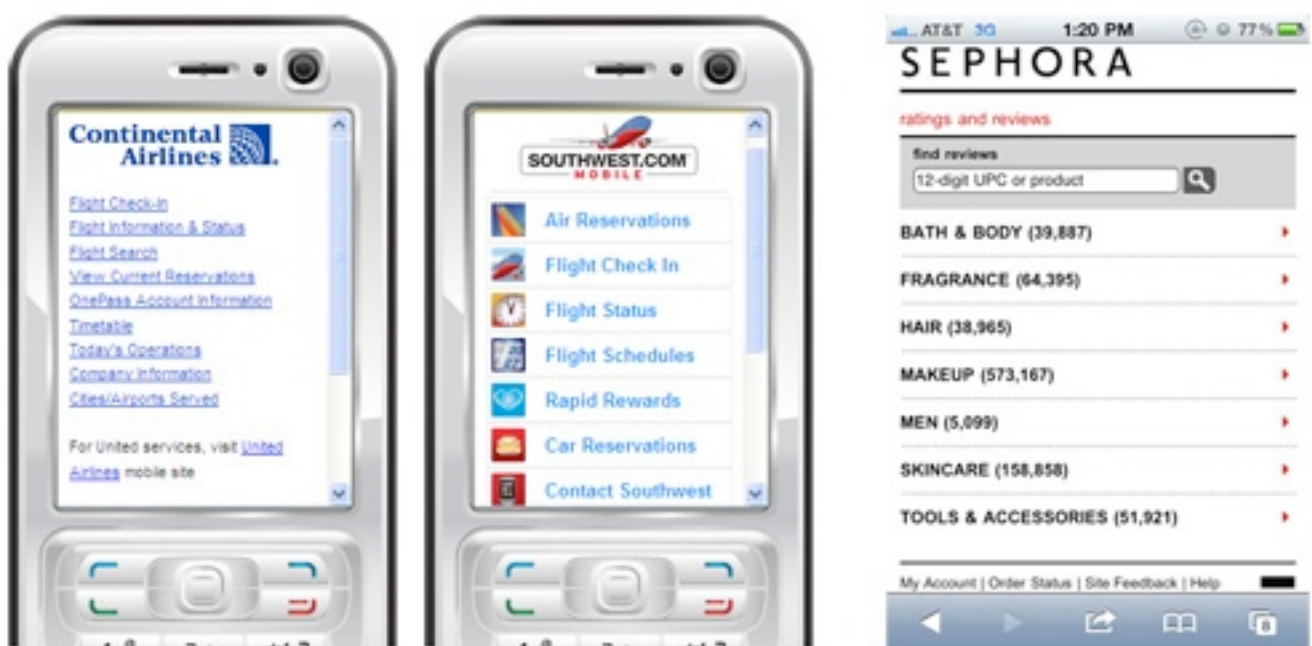
- Keep page sizes small so they can load quickly and are within the memory limitations of most phones.
- Remove unnecessary code, comments and optional tags.
- Reduce images to sizes and resolutions appropriate for mobile devices.
- Minimize the number of embedded images to reduce the number of HTTP requests and to speed up page load time.

OFFER CROSS-CHANNEL CONSISTENCY & INTEGRATION

When you create a mobile website, you cross over into multi-channel territory, and with it, the need to maintain a consistent and integrated user experience across channels.

- Balance form and function. [Continental Airlines](#) takes a minimalistic, no-frills approach, using just their logo for visual branding, while [Southwest](#) makes their mobile website a little more visual with color and icons.

- Maintain continuity. By signing in on the Amazon mobile website, users can view and manage their shopping cart, wishlists and view and track orders, just as they would on the full site. Another good example in the making is [Kindle for the web](#) which, when launched, would allow you to continue reading a Kindle ebook where you left off, irrespective of the device you last used. It would also synchronize your library, bookmarks, notes, and highlights.
- Extend the user experience. Sephora mobile users can look up user reviews and ratings of products (by UPC or name) when trying to decide between two or more products in-store.
- Build a consistent user experience. Citibank users going from the full site, to the smartphone website or an iPhone app have the same user experience across all these channels (though some feature phone users cannot use the site at all).

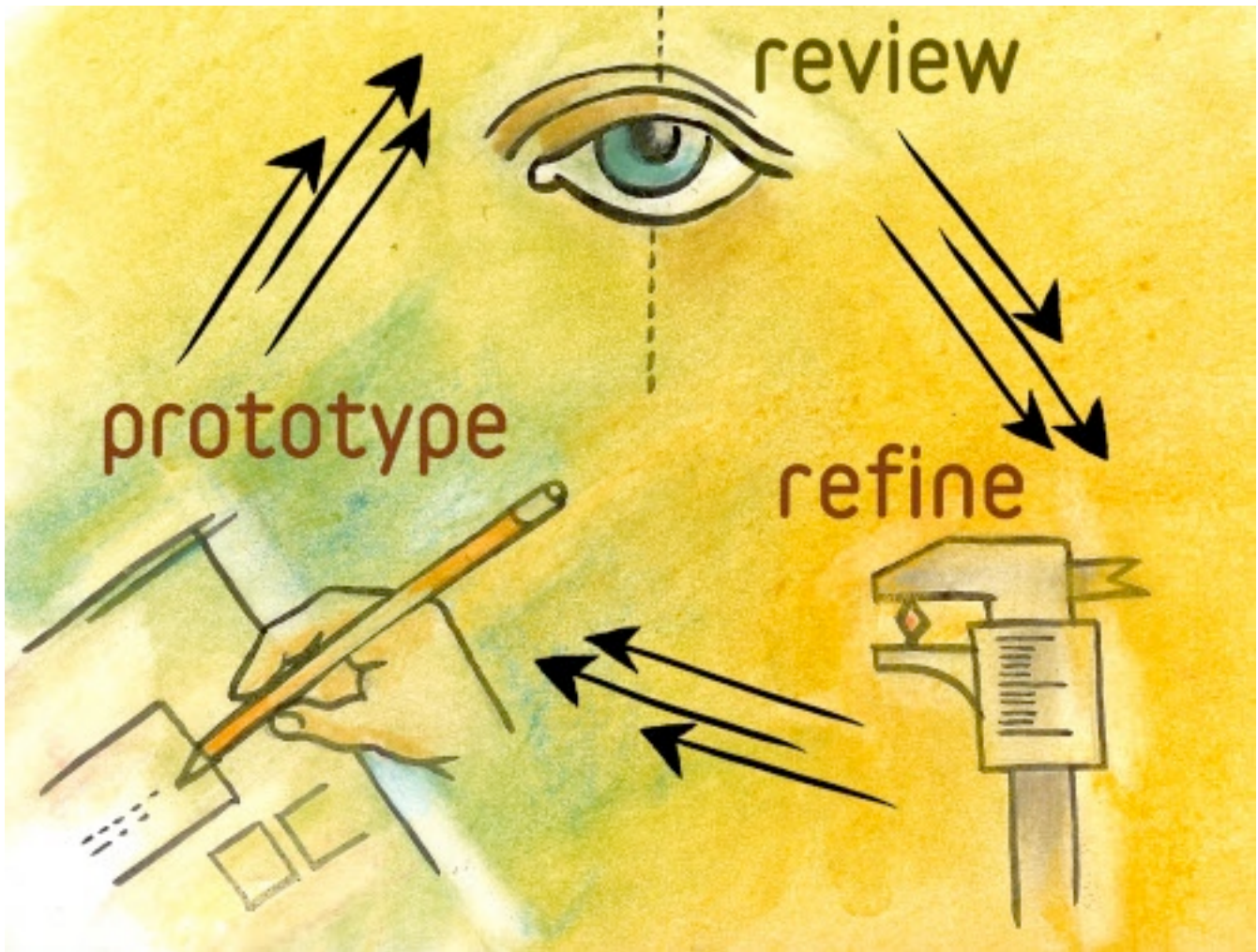


Balancing form and function; Sephora extends the experience for the mobile context

OTHER CONSIDERATIONS

- Detect if your users are using a mobile browser and [direct them to the mobile website](#) (with a link on that screen to switch to the full site).
- Do not rely on technology that is not universally supported by your audience devices like Java, JavaScript, cookies, Flash, frames, pop-ups and auto refreshes.
- If you must make the user scroll, make them scroll in only one direction (most sites scroll vertically).
- Use short, descriptive titles for your pages for easy bookmarking and recall.
- While there are advocates for creating separate mobile sites for feature phones, smartphones and iphones (yes, their own website), others have gone down this road and are opting for a single mobile site. Facebook, one of the most visited mobile properties, recently decided to [unify its mobile websites into one interface](#).

5. Prototype Review & Refine



The rapid prototyping process (from Design Better And Faster With Rapid Prototyping)

Prototyping is important in an iterative user-centered design process, allowing designers to quickly visualize parts of the website, review with users and refine the design. Prototype early and often through the design process, starting with low fidelity paper sketches, refining them into medium fidelity wireframes and subsequently into high fidelity designs, continually testing with users and iterating based on their feedback.

Double check your code for compliance by validating your mobile site using the [W3C mobileOK Checker](#) or [mobiReady](#) to uncover and fix potential issues your users may face. [Mobile emulators](#) and simulators are useful during development, but nothing beats testing your site on actual devices that your users are using.

Ongoing Improvements

Your job is not done when you launch your mobile site — you’ve just come full circle. It’s time to monitor site usage and user feedback to identify changes and requests for new features, in addition to any features that did not make it in the first release. Periodically evaluate your mobile site against competitors using a scorecard (sample from Yankee Group [[pdf](#)]) to identify potential areas for improvement.

In Conclusion

The mobile Web is different, and may be daunting, but the design process is similar enough for Web designers to ease into mobile design. Designing the perfect mobile website is an impossible task even for experts, but using an iterative user-centered design process can help create the best experience for our users with their help.

Mobile Auto-Suggest on Steroids: Tap-Ahead Design Pattern

Smashing Editorial

In contrast to desktop Web search, auto-suggest on mobile devices is subject to two additional limitations: typing avoidance and slower bandwidth. The new patent-pending design pattern, **Tap-Ahead**, uses continuous refinement to create an intuitive, authentically mobile auto-suggest solution. This helps dramatically reduce the amount of typing needed to enter queries, and utilizes slower mobile bandwidth in the most efficient manner. Using this novel design pattern, your customers can quickly access thousands of popular search term combinations by typing just a few initial characters.

Auto-Suggest: Mobile vs. Desktop Web

As John Ferrara wrote in his November 2010 UXMagazine article, [“Psychic Search: a quick primer on search suggestions”](#), today auto-suggest is practically ubiquitous in desktop Web search. In contrast to desktop Web, auto-suggest on mobile is (at least for now) fairly rare. The only mobile Website that currently implements auto-suggest is Google.com, and a handful of mobile auto-suggest implementations that currently exist come from native mobile apps built by leading online retailers like Amazon and Booking.com.

Mobile auto-suggest is non-trivial and quite expensive to implement, but even a large investment does not guarantee a good experience on the mobile device. In many cases, it is not enough to simply transfer the existing

successful desktop Web implementation of the auto-suggest to mobile space. Why not? Our recent study revealed that mobile space is subject to two unique limitations that affect customers' expectations and their use of the auto-suggest feature:

- **Typing Avoidance**

Typing on the mobile keyboard is much harder and more error prone than typing on the full-size desktop Web keyboard. Most people prefer to search using only a few characters — the fewer, the better.

- **Slower Bandwidth**

Mobile signal strength is unpredictable, as is the speed of the Internet connection. Yet the customer expectation is often shaped by their broadband desktop Web experience. Mobile auto-suggest interface must be optimized for slower bandwidth.

The Limitations of the Typical Mobile Auto-Suggest Flow

As I wrote in my UXmatters article, [“Mobile Search: Turning Limitations into Opportunities”](#), mobile phones are notoriously difficult to type on and their Internet connection is often spotty at best. This is especially true in the mobile context of use — that is when the customer is being jostled and bounced around in the moving taxi or metro. In a July 2009 blog post on *Alertbox*, Jakob Nielsen called the mobile experience “miserable” and reported, “Text entry is particularly slow and littered with typos, even on devices with dedicated mini-keyboards.”

Although 3G networks are finally becoming more commonplace, the average speeds US users experience on mobile devices are sometimes as low as one-quarter of the average speeds advertised, according to the Federal Communication Commission (FCC). This implies download speeds of 100-500 Kbps or lower, compared to the speeds of 1 to 1.5Mbps under ideal conditions.

As shown in *Figure 1* below, the difficulty of typing coupled with frequently spotty download speeds of mobile context of use introduce some challenges into the typical auto-suggest process:

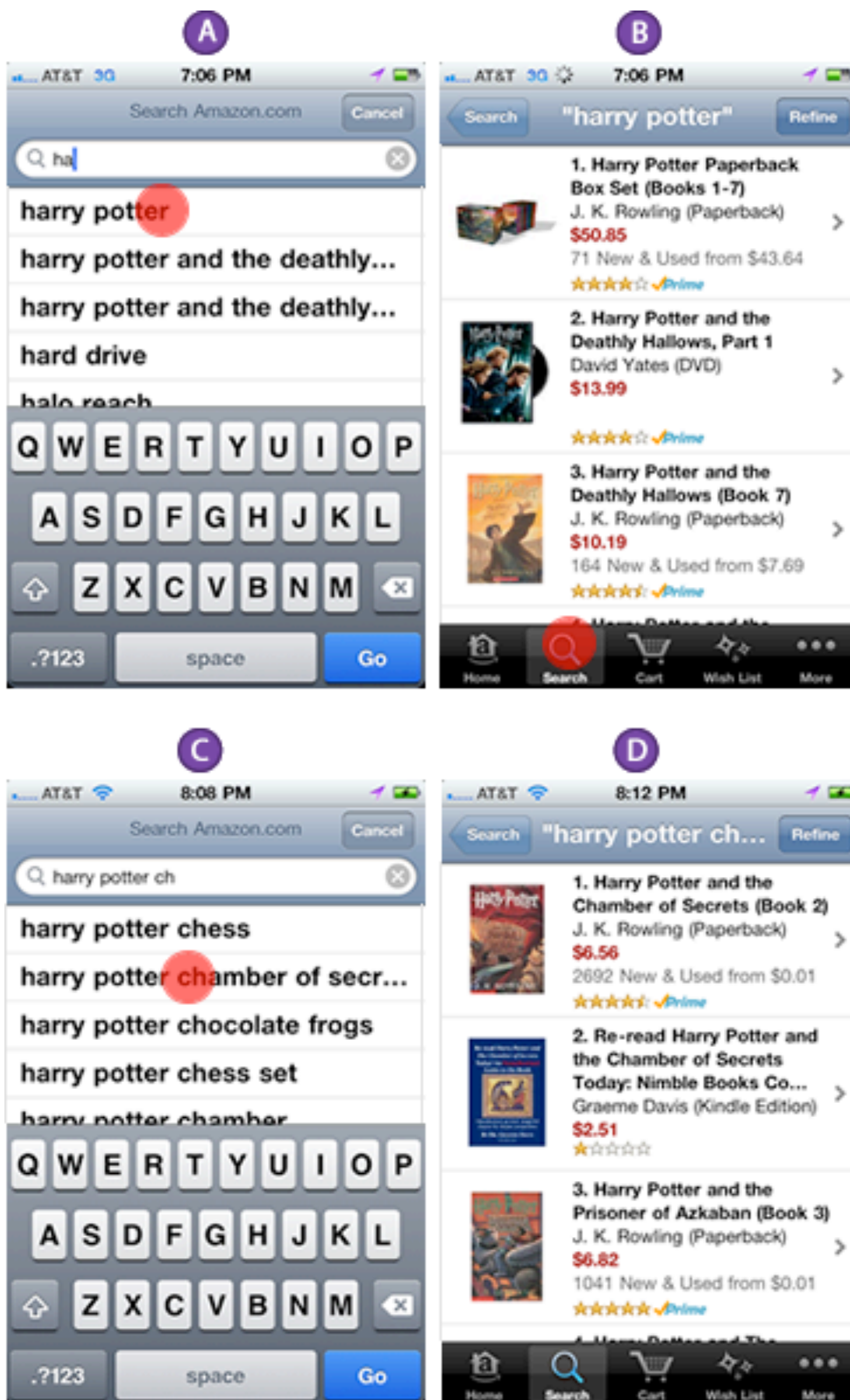


Figure 1. Multi-step auto-suggest search on Amazon iPhone app.

In the example above, the customer (let's call her Anna) is looking for a book called "Harry Potter and The Chamber of Secrets". To begin the search process, Anna types in the first two letters "ha". Using these first letters of the query, the auto-suggest function performs a call to the keywords server, retrieving most frequently used keywords that begin with "ha". The keywords server then quickly returns with a populated auto-suggest layer shown in 1-A, that helpfully suggests "Harry Potter", along with nine other likely queries.

Although the "Harry Potter" does not completely match the query Anna is looking for, it gets her part of the way there and saves a lot of typing. Thus, Anna selects the system recommendation, causing her original query "ha" to be replaced by "Harry Potter". The system then performs a search against the product server, returning up to 50 actual products along with product descriptions, thumbnails, and other pertinent information, as shown in 1-B.

With a fast Internet connection available on the desktop Web, the difference between hitting the keyword server and the products server is negligible — both come back almost as quickly. On the slower mobile connection, however, the difference is not only noticeable, but actually quite annoying because Anna never actually wanted to view "Harry Potter" products, but instead used this auto-suggest query as an interstitial search page — a jumping off point on the way from "ha" to "Harry Potter and The Chamber of Secrets". The only reason why the interstitial search results page shown in 1-B was loaded was to avoid typing the full query on the mobile device.

After the products finally load, Anna again taps the search box to recall the keyboard and adds the letters “ch” to the query, creating the new query “Harry Potter ch”. The auto-suggest again goes to work, this time serving up as a suggestion what looks like the entire query Anna is actually looking for, “Harry Potter and The Chamber of Secr...” as shown in 1-C. Anna taps the suggestion, and the system finally serves up the second search results page, 1-D — the search results page she was originally looking for.

The first search results page is not just annoying and unnecessary — it distorts and pollutes an important asset, the frequently used queries database. The increased frequency with which the query “Harry Potter” is executed in fact helps push it to the top of the most frequently used query list again and again, creating a negative feedback loop in the frequently used queries server. The more something is selected as a jumping off page, the more the interstitial query (and it’s accompanied search results) appears to rise in popularity. Avoidance of typing in conjunction with a slower bandwidth available on mobile devices results in an overall sub-par experience.

Fortunately, there is a better way: *Tap-Ahead Auto-Suggest* design pattern that avoids the need to load the interstitial results page and all of the associated problems. I created *Tap-Ahead* based on my user research specifically to handle typing avoidance and slower bandwidth and optimize the search experience for the way customers use auto-suggest on mobile devices.

Tap-Ahead: A Novel Way of Resolving Typing Avoidance and Slower Bandwidth

Typing avoidance and slower bandwidth are two limitations inherent in mobile devices. Together, these two forces shape how people behave when they search. Tap-Ahead design pattern converts these mobile limitations into opportunities to create a better experience by minimizing the amount of typing and maximizing the use of the limited bandwidth.

The idea for the tap-ahead is simple: avoid serving the interstitial search results page by giving customers a way to narrow their search query using popular keywords without typing. To implement the additional narrow down functionality, I used the established iOS “more actions” icon — a blue circle with an arrow that was familiar to most iPhone users because of its prominence in the Contacts application, shown in Figure 2:

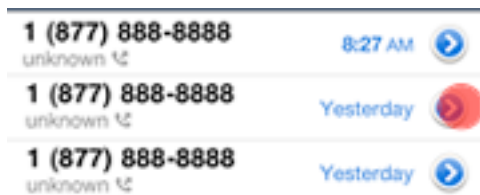


Figure 2. Blue circle with an arrow is used to indicate “more actions” in the iPhone Contacts app.

Of course, the same pattern can be applied on other platforms such as Android, Palm, BlackBerry and Windows 7 Mobile, by replacing the blue iOS arrow with the native platform's standard "more actions" icon. Figure 3 shows what an implementation of the Tap-Ahead on Android might look like:

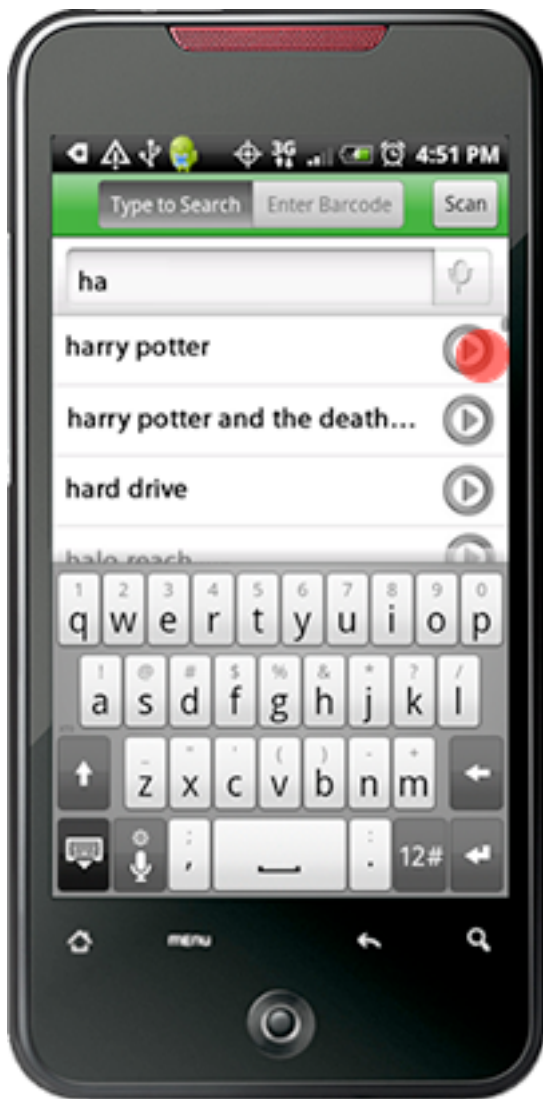


Figure 3: One Possible Android Tap-Ahead implementation.

Let me show you how this feature works in the context of auto-suggest. In this example, the customer (let's call him Ben) is again looking for “Harry Potter and The Chamber of Secrets”, but in contrast to Anna who we followed in the example above, Ben is using the Tap-Ahead auto-suggest interface. Figure 4 shows how this search would proceed using the Tap-Ahead design pattern instead:

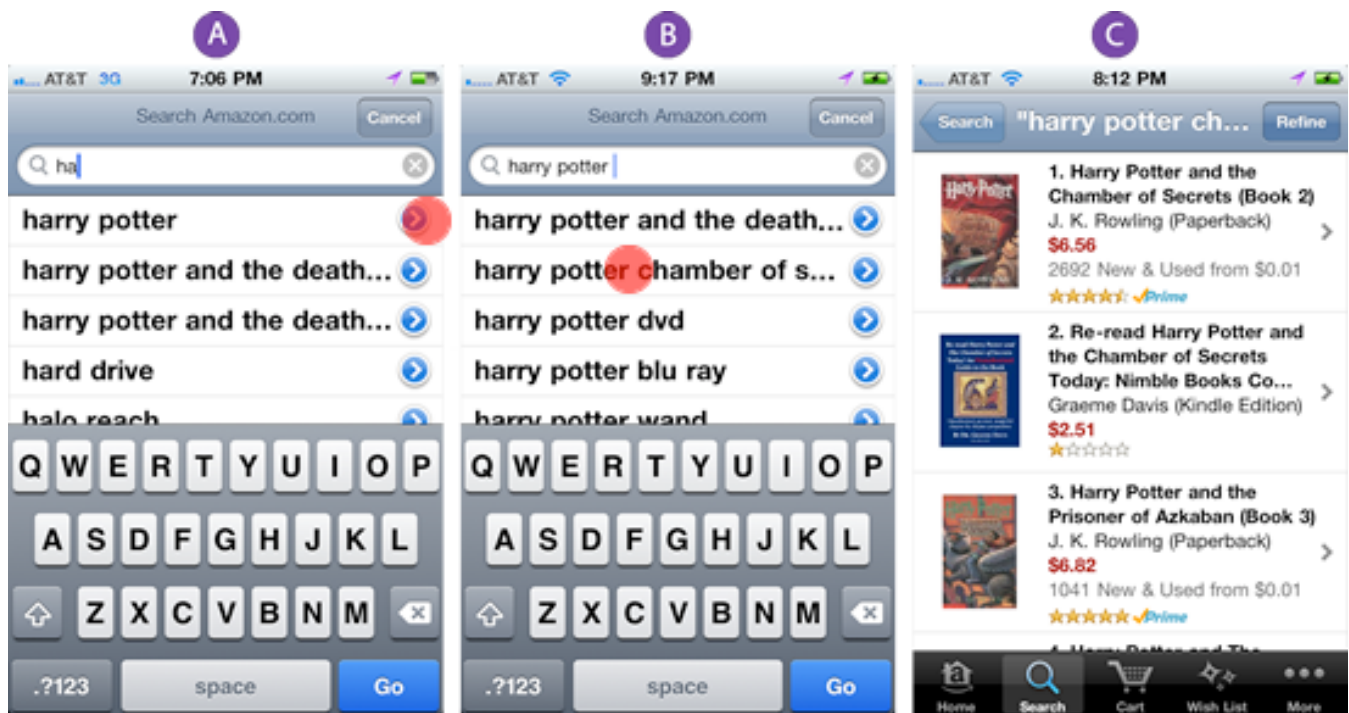


Figure 4: Auto-Suggest Search Process Optimized with Tap-Ahead.

To begin the search process, Ben also types in “ha” as shown in 4-A. Using the first two letters of the query, the auto-suggest function performs a call to the keywords server, retrieving 10 most frequently used keywords that begin with “ha”, among which is “Harry Potter”. Auto-suggestion “Harry Potter” does not completely match “Harry Potter and The Chamber of Secrets”, so instead of selecting the “Harry Potter” suggestion as Anna did in the example above, Ben hits the blue “narrow query” arrow.

This searches through the keyword server for popular queries that contain the keywords “Harry Potter”, serving up the next auto-suggest layer, which contains “Harry Potter and The Chamber of S...”, along with nine other suggestions, as shown in 4-B. This is the query Ben is looking for, so he taps this suggestion and the system serves up the search results page as shown in 4-C — the actual search results page Ben was originally seeking.

Allowing Ben to narrow down the initial auto-suggestion directly using the blue circle with an arrow offers several key user experience benefits:

- **Faster Search**

As we discussed above, hitting the product server to retrieve interstitial search results is expensive, slow and unnecessary. By tapping the blue circle with an arrow, Ben bypassed the useless interstitial search results page and executed his second query, “Harry Potter” *against* the keyword sever — a much faster process, which also returned useful search suggestions. Ben only had to hit the product server *once*, when he had the right search query.

- **Less Typing**

Ben did not need to type in “ch” to find the popular auto-suggestion that contained his second query, “Harry Potter and The Chamber of Secrets”. Although this is not always going to be the case, quickly serving up the popular keyword suggestions upfront, without forcing the customer to type anything, increases the chances of being able to select the desired query faster.

- **Seamless Flow**

Instead of jumping between the auto-suggest list and search results, the system maintained flow by serving pertinent keywords quickly and remaining in the auto-suggest mode until the entire desired query has been entered. This optimized user’s attention on task and maintained flow.

- **Flexibility**

At any point, the customer retained the ability to select the keyword suggestions in a traditional manner or type into the search box or exit the auto-suggest flow. The new mechanism of tapping the blue circle with an arrow to narrow down the search is merely an optional feature that provided additional functionality, allowing the customer to enter his desired query faster and easier.

- **Database Integrity**

Because the interstitial query “Harry Potter” was never actually executed against the product server, it did not “accidentally” count toward the popularity of this query. “Harry Potter and The Chamber of Secrets” was the only query executed against the product server and therefore the only one that counted as a legitimate hit, preserving the integrity of the keyword popularity database.

In our quick usability testing, we found the technique of tap-ahead to be both intuitive and useful. I theorized that this was in part because tap-ahead takes advantage of how people already use the auto-suggest functionality on the mobile device, so the entire process seemed natural and intuitive to our participants. Also, many people remarked that tap-ahead design pattern seemed somehow already familiar. This was because it did not require people to learn anything new: the design uses the established iOS “more actions” icon that most iPhone users already tap several times a day when they use the *Contacts* application.

Although tap-ahead is very useful when combined with the traditional auto-suggest database, its real power comes from redefining the way auto-suggest is used in the context of a mobile device.

Tap-Ahead: From One-Shot to Step-Wise Refinement

Typical auto-suggest on the desktop Web is structured around a one-shot approach: when the customer types in the query, the auto-suggest server attempts to bring back the one exact match to the query the customer is trying to type in. Clicking the auto-suggestion replaces the query the user was typing with the one the system recommended. It's meant to be a one-shot deal: one goal, one query, one suggestion, and one set of results. While this is a decent initial model, in practice, we now know that this is not how people really search. As I describe in my book, *“Designing Search: UX Strategies for Ecommerce Success”* (Wiley, 2011), modern-day search is a multi-step process that takes place in multiple contexts, with the customer moving fluidly between keyword searching and browsing, multiple devices, locations, Web sites and social networks.

One-shot refinement is ill suited to this multi-faceted search paradigm, but after long practice, people on the desktop Web have learned to *satisfice*. It helps that the Internet connection is often blazingly fast and feedback in the form of suggestions and results is nearly immediate. Additionally on the desktop Web, it's really not that difficult to type in the query again or delete some parts of the query auto-suggest has over-delivered using the mouse and keyboard after the interstitial search results page is loaded.

In contrast, on mobile, things are very different. Connection speeds are slower and more sporadic. Also, editing a query string on touch phones is quite a bit harder than doing it on the desktop: for example, on the iPhone, the user must tap and hold the finger on one of the query's keywords, then scroll the tiny handles left and right to select just the right number of letters — not a trivial exercise while bouncing around in the moving vehicle or multi-tasking. Android, Palm and BlackBerry mobile devices require similarly awkward query editing acrobatics.

A more usable way of implementing auto-suggest on the mobile device is through step-wise refinement implemented through the Tap-Ahead interface. Instead of trying to guess the entire query the customer is trying to type in and offer the best one-shot replacement, Tap-Ahead design pattern *guides* the auto-suggest interface through the guessing process one word at a time — a much more natural, flexible and robust auto-suggest method, optimized to solve low bandwidth and fat finger issues people experience on mobile devices.

This is how the step-wise refinement Tap-Ahead interface works. Suppose our two customers, Anna and Ben, are both searching for “Harry Connick Jr.” Anna is using a one-shot auto-suggest flow for this query, shown in Figure 5. Ben, on the other hand, is using the new step-wise tap-ahead refinement alternative as shown in Figure 6:

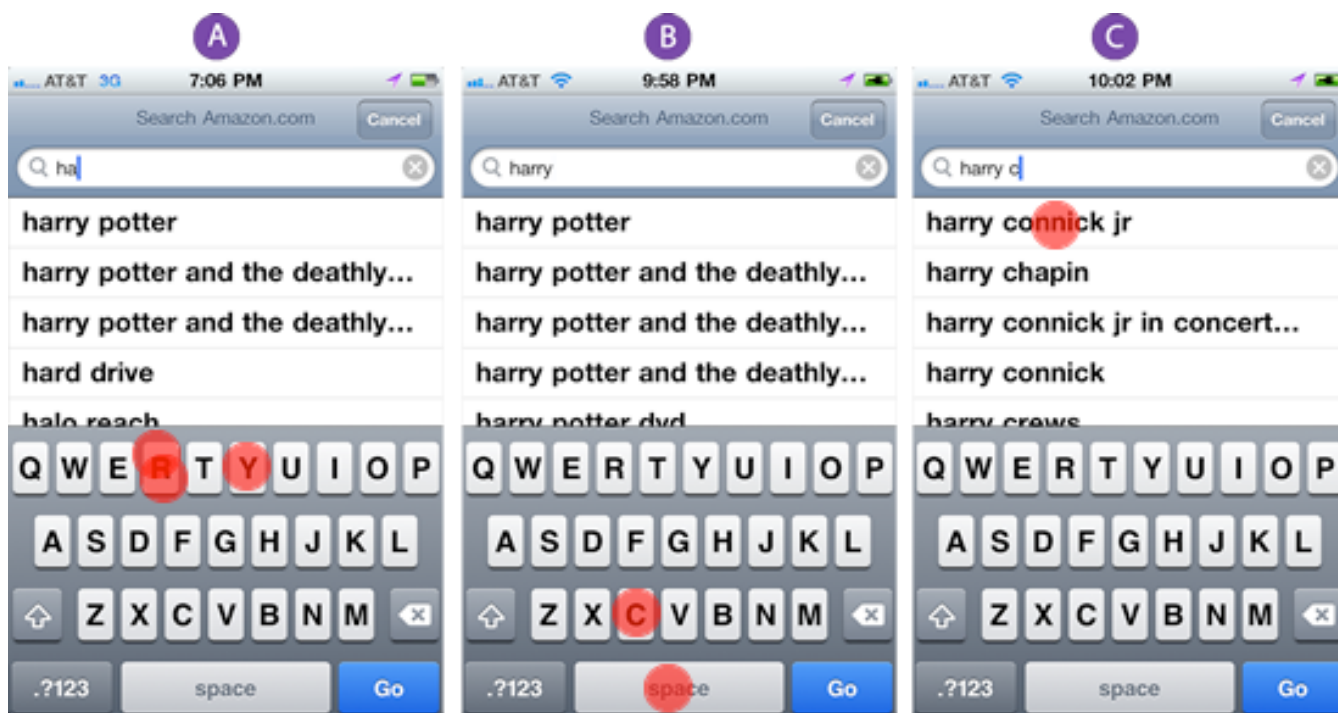


Figure 5: Anna enters “Harry Connick Jr.” using the traditional one-shot auto-suggest flow.

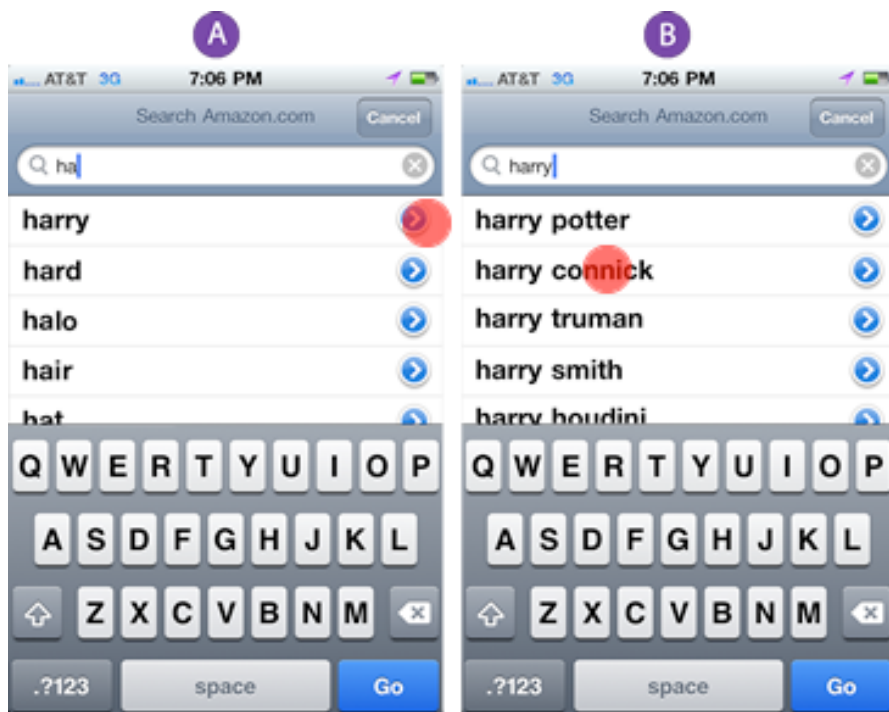


Figure 6: Ben enters “Harry Connick” using a step-wise tap-ahead refinement design pattern.

When Anna types in “ha”, the interface suggests “harry potter”, “hard drive”, “halo reach”, “harry potter and the deathly” and a rather redundant “harry potter and the deathly...” as shown in Figure 5-A. On the other hand, Ben, who is using a step-wise refinement sees a much more humble top 10 one-word suggestions such as “harry”, “hard”, “halo”, “hair” and “hat” shown in Figure 6-A.

Because none of the query terms match the desired query “Harry Connick Jr.” exactly, Anna, who is using the traditional one-shot interface, is forced to keep typing the word “harry”. In contrast, Ben can tap the blue circle with an arrow next to the suggestion “harry”, filling in the entire keyword with *one* tap.

Once both customers enter the keyword “harry”, Anna again sees one-shot auto-suggestions which include “harry potter”, several variations of the “harry potter and the deathly...”, “harry potter dvd”, “harry potter wand” and many other “harry potter” variations, as shown in Figure 5-B. Unfortunately, the set does not include a “harry connick jr.” suggestion, so Anna is again forced to keep typing “c” in order to get the full one-shot auto-suggestion of “harry connick jr.”, shown in Figure 5-C.

In contrast, Ben receives only single keyword suggestions, so his second set of suggestions includes only a *single* instance of the keyword “potter”, which successfully covers *all* of the variations of the query “harry potter”, which had to be listed individually in Anna’s one-shot interface. Thus instead of 10 variations of the “harry potter” query, Ben’s single-word auto-suggestions include a rich set of 10 one-word complements of “harry”: “potter”, “connick”, “truman”, “smith”, “houdini”, “harrison”, “dent”, “david”, “eastwood” and “hendersons”, as shown in Figure 6-B. A one-tap selection selects “connick” which yields the query “harry connick” that is sufficiently close to the desired query “harry connick jr.”. Note that although in this case

it was not needed, the addition of the word “jr.” can be easily accomplished with one more tap on the blue “narrow down” arrow.

To summarize this comparison, after both Anna and Ben typed in the initial “ha”, Ben was able to finish entering the entire query in only 2 easy key-strokes — by selecting two successive auto-suggestions, whereas Anna had to type in the additional “rry c” and select one auto-suggestion, a total of 6 keystrokes. In this quick demo task, tap-ahead interface provided a huge improvement, given how hard and error-prone typing has proven to be on the mobile device.

The advantage of the tap-ahead step-wise refinement interface is that the refinement keywords can be loaded asynchronously for each of the 10 auto-suggestions even while the customer is making the selection of the first keyword. Given that most queries are between two and three keywords long, and each successive auto-suggest layer offers 10 additional keyword suggestions, tap-ahead with step-wise refinement allows customers to reach between 100 ($10 * 10$) and 1,000 ($10 * 10 * 10$) of the top keywords through typing only a few initial characters. Tap-ahead allows the mobile auto-suggest interface to maintain flow and increase speed and responsiveness on tiny screens that is simply not possible to currently achieve with the traditional one-shot auto-suggestion interface.

In Conclusion

I want to close out with this quote from Google, the company that invented the original auto-suggest design pattern, which clearly inspired my tap-ahead design:

“At Google, we often think that speed is the forgotten ‘killer application’ — the ingredient that can differentiate winners from the rest. We know

that the faster we deliver results, the more useful people find our service.”

— Matt Brittin, Managing Director, UK & Ireland Operations, Google

I hope that you find the Tap-Ahead design pattern useful in improving the speed and responsiveness of your own auto-suggest mobile interface and that Tap-Ahead contributes to further experimentation and evolution of search design patterns.

About The Authors

Greg Nudelman

Greg's DesignCaffeine designs stuff that works for today's top companies. Be sure to watch his free webinar: [How to make Agile RITE work for your Mobile Design project.](#)

Twitter: [Follow the author on Twitter](#)

Ivo Weevers

Ivo heads up the talented design team for Ubuntu, the most popular open-source operating system. Ubuntu is used by over 20 million people in over 240 countries in more than 80 languages. Examples are 2000 schools in Andalusia, the French police force, and rural communities in India. The team's goal is to make Ubuntu appealing, usable and accessible on PCs, devices and other media to help technology becoming available for everybody. Ivo has over 12 years of design experience in creating high-quality mobile, web, and product experiences. He has worked in the telecom, social media, banking, travel, entertainment, medical and car industries.

Jen Gordon

Jen Gordon is a web and mobile UI designer based in Atlanta, Georgia. She teaches [iPhone app design](#) and other 7-day design courses at DesignBoost. For fun she enjoys writing about [DIY craft projects](#) and [kids crafts](#) on her personal blog.

Website: [DesignBoost](#)

Twitter: [Follow the author on Twitter](#)

Luke Wroblewski

Most recently [LukeW](#) was the Chief Product Officer (CPO) and co-founder of [Bagcheck](#) which was acquired by Twitter Inc. in 2011. Luke is also the author of the upcoming book [Mobile First](#) and was Chief Design Architect (VP) at Yahoo! Inc.

Website: <http://www.lukew.com/>

Twitter: [Follow the author on Twitter](#)

Lyndon Cerejo

Lyndon Cerejo is a certified [user experience & usability strategist](#) at Capgemini with a successful track record with clients including Allstate, American Express, Coca-Cola, General Motors, Merrill Lynch, and Walmart. His key areas of expertise are user experience analysis, information architecture and rapid prototyping usability testing, online strategy & marketing. He is the co-author of [marketing.com](#) - a book about marketing adaptations on the Internet. Website: [Strategist.Net](#)

Matt Lawson

[Matt Lawson](#) is a passionate web developer, blogger and mountain bike enthusiast living and working in the beautiful city of Bristol, England.

Smashing Editorial

Sven (sl), Vitaly (vf), Iris (il), Jan (jc) and Esther (ea) love high-quality content and care about little details. They also believe that good content and design are crafts worth sharpening.