

Preface

The more things change, the more they stay the same.

We're approaching the 15 year anniversary of the original publication of this book, *Designing Interfaces*. And ten years since the second edition. It's worth looking at what's changed and what hasn't, and what it means for interface design and people who interact with software.

Since then, the big change is that technology and software accelerated their growth and spread in an ever-increasing way. This trend is not stopping. Today, we interact with software in almost every aspect of our daily lives, for work, leisure, communicating, shopping, learning, and more. The list of devices and things with software smarts and Internet connectivity is exploding: Cars, smart speakers, television, toys, watches, homes. Screen sizes and types vary and the prevalence of interfaces that are primarily gesture or voice are found in consumer products. Globally, more than half the population of the planet now accesses the Internet. Finally, software is becoming more powerful, more analytical, more predictive, more able to offer smarter insights and operate more independently. In a phrase, it's becoming more like us.

Interface design, like everything else, struggles to keep up with the changing times. A third edition that tried to be the comprehensive design guide for the possible types of interfaces in this increasing complexity would be enormous and never finished.

Design Patterns Remain Relevant

We asked ourselves, "How is design and *Designing Interfaces* relevant?" The answer is the design patterns. Design patterns come from the ways people perceive and use software. Human senses and psychology don't change, and these patterns work with these, not against them. The patterns are evergreen also because they are based on the tasks—big and small—that people want to do with software. People will always want to use screens to search for things, enter data, create, control or manipulate digital objects, manage money and payments, send and receive information, messages and files to other people. Design patterns form the building blocks for UIs for any screen.

Focus: Screen-based, Web, and Mobile

We chose to focus on screen-based design, for the web and mobile devices since that is the majority of what's out there today. Screens are not going away. There will be more screens. In fact, the complexity of what we need to show on these screens is increasing. This will test our skills as designers and builders even more. We'll always need people to design these interfaces.

We reworked visual design and interaction design chapters to focus on the foundational theories and practices that drive great design. The rest of the book is the discussion of the patterns and how they can be applied.

We've updated the patterns and examples and provided explanations that show how they are relevant today.

We deliberately did not go into a number of newly emerged and still-emerging areas. Not because they're not important, but because they have their own still evolving patterns, and present special design challenges. Already they represent distinct domains of design. Design books that focus on these unique new fields are here now. To go into the specifics of these areas, look for a design book that specializes in that domain.

Voice

We talk to our phones, our cars and our smart music speakers at home to make software work for us. We have conversations with the machine. To find out more about designing for voice, we recommend *Designing Voice User Interfaces* (Pearl, Cathy. *Designing Voice User Interfaces: Principles of Conversational Experiences*. O'Reilly, 2017.)

Social Media

Social media has evolved from being a way for friends and family to stay connected. It is a communication, discussion and interaction layer that is present in almost all software. It has revolutionized business communication and productivity. For more, see *Designing Social Interfaces* (Crumlish, Christian, and Erin Malone. *Designing Social Interfaces: Principles, Patterns, and Practices for Improving the User Experience*. O'Reilly, 2015.)

Streaming digital television

What we call television is now streaming digital video for entertainment on the screen or device of our choice. The interfaces for this are evolving beyond searching

and browsing. TV is an app now, too, with access to all the features and power of our devices. More in Designing Multi-Device Experiences (Levin, Michal. *Designing Multi-Device Experiences: An Ecosystem Approach to User Experiences across Devices*. O'Reilly, 2014.)

Augmented Reality/Virtual Reality/Mixed Reality

Interfaces and software are becoming a layer on top of the physical world or a fully immersive world of its own. Goggles, glasses and other devices are allowing us to mix the digital world in with what we see in front of us. See Creating Augmented and Virtual Realities (Pangilinan, Erin, et al. *Creating Augmented and Virtual Realities: Theory and Practice for Next-Generation Spatial Computing*. O'Reilly, 2019.)

Chatbots and Conversational Design

Software assistants that seem to be human now talk to us every day via voice, messaging and chat. These chatbots understand and respond to a conversation in a highly natural-seeming way. Powered by software that recognizes patterns in data and speech, and then learns and improves, chatbots are able to take over the handling of simple information requests and carrying out basic tasks for almost any business or situation. To achieve this capability, designers must create the source data domain and the conversation frameworks and scenarios that make the chatbot learn and become useful. To learn more about designing bots and conversations, check out Designing Bots: Creating Conversational Experiences (Shevat, Amir. *Designing Bots: Creating Conversational Experiences*. O'Reilly, 2017.)

Natural User Interfaces—Gesture-based interfaces (beyond touch)

This evolving area of design focuses on using the body to interact with technology. Interfaces that one can touch, hold, squeeze or wave at, experiences one can trigger by the movement of the hands, feet or by moving around in space.

Software Is Systems Now

Now more than ever, designers, entrepreneurs and developers, and companies have an effective toolset to design and build great software.

The design and software world has evolved into a systems, components and modules

approach. Starting from scratch to design or code something entirely new is not the norm anymore. There are numerous UI toolkits and frameworks that allow you to create screen-based interfaces that work across many screen sizes quickly. These component libraries should be regarded as a way to get to a solid base quickly. They are not a ceiling for design innovation, they are the floor.

The services and middleware that power software is increasingly an integration of separately owned and operated services as well. Why develop your own registration system when you can use sign up with Google, Facebook, and others? Why develop your own analytics and report-building software when you can integrate your choice of robust, customizable business intelligence platforms? Why code your own mobile apps when you have Amazon Web Services? The same is true for all of your HR processes, all of your IT infrastructure. We increasingly assemble, rather than create *de novo*.

In conclusion, we believe more people than ever are designing and building software. The tools are there. We want a guidebook for this new state of software design that makes it easy to understand and easy to implement. We wanted to write the handbook for web and mobile screen design that we would like to have at hand on our desks, a guide to give to early-career designers, and a reference for product managers, engineers, and executive management. We hope it becomes a useful reference that gives a common vocabulary for designing interfaces.

We see that consumer software experiences are an ever-present part of our lives now. We're spending more time than ever using software interfaces. They deserve to be useful, helpful and make life easier, not harder.

While new modes, devices, and formats are rapidly emerging, screens are with us now and will be with us for a long time. We will be typing, tapping and touching screens to get jobs done and to entertain ourselves, to find something, to buy something, learn something. We hope the principles and examples in this book will give you the knowledge and confidence to use these proven patterns to create great products and services, great design, and great experiences for everyone.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.

Tip

This element signifies a tip or suggestion.

Note

This element signifies a general note.

Warning

This element indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code

from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*Designing Interfaces, Third Edition* by Jenifer Tidwell, Charles Brewer, and Aynne Valencia-Brooks (O'Reilly). Copyright 2019 Jenifer Tidwell, Charles Brewer, and Aynne Valencia-Brooks, 978-1-492-05196-1.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

O'Reilly Online Learning

Note

For almost 40 years, *O'Reilly Media* has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, conferences, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, please visit <http://oreilly.com>.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

- O'Reilly Media, Inc.
- 1005 Gravenstein Highway North
- Sebastopol, CA 95472
- 800-998-9938 (in the United States or Canada)
- 707-829-0515 (international or local)
- 707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at

<http://www.oreilly.com/catalog/0636920254492>.

To comment or ask technical questions about this book, send email to
bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Chapter 1. Designing for People

This book is almost entirely about the look and behavior of applications, web apps, and interactive devices. But this first chapter is the exception to the rule. No screenshots here; no layouts, no navigation, no diagrams, no visuals at all.

Why not? After all, that's probably why you picked up the book in the first place.

It's because good interface design doesn't start with pictures. It starts with an understanding of people: what they're like, why they use a given piece of software, and how they might interact with it. The more you know about them, and the more you empathize with them, the more effectively you can design for them. Software, after all, is merely a means to an end for the people who use it. The better you satisfy those ends, the happier those users will be.

A framework for achieving this is described here. It covers four areas. These are not strict rules or requirements for creating great designs. But having a plan for how you will inform yourself and your team in each area will give you confidence that your work is based on real insights into valuable problems to solve for your target customers. Decide for yourself what level of time and effort is right for your project or company. Revisiting these areas regularly will keep key insights top of mind and help focus everyone's effort, especially UI design, on creating great outcomes for people.

The four part structure for understanding design for people is this:

1. Context: Who is your audience?
2. Goals: What are they trying to do?
3. Research: Ways to understand context and goals
4. The Patterns: Cognition and behavior related to interface design

Context

Know Your Audience

There's a maxim in the field of interface design: "You are not the user!"

So, this chapter will talk about people. It covers a few fundamental ideas briefly in this introduction, and then discusses some patterns that differ from those in the rest of the

book. They describe human behaviors—as opposed to system behaviors—that the software you design may need to support. Software that supports these human behaviors better helps users achieve their goals.

Interactions are conversations

Each time someone uses an application, or any digital product, he carries on a conversation with the machine. It may be literal, as with a command line or phone menu, or tacit, like the “conversation” an artist has with her paints and canvas—the give and take between the craftsperson and the thing being built. With social software, it may even be a conversation by proxy. Whatever the case, the user interface mediates that conversation, helping users achieve whatever ends they had in mind.

The key points are these:

- There are two participants in the conversation, the person and the software.
- There is a constant, back and forth exchange of information.
- The exchange is a series of requests, commands, reception, processing and responses.

The human in the conversation needs continuous feedback from the interface that confirms that things are working normally, inputs are being processed, and that she is proceeding satisfactorily towards the goal of the moment

For this feedback loop to work, the software—which can’t be as spontaneous and responsive as a real human (at least not yet)—should be designed to mimic a conversation partner. It should be understandable to its partner, it should indicate it’s active, when it’s “listening,” and it should be obvious when it’s responding. Another layer on this is having some anticipated next steps or recommendations, in the same way a considerate person might be helpful to another.

As the user interface designer, then, you get to script that conversation, or at least define its terms. And if you’re going to script a conversation, you should understand the human’s side as well as possible. What are the user’s motives and intentions? What “vocabulary” of words, icons, and gestures does the user expect to employ? How can the application set expectations appropriately for the user? How do the user and the machine finally end up communicating meaning to each other?

Match your content and functionality to your audience

Before you start the design process, consider your overall approach. Think about how you might design the interface’s overall interaction style—its personality, if you will.

When you carry on a conversation with someone about a given subject, you adjust what you say according to your understanding of the other person. You might consider how much he cares about the subject, how much he already knows about it, how receptive he is to learning from you, and whether he’s even interested in the conversation in the first place. If you get any of that wrong, bad things happen—he might feel patronized, uninterested, impatient, or utterly baffled.

This analogy leads to some obvious design advice. The subject-specific vocabulary you use in your interface, for instance, should match your users’ level of knowledge; if some users won’t know that vocabulary, give them a way to learn the unfamiliar terms. If they don’t know computers very well, don’t make them use sophisticated widgetry or uncommon interface-design conventions. If their level of interest might be low, respect that, and don’t ask for too much effort for too little reward.

Some of these concerns permeate the whole interface design in subtle ways. For example, do your users expect a short, tightly focused exchange about something very specific, or do they prefer a conversation that’s more of a free-ranging exploration? In other words, how much openness is there in the interface? Too little, and your users feel trapped and unsatisfied; too much, and they stand there paralyzed, not knowing what to do next, unprepared for that level of interaction.

Therefore, you need to choose how much freedom your users have to act arbitrarily. At one end of the scale might be a software installation wizard: the user is carried through it with no opportunity to use anything other than Next, Previous, or Cancel. It’s tightly focused and specific, but quite efficient—and satisfying, to the extent that it works and is quick. At the other end might be an application such as Excel, an “open floorplan” interface that exposes a huge number of features in one place. At any given time, the user has hundreds of things that he could do next, but that’s considered good, because self-directed, skilled users can do a lot with that interface. Again, it’s satisfying, but for entirely different reasons.

Skill Level

1. Designing for People

How well can your audience(s) use your interface now and how much effort are your users willing to spend to learn it?

Some of your customers may use it every day on the job—clearly they'd become an expert user over time. But they will become increasingly unhappy with even small dissatisfactions. Maybe they'll use it sometimes, and learn it only well enough to get by (“Satisficing”). Difficulties in usage may be tolerated more. Maybe they'll only use it once. Be honest: can you expect most users to become intermediates or experts, or will most users remain perpetual beginners?

Software designed for intermediate-to-expert users includes:

- Photoshop
- Excel
- Code development environments
- System-administration tools for web servers

In contrast, here are some things designed for occasional users:

- Kiosks in tourist centers or museums
- Windows or Mac OS controls for setting desktop backgrounds
- Purchase pages for online stores
- Installation wizards
- Automated teller machines

The differences between the two groups are dramatic. Assumptions about users' tool knowledge permeate these interfaces, showing up in their screen-space usage, labeling, and widget sophistication, and in the places where help is (or isn't) offered.

The applications in the first group have lots of complex functionality, but they don't generally walk the user through tasks step by step. They assume users already know what to do, and they optimize for efficient operation, not learnability; they tend to be document-centered or list-driven (with a few being command-line applications). They

1. Designing for People

often have entire books and courses written about them. Their learning curves are steep.

The applications in the second group are the opposite: restrained in functionality but helpful about explaining it along the way. They present simplified interfaces, assuming no prior knowledge of document- or list-centered application styles (e.g., menu bars, multiple selection, etc.). “Wizard” frequently show up, removing attention-focusing responsibility from the user. The key is that users aren’t motivated to work hard at learning these applications—it’s usually just not worth it!

Now that you’ve seen the extremes, look at the applications in the middle of the continuum:

- Microsoft PowerPoint
- Email clients
- Facebook
- Blog-writing tools

The truth is that most applications fall into this middle ground. They need to serve people on both ends adequately—to help new users learn the tool (and satisfy their need for instant gratification), while enabling frequent-user intermediates to get things done smoothly. Their designers probably knew that people wouldn’t take a three-day course to learn an email client. Yet the interfaces hold up under repeated usage. People quickly learn the basics, reach a proficiency level that satisfies them, and don’t bother learning more until they are motivated to do so for specific purposes.

You may someday find yourself in tension between the two ends of this spectrum. Naturally you want people to be able to use your design “out of the box,” but you might also want to support frequent or expert users as much as possible. Find a balance that works for your situation. Organizational patterns in Chapter 2, such as “Multi-Level Help”, can help you serve both constituencies.

Goals: Your interface is just a means to their ends

Everyone who uses a tool—software or otherwise—has a reason for using it. These are their goals. Goals could be outcomes such as:

- Finding some fact or object
- Learning something
- Performing a transaction
- Controlling or monitoring something
- Creating something
- Conversing with other people
- Being entertained

Well-known idioms, user behaviors, and design patterns can support each of these abstract goals. User experience designers have learned, for example, how to help people search through vast amounts of online information for specific facts. They've learned how to present tasks so that it's easy to walk through them. They're learning ways to support the building of documents, illustrations, and code.

Ask Why

The first step in designing an interface is to figure out what its users are really trying to accomplish. Filling out a form, for example, is almost never a goal in and of itself—people only do it because they're trying to buy something online, renew their driver's license, or install software. They're performing some kind of transaction.

Asking the right questions can help you connect user goals to the design process. Users and clients typically speak to you in terms of desired features and solutions, not of needs and problems. When a user or client tells you he wants a certain feature, ask why he wants it—determine his immediate goal. Then to the answer of this question, ask “why” again. And again. Keep asking until you move well beyond the boundaries of the immediate design problem.¹

Design's value: Solve the right problem, then solve it right

Why should you ask these questions if you have clear requirements? Because if you love

1. Designing for People

designing things, it's easy to get caught up in an interesting interface design problem. Maybe you're good at building forms that ask for just the right information, with the right controls, all laid out nicely. But the real art of interface design lies in solving the right problem, defined as helping the user achieve their goal.

So, don't get too fond of designing that form. If there's any way to finish the transaction without making the user go through that form at all, get rid of it altogether. That gets the user closer to his goal, with less time and effort spent on his part (and maybe yours, too).

Let's use the "why" approach to dig a little deeper into some typical design scenarios.

Why does a mid-level manager use an email client? Yes, of course—"to read email." Why does she read and send email in the first place? To converse with other people. Of course, other means might achieve the same ends: the phone, a hallway conversation, a formal document. But apparently, email fills some needs that the other methods don't. What are they, and why are they important to her? The convenience of choosing when to send or respond? Privacy? The ability to archive a conversation? Social convention? What else?

A father goes to an online travel agent, types in the city where his family will be taking a summer vacation, and tries to find plane ticket prices on various dates. He's learning from what he finds, but his goal isn't just to browse and explore different options. Ask why. His goal is actually a transaction: to buy plane tickets. Again, he could have done that at many different websites, or over the phone with a live travel agent. How is this site better than those other options? Is it faster? Friendlier? More likely to find a better deal?

Sometimes goal analysis really isn't enough. A snowboarding site might provide information (for learning), an online store (for transactions), and a set of Flash movies (for entertainment). Let's say someone visits the site for a purchase, but she gets sidetracked into the information on snowboarding tricks—she has switched goals from accomplishing a transaction to browsing and learning. Maybe she'll go back to purchasing something, maybe not. And does the lifestyle and entertainment part of the site successfully entertain both the 12-year-old and the 35-year-old? Will the 35-year-old go elsewhere to buy his new board if he doesn't feel at home there, or does he not care? It's useful to expand your goal framework to include an understanding of the specific business purchase cycle. Your snowboarding customer will have different goals at different stages of this cycle. Alternately, you may want to design how you could foster a long term loyalty between the brand and the customer. This could be done via content

and functionality that fosters an identity, builds a community, and celebrates a lifestyle.

It's deceptively easy to model users as a single faceless entity—"The User"—walking through a set of simple use cases, with one task-oriented goal in mind. But that won't necessarily reflect your users' reality.

To do design well, you need to take many "softer" factors into account: expectations, gut reactions, preferences, social context, beliefs, and values. All of these factors could affect the design of an application or site. Among these softer factors, you may find the critical feature or design factor that makes your application more appealing and successful.

So, be curious. Specialize in finding out what your users are really like, and what they really think and feel.

Understanding People with Research

Empirical discovery is the only really good way to obtain this information. Qualitative research, such as one on one interviews, gives you the basis for understanding your audience's expectations, vocabulary, and how they think about their goals or structure their work. You can often detect patterns in what you're hearing. These are your signals for guiding the design. Quantitative research, such as a survey, can give numerical validation or disqualification to your quant findings.

To get a design started, you'll need to characterize the kinds of people who will be using your design (including the softer factors just mentioned), and the best way to do that is to go out and meet them.

Each user group is unique, of course. The target audience for, say, a new mobile phone app will differ dramatically from the target audience for a piece of scientific software. Even if the same person uses both, his expectations for each are different—a researcher using scientific software might tolerate a less-polished interface in exchange for high functionality, whereas that same person may stop using the mobile app if he finds its UI to be too hard to use after a few days.

Each user is unique, too. What one person finds difficult, the next one won't. The trick is to figure out what's generally true about your users, which means learning about enough individual users to separate the quirks from the common behavior patterns.

Specifically, you'll want to learn:

- Their goals in using the software or site
- The specific tasks they undertake in pursuit of those goals
- The language and words they use to describe what they're doing
- Their skill at using software similar to what you're designing
- Their attitudes toward the kind of thing you're designing, and how different designs might affect those attitudes

I can't tell you what your particular target audience is like. You need to find out what they might do with the software or site, and how it fits into the broader context of their lives. Difficult though it may be, try to describe your potential audience in terms of how and why they might use your software. You might get several distinct answers, representing distinct user groups; that's OK. You might be tempted to throw up your hands and say, "I don't know who the users are" or "Everyone is a potential user." But that doesn't help you focus your design at all—without a concrete and honest description of those people, your design will proceed with no grounding in reality.

This user-discovery phase will consume time and resources early in the design cycle, especially if you don't really have a handle on who your audience is and why they might use your designs. It's an investment. It's worth it, because the understanding you and the team gain gives long term payback in better designs: Solving the right problems, and fit for purpose.

Fortunately, lots of books, courses, and methodologies now exist to help you. Although this book does not address user research, here are some methods and topics to consider:

Direct observation

Interviews and onsite user visits put you directly into the user's world. You can ask users about what their goals are and what tasks they typically do. Usually done "on location," where users would actually use the software (e.g., in a workplace or at home), interviews can be structured—with a predefined set of questions—or unstructured, where you probe whatever subject comes up. Interviews give you a lot of flexibility; you can do many or a few, long or short, formal or informal, on the phone or in person. These are great

opportunities to learn what you don't know. Ask why. Ask it again.

Case studies

Case studies give you deep, detailed views into a few representative users or groups of users. You can sometimes use them to explore “extreme” users that push the boundaries of what the software can do, especially when the goal is a redesign of existing software. You can also use them as longitudinal studies—exploring the context of use over months or even years. Finally, if you’re designing custom software for a single user or site, you’ll want to learn as much as possible about the actual context of use.

Surveys

Written surveys can collect information from many users. You can actually get statistically significant numbers of respondents with these. Since there’s no direct human contact, you will miss a lot of extra information—whatever you don’t ask about, you won’t learn about—but you can get a very clear picture of certain aspects of your target audience. Careful survey design is essential. If you want reliable numbers instead of a qualitative “feel” for the target audience, you absolutely must write the questions correctly, pick the survey recipients correctly, and analyze the answers correctly—and that’s a science.

Personas

Personas aren’t a data-gathering method, but they do help you figure out what to do with your data once you’ve got it. This is a design technique that “models” the target audiences. For each major user group, you create a fictional person that captures the most important aspects of the users in that group: what tasks they’re trying to accomplish, their ultimate goals, and their experience levels in the subject domain and with computers in general. Personas can help you stay focused. As your design proceeds, you can ask yourself questions such as “Would this fictional person really do X? What would she do instead?”

Design research is not marketing research

You might notice that some of these methods and topics, such as interviews and surveys,

sound suspiciously like marketing activities. They are closely related. Focus groups, for example, can be useful, but be careful. In group settings, not everyone will speak up, and just one or two people may dominate the discussion and skew your understanding. There is also the very robust marketing practice of market segmentation. It resembles the definition of target audiences used here, but market segments are defined by demographics, psychographics, and other characteristics. Target audiences from a UI design perspective are defined by their task goals and behaviors.

In both cases, the whole point is to understand the audience as best you can. The difference is that as a designer, you’re trying to understand the people who use the software. A marketing professional tries to understand those who buy it.

It’s not easy to understand the real issues that underlie users’ interactions with a system. Users don’t always have the language or introspective skill to explain what they really need to accomplish their goals, and it takes a lot of work on your part to ferret out useful design concepts from what they can tell you—self-reported observations are usually biased in subtle ways.

Some of these techniques are very formal, and some aren’t. Formal and quantitative methods are valuable because they’re good science. When applied correctly, they help you see the world as it actually is, not how you think it is. If you do user research haphazardly, without accounting for biases such as the self-selection of users, you may end up with data that doesn’t reflect your actual target audience—and that can only hurt your design in the long run.

But even if you don’t have time for formal methods, it’s better to just meet a few users informally than to not do any discovery at all. Talking with users is good for the soul. If you’re able to empathize with users and imagine those individuals actually using your design, you’ll produce something much better.

The Patterns

The following patterns describe some of the most common ways people think and behave as it relates to software interfaces. Even though individuals are unique, people in general behave predictably. Designers have been doing site visits and user observations for years; cognitive scientists and other researchers have spent many hundreds of hours watching how people do things and how they think about what they do.

So, when you observe people using your software, or doing whatever activity you want to support with new software, you can expect them to do certain things. The behavioral patterns that follow are often seen in user observations. Odds are good that you'll see them too, especially if you look for them.

(A note for pattern enthusiasts: these patterns aren't like the others in this book. They describe human behaviors—not interface design elements—and they're not prescriptive, like the patterns in other chapters. Instead of being structured like the other patterns, these are presented as small essays.)

Again, an interface that supports these patterns well will help users achieve their goals far more effectively than interfaces that don't support them. And the patterns are not just about the interface, either. Sometimes the entire package—interface, underlying architecture, feature choice, documentation, everything—needs to be considered in light of these behaviors. But as the interface designer or interaction designer, you should think about these as much as anyone on your team. You might be in a better place than anyone to advocate for the users.

- “Safe Exploration”
- “Instant Gratification”
- “Satisficing”
- “Changes in Midstream”
- “Deferred Choices”
- “Incremental Construction”
- “Habituation”
- “Microbreaks”
- “Spatial Memory”
- “Prospective Memory”
- “Streamlined Repetition”

- “Keyboard Only”
- “Social Proof”

Safe Exploration

“Let me explore without getting lost or getting into trouble.”

When someone feels like she can explore an interface and not suffer dire consequences, she’s likely to learn more—and feel more positive about it—than someone who doesn’t explore. Good software allows people to try something unfamiliar, back out, and try something else, all without stress.

Those “dire consequences” don’t even have to be very bad. Mere annoyance can be enough to deter someone from trying things out voluntarily. Clicking away pop-up windows, reentering data that was mistakenly erased, suddenly muting the volume on one’s laptop when a website unexpectedly plays loud music—all can be discouraging. When you design almost any kind of software interface, make many avenues of exploration available for users to experiment with, without costing the user anything.

This pattern encompasses several of the most effective usability guidelines, based on research, as identified by usability expert Jakob Nielsen. These guidelines are²:

- Visibility of system status
- Match between the system and the real world
- User control and freedom

Here are some examples of what “Safe Exploration” is like:

A photographer tries out a few image filters in an image-processing application. He then decides he doesn’t like the results, and clicks Undo a few times to get back to where he was. Then he tries another filter, and another, each time being able to back out of what he did. (The pattern named “Multi-Level Undo”, in Chapter 6, describes how this works.)

A new visitor to a company’s home page clicks various links just to see what’s there, trusting that the Back button will always get her back to the main page. No extra windows or pop ups open, and the Back button keeps working predictably. You can

imagine that if a web app does something different in response to the Back button—or if an application offers a button that seems like a Back button, but doesn’t behave quite like it—confusion might ensue. The user can get disoriented while navigating, and may abandon the app altogether.

Instant Gratification

“I want to accomplish something now, not later.”

People like to see immediate results from the actions they take—it’s human nature. If someone starts using an application and gets a “success experience” within the first few seconds, that’s gratifying! He’ll be more likely to keep using it, even if it gets harder later. He will feel more confident in the application, and more confident in himself, than if it had taken a while to figure things out.

The need to support instant gratification has many design ramifications. For instance, if you can predict the first thing a new user is likely to do, you should design the UI to make that first thing stunningly easy. If the user’s goal is to create something, for instance, then create a new canvas, put a call to action on it, and place a palette next to it. If the user’s goal is to accomplish some task, point the way toward a typical starting point.

This also means you shouldn’t hide introductory functionality behind anything that needs to be read or waited for, such as registrations, long sets of instructions, slow-to-load screens, advertisements, and so on. These are discouraging because they block users from finishing that first task quickly.

To summarize, anticipate their need, provide an obvious entry point, provide value to the customer first before asking for something valuable (email address, a sale) in return.

Satisficing

“This is good enough. I don’t want to spend more time learning to do it better.”

When people look at a new interface, they don’t read every piece of it methodically and then decide, “Hmmm, I think this button has the best chance of getting me what I want.” Instead, a user will rapidly scan the interface, pick whatever he sees first that might get him what he wants, and try it—even if it might be wrong.

1. Designing for People

The term satisficing is a combination of satisfying and sufficing. It was coined in 1957 by the social scientist Herbert Simon, who used it to describe the behavior of people in all kinds of economic and social situations. People are willing to accept “good enough” instead of “best” if learning all the alternatives might cost time or effort.

Satisficing is actually a very rational behavior, once you appreciate the mental work necessary to “parse” a complicated interface. As Steve Krug points out in his book *Don’t Make Me Think* (Krug, Steve. *Don’t Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders, 2014.), people don’t like to think any more than they have to—it’s work! But if the interface presents an obvious option or two that the user sees immediately, he’ll try it. Chances are good that it will be the right choice, and if not, there’s little cost in backing out and trying something else (assuming that the interface supports “Safe Exploration”).

This means several things for designers:

- Use “calls to action” in the interface. Give directions on what to do first: type here, drag an image here, tap here to begin, and so forth.
- Make labels short, plainly worded, and quick to read. (This includes menu items, buttons, links, and anything else identified by text.) They’ll be scanned and guessed about; write them so that a user’s first guess about meaning is correct. If he guesses wrong several times, he’ll be frustrated, and you’ll both be off to a bad start.
- Use the layout of the interface to communicate meaning. Chapter 4 explains how to do so in detail. Users “parse” color and form on sight, and they follow these cues more efficiently than labels that must be read.
- Make it easy to move around the interface, especially for going back to where a wrong choice might have been made hastily. Provide “escape hatches” (see Chapter 3). On typical websites, using the Back button is easy, so designing easy forward/backward navigation is especially important for web apps, installed applications, and mobile devices.

Keep in mind that a complicated interface imposes a large cognitive cost on new users. Visual complexity will often tempt nonexperts to satisfice: they look for the first thing that may work.

Satisficing is why many users end up with odd habits after they’ve been using a system

for a while. Long ago, a user may have learned Path A to do something, and even though a later version of the system offers Path B as a better alternative (or maybe it was there all along), he sees no benefit in learning it—that takes effort, after all—and keeps using the less-efficient Path A. It's not necessarily an irrational choice. Breaking old habits and learning something new takes energy, and a small improvement may not be worth the cost to the user.

Changes in Midstream

“I changed my mind about what I was doing.”

Occasionally, people change what they're doing while in the middle of doing it. Someone may walk into a room with the intent of finding a key she had left there, but while she's there, she finds a newspaper and starts reading it. Or she may visit Amazon.com to read product reviews, but ends up buying a book instead. Maybe she's just sidetracked; maybe the change is deliberate. Either way, the user's goal changes while she's using the interface you designed.

This means designers should provide opportunities for people to do that. Make choices available. Don't lock users into a choice-poor environment with no connections to other pages or functionality unless there's a good reason to do so. Those reasons do exist. See the patterns called “Wizard” (Chapter 2) and “Modal Panel” (Chapter 3) for examples.

You can also make it easy for someone to start a process, stop in the middle, and come back to it later to pick up where he left off—a property often called reentrance. For instance, a lawyer may start entering information into a form on an iPad. Then, when a client comes into the room, the lawyer turns off the device, with the intent of coming back to finish the form later. The entered information shouldn't be lost.

To support reentrance, you can make dialogs and web forms remember values typed previously, and they don't usually need to be modal; if they're not modal, they can be dragged aside on the screen for later use. Builder-style applications—text editors, code development environments, and paint programs—can let a user work on multiple projects at one time, thus letting her put any number of projects aside while she works on another one. See the “Many Workspaces” pattern in Chapter 2 for more information.

Deferred Choices

“I don’t want to answer that now; just let me finish!”

This follows from people’s desire for instant gratification. If you ask a task-focused user unnecessary questions in the process, he may prefer to skip the questions and come back to them later.

For example, some web-based bulletin boards have long and complicated procedures for registering users. Screen names, email addresses, privacy preferences, avatars, self-descriptions...the list goes on and on. “But I just wanted to post one little thing,” says the user plaintively. Why not allow him to skip most of the questions, answer the bare minimum, and come back later (if ever) to fill in the rest? Otherwise, he might be there for half an hour answering essay questions and finding the perfect avatar image.

Another example is creating a new project in a video editor. There are some things you do have to decide up front, such as the name of the project, but other choices—where on the server are you going to put this when you’re done? I don’t know yet!—can easily be deferred.

Sometimes it’s just a matter of not wanting to answer the questions. At other times, the user may not have enough information to answer yet. What if a music-writing software package asked you up front for the title, key, and tempo of a new song, before you’ve even started writing it? (See Apple’s GarageBand for this bit of “good” design.)

The implications for interface design are simple to understand, though not always easy to implement:

- Don’t accost the user with too many upfront choices in the first place.
- On the forms that he does have to use, clearly mark the required fields, and don’t make too many of them required. Let him move on without answering the optional ones.
- Sometimes you can separate the few important questions or options from others that are less important. Present the short list; hide the long list.
- Use “Good Defaults” (Chapter 8) wherever possible, to give users some reasonable default answers to start with. But keep in mind that prefilled answers still require the user to look at them, just in case they need to be changed. They have a small cost, too.

- Make it possible for users to return to the deferred fields later, and make them accessible in obvious places. Some dialog boxes show the user a short statement, such as “You can always change this later by clicking the Edit Project button.” Some websites store a user’s half-finished form entries or other persistent data, such as shopping carts with unpurchased items.

If registration is required at a website that provides useful services, users may be far more likely to register if they’re first allowed to experience the website—drawn in and engaged—and then asked later about who they are. Some sites let you complete an entire purchase without registering, then ask you at the end if you want to create a no-hassle login with the personal information provided in the purchase step.

Incremental Construction

“Let me change this. That doesn’t look right; let me change it again. That’s better.”

When people create things, they don’t usually do it all in a precise order. Even an expert doesn’t start at the beginning, work through the creation process methodically, and come out with something perfect and finished at the end.

Quite the opposite. Instead, she starts with some small piece of it, works on it, steps back and looks at it, tests it (if it’s code or some other “runnable” thing), fixes what’s wrong, and starts to build other parts of it. Or maybe she starts over, if she really doesn’t like it. The creative process goes in fits and starts. It moves backward as much as forward sometimes, and it’s often incremental, done in a series of small changes instead of a few big ones. Sometimes it’s top-down; sometimes it’s bottom-up.

Builder-style interfaces need to support that style of work. Make it easy for users to build small pieces. Keep the interface responsive to quick changes and saves. Feedback is critical: constantly show the user what the whole thing looks and behaves like, while the user works. If the user builds code, simulations, or other executable things, make the “compile” part of the cycle as short as possible, so the operational feedback feels immediate—leave little or no delay between the user making changes and seeing the results.

When creative activities are well supported by good tools, they can induce a state of flow in the user. This is a state of full absorption in the activity, during which time distorts, other distractions fall away, and the person can remain engaged for hours—the

enjoyment of the activity is its own reward. Artists, athletes, and programmers all know this state.

But bad tools will keep users distracted, guaranteed. If the user has to wait even half a minute to see the results of the incremental change she just made, her concentration is broken; flow is disrupted.

If you want to read more about flow, there are multiple books by researcher Mihaly Csikszentmihalyi. One title is *Flow* (Csikszentmihalyi, Mihaly. *Flow: The Psychology of Optimal Experience*. Harper Row, 2009.)

Habituation

“That gesture works everywhere else; why doesn’t it work here, too?”

When one uses an interface repeatedly, some frequent physical actions become reflexive: pressing Ctrl-S to save a document, clicking the Back button to leave a web page, pressing Return to close a modal dialog box, using gestures to show and hide windows—even pressing a car’s brake pedal. The user no longer needs to think consciously about these actions. They’ve become habitual.

This tendency helps people become expert users of a tool (and helps create a sense of flow, too). Habituation also measurably improves efficiency, as you can imagine. But it can also lay traps for the user. If a gesture becomes a habit, and the user tries to use it in a situation when it doesn’t work—or, worse, does something destructive—the user is caught short. He suddenly has to think about the tool again (What did I just do? How do I do what I intended?), and he might have to undo any damage done by the gesture.

Millions of people have learned the following keyboard shortcuts based on using Microsoft Word and other word processors. They are true universals now. Consistency across applications can be an advantage to use in your software design.

- Ctrl-X: Cut the selection
- Ctrl-V: Paste the selection
- Ctrl-S: Save the document

Just as important, though, is consistency within an application. Some applications are evil

because they establish an expectation that some gesture will do Action X, except in one special mode where it suddenly does Action Y. Don't do that. It's a sure bet that users will make mistakes, and the more experienced they are—that is, the more habituated they are—the more likely they are to make that mistake.

Consider this carefully if you're developing gesture-based interfaces for mobile devices. Once someone learns how to use his device and gets used to it, he will depend on the standard gestures working consistently on all applications. Check that gestures in your design all do the expected things.

This is also why confirmation dialog boxes often don't work to protect a user against accidental changes. When modal dialog boxes pop up, the user can easily get rid of them just by clicking OK or pressing Return (if the OK button is the default button). If the dialogs pop up all the time when the user makes intended changes, such as deleting files, clicking OK becomes a habituated response. Then, when it actually matters, the dialog box doesn't have any effect, because it slips right under the user's consciousness.

(I've seen at least one application that sets up the confirmation dialog box's buttons randomly from one invocation to another. One actually has to read the buttons to figure out what to click! This isn't necessarily the best way to do a confirmation dialog box—in fact, it's better to not have them at all under most circumstances—but at least this design sidesteps habituation creatively.)

Microbreaks

“I'm waiting for the train. Let me do something useful for two minutes.”

People often find themselves with a few minutes of down time. They might need a mental break while working; they might be in line at a store or sitting in a traffic jam. They might be bored or impatient. They want to do something constructive or entertaining to pass the time, knowing they won't have enough time to get deep into an online activity.

This pattern is especially applicable to mobile devices, because people can easily pull them out at times such as these. The enormous success of the social media technology sector was built in no small part by taking advantage of this. Social and casual gaming, Facebook, Instagram, Snap...all are enjoyed in microbreaks.

Here are some typical activities during microbreaks:

- Checking email
- Reading a “News Stream” (in Chapter 2) such as Facebook or Twitter
- Visiting a news site to find out what’s going on in the world
- Watching a short video
- Doing a quick web search
- Reading an online book
- Playing a short game

The key to supporting microbreaks is to make an activity easy and fast to reach—as easy as turning on the device and selecting an application (or website). Don’t require complicated setup. Don’t take forever to load. And if the user needs to sign in to a service, try to retain the previous authentication so that she doesn’t have to sign in every time.

For “News Stream” services, load the freshest content as quickly as possible and show it in the first screen the user sees. Other activities, such as games, videos, or online books, should remember where the user left them last time and restore the app or site to its previous state, without asking (thus supporting reentrance).

If you’re designing an email application, or anything else for which the user needs to do “housekeeping” to maintain order, give her a way to triage items efficiently. This means showing enough data per item so that she can identify, for instance, a message’s contents and sender. You can also give her a chance to “star” or otherwise annotate items of interest, delete items easily, and write short responses and updates.

Long load times deserve another mention. Taking too long to load content is a sure way to make users give up on your app—especially during microbreaks! Make sure the page is engineered so that readable, useful content loads first, and with very little delay.

Spatial Memory

“I swear that button was here a minute ago. Where did it go?”

When people manipulate objects and documents, they often find them again later by remembering where they are, not what they're named.

Take the Windows, Mac, or Linux desktop. Many people use the desktop background as a place to put documents, frequently used applications, and other such things. It turns out that people tend to use spatial memory to find things on the desktop, and it's very effective. People devise their own groupings, for instance, or recall that "this document was at the top right over by such-and-such." (Naturally, there are real-world equivalents, too. Many people's desks are "organized chaos," an apparent mess in which the office owner can find anything instantly. But heaven forbid that someone should clean it up for him.)

Many applications put their dialog buttons—OK, Cancel, and so on—in predictable places, partly because spatial memory for them is so strong. In complex applications, people may also find things by remembering where they are relative to other things: tools on toolbars, objects in hierarchies, and so forth. Therefore, you should use patterns such as "Responsive Disclosure" (Chapter 4) carefully. Adding items to blank spaces in an interface doesn't cause problems, but rearranging existing controls can disrupt spatial memory and make things harder to find. It depends. Try it out on your users if you're not sure.

Many mobile applications and games consist of just a few screens. Often the start screen is designed to be where users spend all their time. There might not be any apparent navigation. But users learn to swipe left, right, up, or down to get to the other screens (such as messaging or settings). These other screens are there, just off to the side. Snap is a good example of a mobile app that is designed to use people's spatial memory.

Along with habituation, which is closely related, spatial memory is another reason why consistency across and within a platform's applications is good. People may expect to find similar functionality in similar places. See the "Sign-in Tools" pattern (Chapter 3) for an example.

Spatial memory explains why it's good to provide user-arranged areas for storing documents and objects, such as the aforementioned desktop. Such things aren't always practical, especially with large numbers of objects, but it works quite well with small numbers. When people arrange things themselves, they're likely to remember where they put them. (Just don't rearrange it for them unless they ask!) The "Movable Panels" pattern in Chapter 4 describes one particular way to do this.

Also, this is why changing menus dynamically can sometimes backfire. People get used to seeing certain items on the tops and bottoms of menus. Rearranging or compacting menu items “helpfully” can work against habituation and lead to user errors. So can changing navigation menus on web pages. Try to keep menu items in the same place, and in the same order, on all subpages in a site.

Incidentally, the tops and bottoms of lists and menus are special locations, cognitively speaking. People notice and remember them more than items in the middle of a list. The first and last items are perhaps the worst ones to change.

Prospective Memory

“I’m putting this here to remind myself to deal with it later.”

Prospective memory is a well-known phenomenon in psychology that doesn’t seem to have gained much traction yet in interface design. But I think it should.

We engage in prospective memory when we plan to do something in the future, and we arrange some way of reminding ourselves to do it. For example, if you need to bring a book to work the next day, you might put it on a table beside the front door the night before. If you need to respond to someone’s email later (just not right now!), you might leave that email on your screen as a physical reminder. Or if you tend to miss meetings, you might arrange for Outlook or your mobile device to ring an alarm tone five minutes before each meeting.

Basically, this is something almost everyone does. It’s a part of how we cope with our complicated, highly scheduled, multitasked lives: we use knowledge “in the world” to aid our own imperfect memories. We need to be able to do it well.

Some software does support prospective remembering. Outlook and most mobile platforms, as mentioned earlier, implement it directly and actively; they have calendars, and they sound alarms. Trello is another example; it is a kanban board of cards. Memory aids that people use can include:

- Notes to oneself, like virtual “sticky notes”
- Windows left on-screen
- Annotations put directly into documents (such as “Finish me!”)

- Browser bookmarks, for websites to be viewed later
- Documents stored on the desktop, rather than in the usual places in the filesystem
- Email kept in an inbox (and maybe flagged) instead of filed away

People use all kinds of artifacts to support passive prospective remembering. But notice that almost none of the techniques in the preceding list were designed with that in mind! What they were designed for is flexibility—and a laissez-faire attitude toward how users organize their stuff. A good email client lets you create folders with any names you want, and it doesn't care what you do with messages in your inbox. Text editors don't care what you type, or what giant bold magenta text means to you; code editors don't care that you have a “Finish this” comment in a method header. Browsers don't care why you keep certain bookmarks around.

In many cases, that kind of hands-off flexibility is all you really need. Give people the tools to create their own reminder systems. Just don't try to design a system that's too smart for its own good. For instance, don't assume that just because a window's been idle for a while, that no one's using it and it should be closed. In general, don't “helpfully” clean up files or objects that the system may think are useless; someone may be leaving them around for a reason. Also, don't organize or sort things automatically unless the user asks the system to do so.

As a designer, is there anything positive you can do for prospective memory? If someone leaves a form half-finished and closes it temporarily, you could retain the data in it for the next time—it will help remind the user where she left off. (See the “Deferred Choices” pattern.) Similarly, many applications recall the last few objects or documents they edited. You could offer bookmark-like lists of “objects of interest”—both past and future—and make those lists easily available for reading and editing. You can implement “Many Workspaces”, which lets users leave unfinished pages open while they work on something else.

Here's a bigger challenge: if the user starts tasks and leaves them without finishing them, think about how to leave some artifacts around, other than open windows, that identify the unfinished tasks. Another idea: how might a user gather reminders from different sources (email, documents, calendars, etc.) into one place? Be creative!

Streamlined Repetition

“I have to repeat this how many times?”

In many kinds of applications, users sometimes find themselves having to perform the same operation over and over again. The easier it is for them, the better. If you can help reduce that operation down to one keystroke or click per repetition—or, better, just a few keystrokes or clicks for all repetitions—you will spare users much tedium.

Find and Replace dialog boxes, often found in text editors (Word, email composers, etc.), are one good adaptation to this behavior. In these dialog boxes, the user types the old phrase and the new phrase. Then it takes only one Replace button click per occurrence in the whole document. And that’s only if the user wants to see or veto each replacement—if she’s confident that she really should replace all occurrences, she can click the Replace All button; one gesture does the whole job.

Here’s a more general example. Photoshop lets you record “actions” when you want to perform some arbitrary sequence of actions with a single click. If you want to resize, crop, brighten, and save 20 images, you can record those four steps as they’re done to the first image, and then click that action’s Play button for each of the remaining 19. See the “Macros” pattern in Chapter 6 for more information.

Scripting environments are even more general. Unix and its variants allow you to script anything you can type into a shell. You can recall and execute single commands, even long ones, with a Ctrl-P and Return. You can take any set of commands you issue to the command line, put them in a for loop, and execute them by pressing the Return key once.

Or you can put them in a shell script (or in a for loop in a shell script) and execute them as a single command. Scripting is very powerful, and when complex, it becomes full-fledged programming.

Other variants include copy-and-paste capability (preventing the need to retype the same thing in a million places), user-defined “shortcuts” to applications on operating-system desktops (preventing the need to find those applications’ directories in the filesystem), browser bookmarks (so users don’t have to type URLs), and even keyboard shortcuts.

Direct observation of users can help you figure out just what kinds of repetitive tasks you need to support. Users won’t always tell you outright. They may not even be aware that they’re doing repetitive things that could be streamlined with the right tools—they may have been doing it so long that they don’t even notice anymore. By watching them work, you may see what they don’t see.

In any case, the idea is to offer users ways to streamline the repetitive tasks that could otherwise be time-consuming, tedious, and error-prone.

Keyboard Only

“Please don’t make me use the mouse.”

Some people have real physical trouble using a mouse. Others prefer not to keep switching between the mouse and keyboard because that takes time and effort—they’d rather keep their hands on the keyboard at all times. Still others can’t see the screen, and their assistive technologies often interact with the software using just the keyboard API.

For the sakes of these users, some applications are designed to be “driven” entirely via the keyboard. They’re usually mouse-driven too, but there is no operation that must be done with only the mouse—keyboard-only users aren’t shut out of any functionality.

Several standard techniques exist for keyboard-only usage:

- You can define keyboard shortcuts, accelerators, and mnemonics for operations reachable via application menu bars, such as Ctrl-S for Save. See your platform style guide for the standard ones.
- Selection from lists, even multiple selection, is usually possible using arrow keys in combination with modifiers (such as the Shift key), though this depends on which component set you use.
- The Tab key typically moves the keyboard focus—the control that gets keyboard entries at the moment—from one control to the next, and Shift-Tab moves backward. This is sometimes called tab traversal. Many users expect it to work on form-style interfaces.
- Most standard controls, even radio buttons and combo boxes, let users change their values from the keyboard by using arrow keys, the Return key, or the space bar.
- Dialog boxes and web pages often have a “default button”—a button representing an action that says “I’m done with this task now.” On web pages, it’s often Submit or Done; on dialog boxes, OK or Cancel. When users press the Return key on this page or dialog box, that’s the operation that occurs. Then it moves the user to the next page or returns him to the previous window.

There are more techniques. Forms, control panels, and standard web pages are fairly easy to drive from the keyboard. Graphic editors, and anything else that's mostly spatial, are much harder, though not impossible.

Keyboard-only usage is particularly important for data-entry applications. In these, speed of data entry is critical, and users can't afford to move their hands off the keyboard to the mouse every time they want to move from one field to another or even one page to another. (In fact, many of these forms don't even require users to press the Tab key to traverse between controls; it's done automatically.)

Social Media, Social Proof, and Collaboration

“What did everyone else say about this?”

People are social. As strong as our opinions may sometimes be, we tend to be influenced by what our peers say and do. And we are powerfully attuned to seeking approval from others and belonging to a group. We maintain social media identities. We contribute to groups and people we care about.

The advice of peers, whether direct or indirect, influences people's choices when they decide any number of things. Finding things online, performing transactions (Should I buy this product?), playing games (What have other players done here?), and even building things—people can be more effective when aided by others. If not, they might at least be happier with the outcome.

We are much more likely to watch, read, buy, join, share, comment, or take any other action if we see that someone we know has recommended it or done it or is present. This is called social proof.

All of these real world dynamics underpin the massive scale and success of social computing in its many forms. It is fair to say that a social aspect or layer is part of almost all software today. Enabling social dynamics in your software can bring increased engagement, virality, community and growth.

Some examples of social functionality include:

User-generated reviews and comments

These allow individuals to get a sense of the wisdom of the crowd. Reviews can be

1. Designing for People

rated, and participants can gain fame or other rewards for being rated as a good reviewer

Everything is a social object

Text posts, images, video, check ins, almost anything that users create in social media becomes an object that people can virtually gather around. Anything can be shared, rated, have a discussion thread attached to it, and similar activities.

Collaboration

Business productivity and communication software has been transformed by software that allows people separated by space and time to come together in discussion threads, document reviews, video conferencing, tracking status, live and asynchronous communications, and many other activities.

Social proof motivates people to take action

Social group identity, participation and recognition is powerfully rewarding to people.

Designing these capabilities into your interface creates the opportunity for social dynamics to increase your audiences' engagement, reward and growth.

Of the patterns in this book, “Multi-Level Help” (Chapter 2) most directly addresses this idea; an online support community is a valuable part of a complete help system for some applications.

¹ This is the same principle that underlies a well-known technique called *root-cause analysis*. But root-cause analysis is a tool for fixing organizational failures; here, we use its “five whys” (more or less) to understand everyday user behaviors and feature requests.

² Nielsen, Jakob. “10 Heuristics for User Interface Design: Article by Jakob Nielsen.” *Nielsen Norman Group*, 24 Apr. 1994, www.nngroup.com/articles/ten-usability-heuristics/.

Chapter 2. Organizing the Content: Information Architecture and Application Structure

In the first chapter, we outlined the purpose and outcomes of understanding how people use software. Specifically, you got a sense for what is critical to people when it comes to designing web sites, applications and interfaces:

- Overall goals in using your site or application
- The breakdown of tasks in carrying out those goals
- How they think about the particular subject or domain
- The language they use to think and talk about this subject
- How skilled or unskilled they are in doing the work
- Their attitudes towards the subject.

At this point, you might feel confident that you understand what your users want out of your application or site. If you're really on the ball, you've written down some typical scenarios that describe how people might use high-level elements of the application to accomplish their goals. You have a clear idea of what value this application adds to people's lives.

It's tempting at this stage to want to go directly to designing the screens and components of your interface, working with colors, typography, language and layouts. And if you are a visual thinker, there's nothing wrong with starting with sketches to capture early ideas. If you're the kind of person who likes to think visually and needs to play with sketches while working out the broad strokes of the design, go for it.

However, to take advantage fully of your customer insights and give your design the best chance for success, the next step is to use your understandings to develop your information architecture. Don't get locked in to specific interface design decisions just yet. Instead, take a step back and think about designing the overall structure and framework of your software so that it makes sense from your user's point of view. Think through the information, the workflows, the language of the site or application and then organizing them so that they are easy to learn and easy to use.

2. Organizing the Content: Information Architecture and Application Structure

This is Information Architecture. Let's break this down by looking at the benefits and scope of information architecture design.

Purpose

The purpose of information architecture is to create the framework for your digital product, service, site or application to be successful. A critical part of this for digital, interactive experiences is that the interface shouldn't get in the way.

The irony with working on your information architecture is that customers only notice Information Architecture when it's bad. They experience this in a number of ways: The organization of the experience makes no sense. The interface is confusing and the screens are frustrating. Customers don't understand the language they see on screen. They can't find what they need when they need it. Basically, it gets in the way.

On the other hand, if we've done our jobs properly then our design is invisible. Users don't really notice great Information Architecture. All users know is that they have a natural, efficient and pleasant digital experience.

What does that mean in practice?

Stepping back, we can say that the context is people who are trying to do something: Find information, watch a video, buy something, sign up for something. In short, they have a task to do. But the people who make the digital product--that's you!--can't be there in person. You have to design your app to mimic what a good customer service representative would do:

- Anticipate what they need
- Organize and talk about the information from the customer's point of view
- Offer information in a clear, simple way
- Offer clear next steps
- Make it really obvious where you are and what's happening

Definition

2. Organizing the Content: Information Architecture and Application Structure

Information architecture (IA) is the art of organizing and labeling an information space for optimal understanding and use. Specifically, information architecture is using your understanding of the people who will use your site or application to design:

- Structures or categories for organizing your content and functionality
- Different ways that people can navigate through the experience
- Intuitive workflows or multi-step processes for getting tasks done
- Labels and language for communicating about this content
- Searching, browsing and filtering tools to help them find what they're looking for
- A standardized system of screen types, templates, or layouts so information is presented consistently and for maximum usability

Information architecture encompasses many things: presenting, searching, browsing, labeling, categorizing, sorting, manipulating, and strategically hiding information. Especially if you're working with a new product, this is where you should start. The goal is for all this to make sense from the user's point of view so they can use your site or app successfully.

Designing an information space for people

In the same way that construction architects draw up blueprints before a house is actually constructed, designers--information architects--create a plan for how their information space will be laid out in a usable manner and how people will move around in it, and get work done there. In both cases, there is efficiency and value in thinking through how people will use what you going to build, before you build it.

Approach

It can be helpful to think about an application in terms of its underlying data and tasks. Do this without thinking about the look and feel that they will ultimately take on. Think more abstractly.

- What information and tools do you need to show to the users?

2. Organizing the Content: Information Architecture and Application Structure

- Based on their expectations and immediate situation, when do you show them?
- How are they categorized and ordered?
- What do users need to do with them?
- How many ways can you present those things and tasks? You may need more than one.
- How can you make it usable from their point of view?

These lines of inquiry may help you think more creatively about the information architecture you're designing.

Separate Information from Presentation

Thinking about information architecture separately from the visual design is so important it's worth looking at in more detail. The design challenge becomes more manageable when you tackle it in phases. A critical part of this is to separate your content, data and information from its presentation.

In fact, it's useful to think of designing as a process of building up layers of design, one on top of the other.

In the same way that software engineers think of applications in three layers--databases, tools and queries, and reports, results, and responses--designers can think of their design having three layers.

[Figure 2-1](#) gives a schematic representation of this approach. Information architecture is the lowest layer, the foundation. Much like with a physical building, the structure of this foundation will ultimately be invisible but will affect what gets built on top of it. In the digital world, we are concerned with creating an information architecture foundation that has appropriate concepts, labels and categories. It lays out the permanent information structure that users can navigate, search through, and manipulate on the upper layers of the experience. Making sure your organization decisions make sense from your user's point of view is important since they form the framework for everything else.

2. Organizing the Content: Information Architecture and Application Structure



Figure 2-1. Layered Design: Designing up from the content/data layer to the presentation layer. (Based on ideas by: Garrett, Jesse James. *The Elements of User Experience: User-Centered Design for the Web and Beyond*. New Riders, 2011.)

The middle layer is the functionality and information delivery layer of your site or app. It is the screens, pages, stories, lists and cards they browse, search and read. It is the tools they use to search, filter, monitor, analyze, communicate and create.

The topmost layer is the presentation layer. It is the visual design and editorial system for presenting and rendering. It consists of colors, typography, layouts, graphics, and more. When done well, the presentation layer design creates focus, flow and clarity.

Mutually Exclusive, Collectively Exhaustive

As mentioned above, your content and information needs to be organized in a way that makes sense to your audience. As you go through the process of organizing the data and content into major categories or sections, there is a useful rule of thumb: “MECE.”

MECE stands for “mutually exclusive, collectively exhaustive.” What that means is two things. First, your information architecture should have categories, or silos, or buckets of content that are clearly distinct from one another, with no confusing overlap. Second, “collectively exhaustive” means that taken all together, your organization schema is complete. It accounts for all the information your site or application is supposed to handle--all the situations and use cases you are designing for. There’s a place to find everything or put anything. Later in the process, your information structure can expand to accommodate new data without things becoming confusing.

These categories will form the foundation of your navigation system, which is discussed in more detail in Chapter 4 (“Getting Around”).

To document and communicate this organization scheme, information architects develop

2. Organizing the Content: Information Architecture and Application Structure
such tools as site maps and content outlines.

Ways to Organize and Categorize Content

There are some additional methods and practices you can use to help organize and categorize the information in your site or app. These are especially useful when planning how to display large amounts of structured data in tables. They also are important for planning for how users will search and browse, filter information, and sort and refine their results. There are six described briefly below. This list is based on the work of Richard Saul Wurman in his book on information architecture, *Information Anxiety 2* (Wurman, Richard Saul. *Information Anxiety 2*. Que, 2001.), and Abby Covert's *How to Make Sense of Any Mess* (Covert, Abby, and Nicole Fenton. *How to Make Sense of Any Mess*. Abby Covert, 2014.). Both make excellent further reading.

Alphabetical

This means organizing lists, names and any labelled items according to the sequence of the alphabet. This can be done in descending order, from A to Z, or in ascending or reverse order, from Z to A. This can also include numbers if they are part of the name or label, with the numbers “0” and up preceding the alphabetical letters. This is a great default for any list or menu of items.

Number

Organizing by number is another common method. It can include a number of variations. First is according to integer, where items or the numbers themselves are sorted in ascending or descending order based on the sequence of the number system. The second is by ordinal position: first, second, third and so on. A third way is by value or total. Things such as financial amounts, discounts and rates of change can be arranged from largest value to smallest, and vice versa. Tabular data uses this pattern heavily.

Time

Chronological order is another useful way to organize content. This is very common in social media feeds, where reverse chronological order is common--the most recent item is

2. Organizing the Content: Information Architecture and Application Structure at the top of the list, with older items lower in the list. Information can be organized by date, time, or duration in ascending or descending order. They can also be arranged by frequency--low to high, and the reverse. They can also be arranged by sequence in time: which happened first, or should happen first, as in steps in a process. Tasks are also often split into a sequence of steps (can also be thought of as a number, see above).

Location

Location can mean organizing by a geographic or spatial location. There are many systems for specifying geographic location, such as latitude and longitude. Geographic categories are often nested or hierarchical, such as countries containing states, which contain cities (see below). Location can also be distance from or distance to some reference point, and ordering based on that. In digital systems, it's critical for users to learn their location in the information space, both overall and on an individual screen.

Hierarchy

This is a more useful method because it can be used to organize more broadly. It can mean organizing by size, importance, rank, or scale. A classic example is organizing from biggest to smallest. Action items can be organized from high priority to low priority or importance. There can also be a container or parent-child relationship, with the larger item containing a smaller one. Examples of this can be a country category, which contains states, as mentioned above.

Category or Facet

In information architecture, content can be labelled and then grouped into categories or topics. A category can be thought of as a feature or quality that a set of items has in common. This is a very useful method of organizing because it is so broad. Often there is a spectrum or degree implied in the category that can be used to order the items in the set. A simple example is organizing by color. Advanced organizing schemes use facets. A facet system uses multiple qualities or categories, with a range of values in each, to categorize information. A good example of faceted classification in action is Amazon, where customers can use multiple values to narrow down their product search: Price, availability, and customer rating, for example.

Designing for Task and Workflow-Dominant Apps

Information architecture also includes designing workflows and tasks. Documentation related to this often includes such things as user stories and flow diagrams.

Make Frequently Used Items Visible

The first process or task rule of thumb in designing for tasks or workflow is frequency of use. Tasks, controls, commands or topics that are repeated often or used frequently should be immediately available to the user. This often means that there is a selector on a screen or a topic in the navigation that has been elevated or promoted so it's a shortcut or quick link. Basically, frequently accessed or needed items should be available without having to search or browse. On the other hand, controls or information that are not needed very often can be hidden away or accessed only by navigating to it. User settings and help systems are good examples of features that hidden in standard use but accessible when needed.

“Chunk Up” Jobs into a Sequence of Steps

Sequencing is the second organizing principle for IA for tasks and workflows. This means breaking up a big task or process into a series of steps. Do this to make each individual step less demanding on the user. This is often the structure for a wizard or multi-step process that leads the user through a series of steps. Along with this, plan for communicating to the user where they are in the process.

Design for Novice and Experienced Users

Another IA organizing principle considering the level of learning, skill or mastery that a particular user has. Like in computer games, it's useful to consider how first time users may need a simplified interface, or special additional help, to assist them. This can take the form of additional instructions, screen overlays or wizards for complex processes. Many applications and web sites devote resources to designing a “new user experience.” On the other end of the skill spectrum, advanced or experienced users can be fast and efficient with complicated interfaces that are densely packed with information and selectors. Offering them “accelerators” such as short cuts and the ability to customize their interface help them be efficient. Designing for keyboard only navigation and input is

2. Organizing the Content: Information Architecture and Application Structure also valuable here.

Design for Multiple Screens and Channels

It goes without saying that consumers and business users alike now expect to access information, sites and applications via multiple channels such as desktop, mobile and messaging, and across a multitude of screen sizes and devices. Voice-activated services interfaces don't have screens at all.

Multiple Channels and Screen Sizes are Today's Reality

In designing your information architecture, consider what channels, modes and devices your site or application will need to function across. This will drive how your information is organized, segmented and sequenced.

“Chunk Up” Your Information into Cards

A common pattern in several of the examples below is a reliance on a “cards” pattern. With the majority of all digital, interactive experiences being via mobile devices, it makes sense to make the building block of your experience a card that fits on a smaller screen. This small container of information, photos and other data can work individually or can be displayed in a list or grid on larger screens. The key is to plan for how to scale down and scale up the experience while still delivering access and control to your information and features.

Designing a System of Screen Types

As mentioned above, information architecture also includes designing a system of screen types. Each screen type has a differentiated function. In this way, the user can learn how to use each screen reliably, even when the content in it changes based on topic, filters and other selections

A useful framework for approaching screen types is from Theresa Neil. She developed ideas for application structures in the context of rich Internet applications (RIAs). She defines three types of structures based on the user's primary goal: information, process, and creation.¹

2. Organizing the Content: Information Architecture and Application Structure

This list gives us a framework within which to fit the idioms and patterns we'll talk about in this and other chapters.

Now let's look at pages that serve single important functions. In an application, this might be a main screen or a major interactive tool; in a richly interactive website, it might be a single page, such as Gmail's main screen; in a more static website, it might be a group of pages devoted to one process or function.

Any such page will primarily do one of these things:

- Overview: Show a list or set of things
- Focus: Show one single thing, such as a map, book, video, or game
- Make: Provide tools to create a thing
- Do: Facilitate a single task
- Most apps and sites do some combination of these things, of course. But consider developing a screen system where each has a particular organizing principle.

Overview: Show a List or Grid of Things or Options

This is what most of the world's home pages, start screens and content sites seem to do. The digital world has converged on many common idioms for showing lists, most of which are familiar to you:

- simple text lists
- menus
- grids of cards or images
- search results in list or grid form
- lists of email messages or other communications
- tables of data
- trees, panels and accordions

2. Organizing the Content: Information Architecture and Application Structure

These overview screens present information organization challenges. Some questions to consider in designing your overview screens include:

- How big is the data set or how long is the list?
- How much space is available to display it?
- Is it flat or hierarchical, and if it is a hierarchy, what kind?
- How is it ordered, and can the user change that ordering dynamically?
- How can the user search, filter and sort?
- What information or operations are associated with each list item, and when and how should they be shown?

Because lists and grids are so common, a solid grasp of the different ways to present them can benefit any designer. A few patterns for designing an interface around a list are described in this chapter (others are in Chapter 5).

You can build either an entire app or site, or a small piece of a larger artifact, around one of these patterns. They set up a structure that other display techniques—text lists, thumbnail lists, and so on—can fit into. Other top-level organizations not listed here might include calendars, full-page menus, and search results.

“Feature, Search, and Browse” is the pattern followed by countless websites that show products and written content. Searching and browsing provide two ways for users to find items of interest, while the front page features one item to attract interest.

“Streams and Feeds” Blogs, news sites, email readers, and social sites such as Twitter all use a news stream or social stream pattern to list their content, with the most recent updates at the top.

“Grids” is a well-defined interface type for presenting stories, actions, cards and selectors. It is also used for handling photos and other pictorial documents. It can accommodate hierarchies and flat lists, tools to arrange and reorder documents, tools to operate directly on pictures, launch apps, drill down to details and so on.

Once you’ve chosen an overall design for the interface, you might look at other patterns and techniques for displaying lists. See Chapter 5 for a thorough discussion.

Focus: Show One Single Thing

The whole point of this screen type is to show or play a single piece of content or functionality. This can be a single story or a single image. There might be small-scale tools clustered around the content—scrollers and sliders, sign-in box, global navigation, headers and footers, and so forth—but they are minor and easily designed. Your design might take one of these shapes:

A long, vertically scrolled page of flowed text (articles, books, and similar long-form content).

A zoomable interface for very large, fine-grained artifacts, such as maps, images, or information graphics. Map sites such as Google Maps provide some well-known examples:

- The “media player” idiom, including video and audio players.
- As you design this interface, consider the following patterns and techniques to support the design:
 - “Mobile Direct Access,” to take the user directly into the main function of your app, often using location and time data to generate valuable information without the user providing any input
 - “Alternative Views”, to show the content in more than one way.
 - “Many Workspaces”, in case people want to see more than one place, state, or document at one time.
 - “Deep-linked State”, in Chapter 3. With this, a user can save a certain place or state within the content so that he can come back to it later or send someone else a URL.
 - “Sharing Widget” and other social patterns, in Chapter 9.

Some of the mobile patterns described in Chapter 10, if one of your design goals is to deliver the content on mobile devices.

Make: Provide Tools to Create a Thing

2. Organizing the Content: Information Architecture and Application Structure

This screen type is for creating or updating digital objects. Most people are familiar with the idioms used by these tools: text editors, code editors, image editors, editors that create vector graphics, and spreadsheets.

At the level of application structure or IA, the following patterns are often found:

- “Canvas Plus Palette” describes most of these applications. This highly recognizable, well-established pattern for visual editors sets user expectations very strongly.
- Almost all applications of this type provide “Many Workspaces”—usually windows containing different documents, which enable users to work on them in parallel.
- “Blank Slate Invitation” is named and written about in *Designing Web Interfaces* (<http://oreilly.com/catalog/9780596516253/>) by Bill Scott and Theresa Neil (O’Reilly), and is a profoundly useful pattern for builders and editors. It is closely related to the “Input Hints” pattern in Chapter 8.

Do: Facilitate a Single Task

Maybe your interface’s job isn’t to show a list of anything or create anything, but simply to get a job done. Signing in, registering, posting, printing, uploading, purchasing, changing a setting—all such tasks fall into this category.

Forms do a lot of work here. Chapter 8 talks about forms at length and lists many controls and patterns to support effective forms. Chapter 6 defines another useful set of patterns that concentrate more on “verbs” than “nouns.”

Not much IA needs to be done if the user can do the necessary work in a small, contained area, such as a sign-in box. But when the task gets more complicated than that—if it’s long, or branched, or has too many possibilities—part of your job is to work out how the task is structured.

Much of the time, you’ll want to break the task down into smaller steps or groups of steps. For these, a “Wizard” might work well for users who need to be walked through the task.

A “Settings Editor” is a very common type of interface that gives users a way to change the settings or preferences of something—an application, a document, a product, and so on. This isn’t a step-by-step task at all. Here, your job is to give users open access to a

2. Organizing the Content: Information Architecture and Application Structure
wide variety of choices and switches and let them change only what they need, when they need it, knowing that they will skip around.

The Patterns

Feature, Search, and Browse

What

Put three elements on the main page of the site or app: a featured article or product, a search box (expanded by default, or collapsed), and a list of items or categories that can be browsed.

Use when

Your site offers users long lists of items—articles, products, videos, and so on—that can be browsed and searched. You want to engage incoming users immediately by giving them something interesting to read or watch.

Alternately, your site focuses on enabling searching or transacting. In this case, search is the dominant element on the screen. Featured content and browsing have secondary importance.

Why

These three elements are found together on many sites. Searching and browsing go hand in hand to find desired items. Some people will know what they’re looking for and use the search box, while others will do more open-ended browsing through the lists and categories you show them.

Featured items are how you engage the user. They’re far more interesting than just category lists and search boxes, especially when you use appealing images and headlines. A user who lands on your page now has something to read or experiment with, without doing any additional work at all—and he may find it more interesting than whatever he originally came for.

2. Organizing the Content: Information Architecture and Application Structure

How

Place a search box in a prominent location, such as an upper corner, or in a banner across the middle top of the site. Demarcate it well from the rest of the site—use whitespace to set it off, and use a different surrounding background color if necessary.

Alternately, display search in a collapsed or compacted state. It still needs to be easy to see and access, but it can be an icon or the label “Search.” Selecting this opens the full search field. This pattern saves space on smaller screens.

Set aside “Center Stage” (see Chapter 4) for the featured article, product, or video. Very near it, and still above the fold, place an area for browsing the rest of the site’s content. Most sites show a list of stories, cards, topics or product categories. These might be links to pages devoted to those categories.

If the category labels open in place to show subcategories, the list behaves like a tree. Some sites, such as Amazon, turn the category labels into menus: when the pointer rolls over the label, a menu of subcategories appears.

Choose the features well. Features are a good way to sell items, advertise specials, and call attention to breaking news. However, they are the front door and also define what your site is about. What will they want to know about? What will capture their attention and hold them at your site?

As the user browses through categories and subcategories, help him “stay found” with the “Breadcrumbs” pattern (Chapter 3).

Examples

Section 1: Content-Centric Websites

The following three examples demonstrate the classic pattern of Feature, Search, Browse. WebMD ([Figure 2-2](#)), Yahoo! ([Figure 2-3](#)) and Sheknows ([Figure 2-4](#)) are news- and content-centric digital publishers. WebMD and Yahoo! have search at top with a single large feature. Sheknows offers a variation: Two features above a prominent search input.



2. Organizing the Content: Information Architecture and Application Structure

ADVERTISEMENT



Tell us where it hurts.

Check Your Symptoms

WebMD Symptom Checker

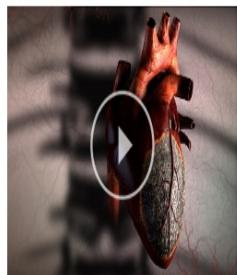
Too Much Media, Too Little Attention

Being immersed in electronic stimuli may be taking its toll.

More: Attention Issues Related to Technology Seen at Age 5



Trending Videos



Inside a Widow-Maker Heart Attack



5 Facts About Allergies



Unexpected Anxiety Triggers

WebMD
NEWSLETTERS

Support, inspiration
and timely
health information

Subscribe 

ADVERTISEMENT

Top Stories



Dangerous Kissing Bug Marches North in U.S.

UCLA Students Under Measles Quarantine

Which Doctors Make the Most Money?

More Kids Having 'Tommy John' Surgery

2. Organizing the Content: Information Architecture and Application Structure



Figure 2-2. WebMD

Make Yahoo Your Homepage
Discover something new every day from News, Sports, Finance, Entertainment and more!

Get No, thanks

YAHOO! Search Sign in Mail

Mail News Finance Sports Politics Entertainment Lifestyle Good News More...

Trending Now

1. Tiger Woods	6. Car Insurance Quo...
2. Channing Tatum	7. Little Richard
3. Robin Lehner	8. Kim Jong-un
4. Nipsey Hussle	9. VA Loan
5. Storage Near Me	10. Shannen Doherty

Top stories

Man's 9-year battle to get his lung disease diagnosed

"My mental state was really messed up," 74-year-old Ken Benson says of his struggle to get his COPD properly diagnosed.

Perpetual shortness of breath

239 people reacting

WALK FOR A REASON CURE ALZHEIMER'S

Top worry for Americans remains the same Synagogue shooting suspect linked to mosque arson Fed Board nominee Stephen Moore apologizes From Leo to 'Lion King': 30 movies you can't miss Warren Buffet: Why college isn't for everybody

YAHOO! MAIL
Faster, beautiful and intuitive.

Learn more

Politics Reuters
Former U.S. deputy attorney general: If Trump was not president, he would be indicted

Former top Justice Department official Sally Yates said on Sunday that if Donald Trump were not president, he would have been indicted on obstruction charges in the Russia investigation.

AG William Barr won't testify under proposed format, DOJ tells House committee: Alida Good Morning America

Full Sally paints a devastating portrait' Yahoo View

1307

Don't like this news source? Customize here

San Francisco, CA

Today	Mon	Tue	Wed
58° 52°	58° 52°	58° 53°	61° 51°

Scoreboard

Sponsored Wells Fargo Checking - Member FDIC.

Wells Fargo Welcome Bonus of \$400

Open an eligible consumer checking account and receive qualifying direct deposits to qualify.

Houston 100 Final

2. Organizing the Content: Information Architecture and Application Structure

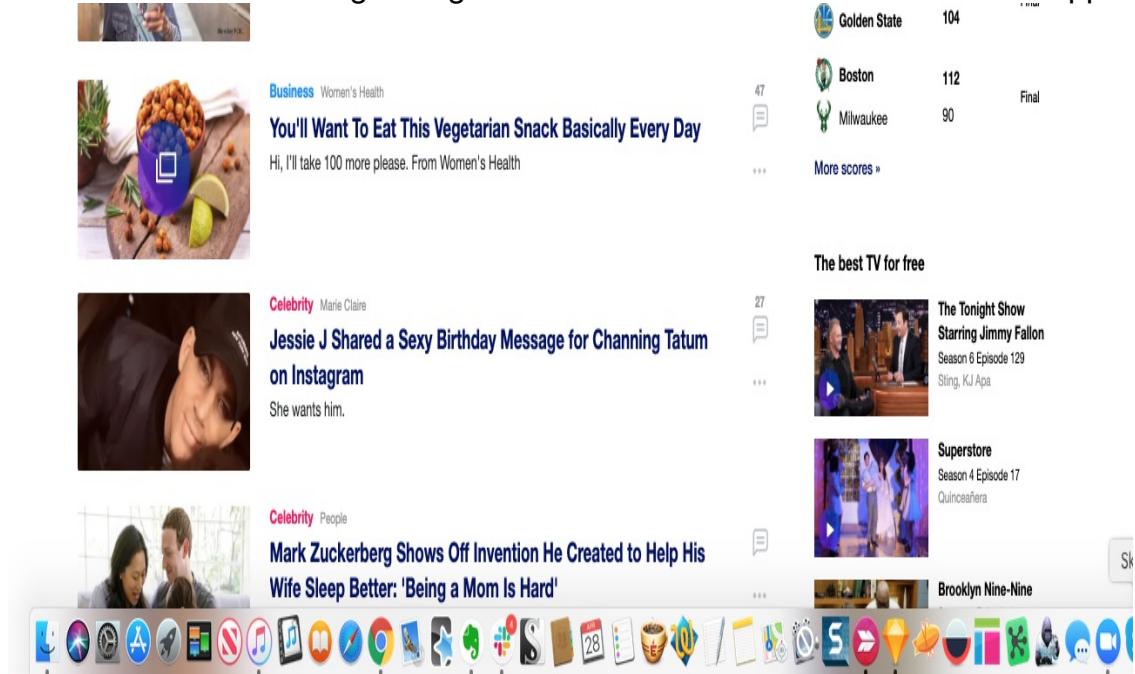


Figure 2-3. Yahoo

The screenshot shows the sheknows.com homepage with a grid layout. At the top, there's a navigation bar with a video icon and the text "sheknows" followed by a dimension of "1440 x 1451".

Below the navigation, there are two main article cards:

- Viola Davis On Diabetes, Motherhood & Asking For The Top**: An image of actress Viola Davis.
- My Body Can Have a Baby But My Finances Can't**: An illustration of a pink piggy bank.

At the bottom of the page, there's a search bar with the placeholder "What would you like to know?", a "Search" button, and social media links for Facebook, Twitter, Pinterest, and Instagram.

2. Organizing the Content: Information Architecture and Application Structure

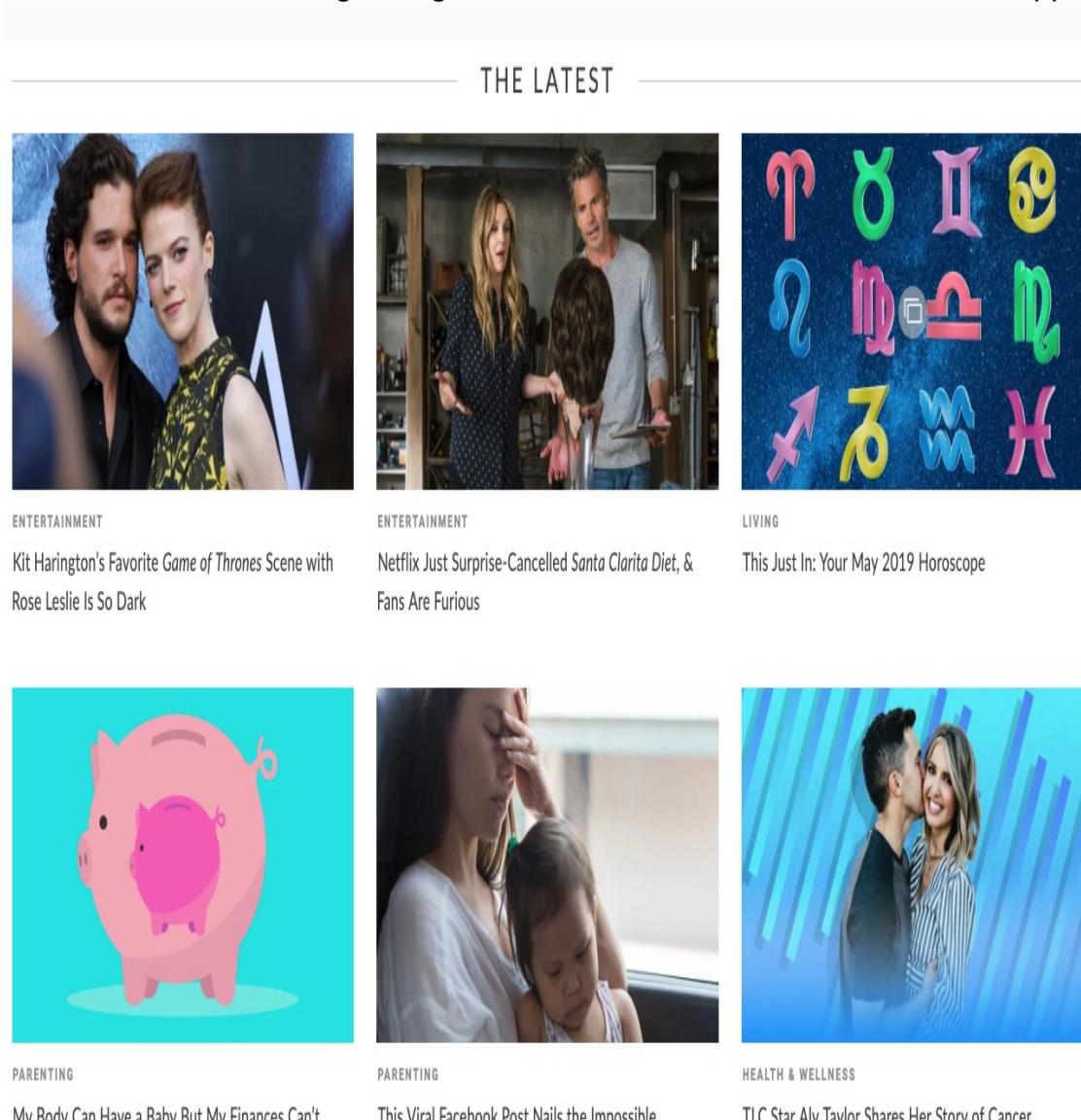
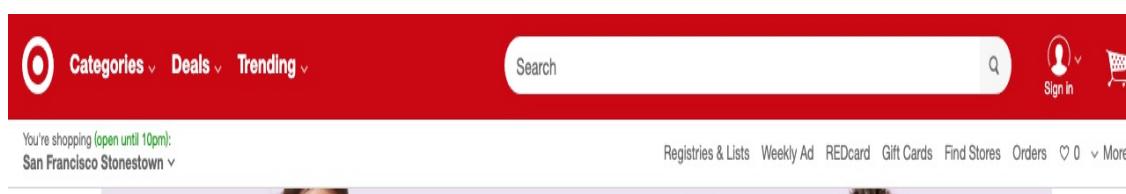


Figure 2-4. Sheknows

Section 2: Commerce-Centric Websites

Major retailers Target (Figure 2-5) and Ace Hardware (Figure 2-6) follow the same pattern. Search is at the top, with large features (sale promotions) below. Both sites support browse with grids of cards.



2. Organizing the Content: Information Architecture and Application Structure



20% off women's dresses*

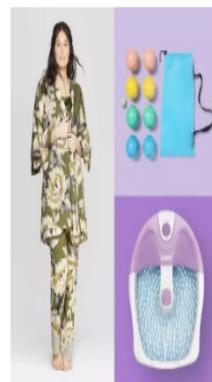


Hearth & Hand with Magnolia

Designs that Mom will love.



Great gift ideas



Mother's Day must-dos

Ways to celebrate that beat breakfast in bed.



New & only at Target



2. Organizing the Content: Information Architecture and Application Structure

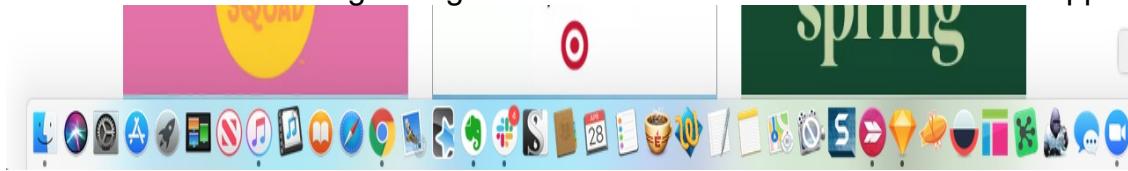


Figure 2-5. Target

We Deliver. Get what you need, when you need it. [Learn More](#)

Sign In / Register Customer Service Store Locator

ACE The helpful place.

What can we help you find?

ACE Rewards
The best tools for saving money.™

Shopping Cart

Departments The Paint Studio Sales & Specials Local Ad Tips & Advice

Duke's - San Francisco, CA Open until 6:00 PM | Store Info / Directions | Services / Brands | Change Store

BOGO SALE
April 24-29
Savings Online & In-Store

[Shop Now](#)

Ace Rewards members **Free Assembly & Delivery** on grills & grilling accessories totaling \$399 & up[†]

[†]Available at participating stores. Click [here](#) for details.

Our Newest Offers & Items

2. Organizing the Content: Information Architecture and Application Structure

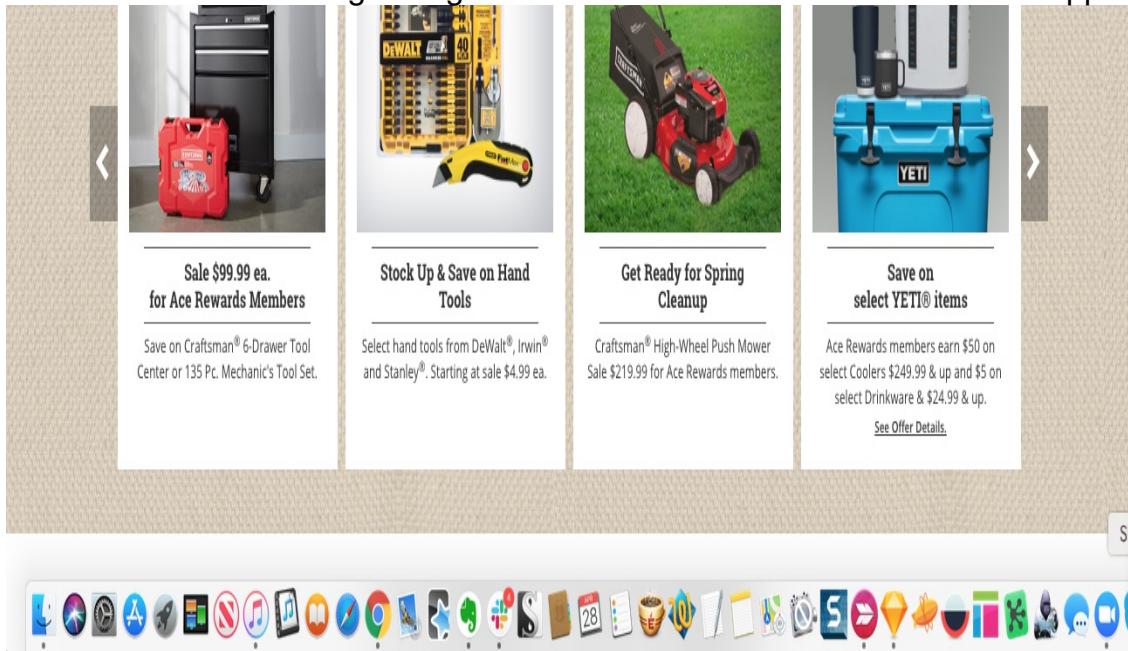
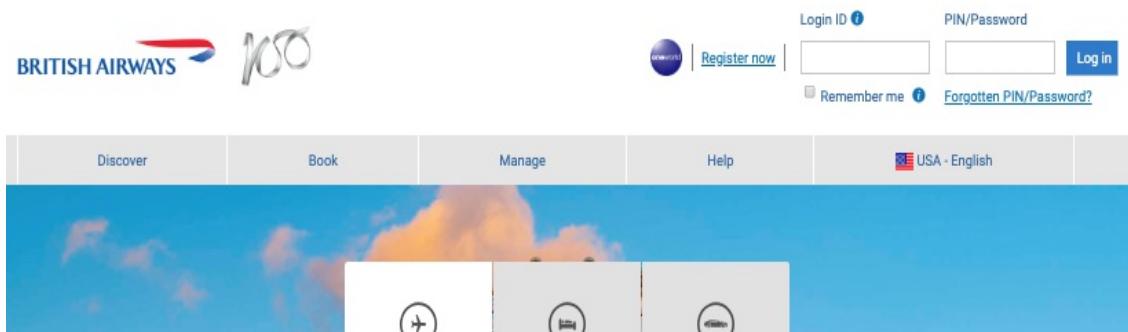


Figure 2-6. Ace Hardware

Section 3: Task-Centric Websites

These two examples show how Feature, Search, Browse changes when deployed in a context where getting the user to carry out a task (search) is the priority. For British Airways ([Figure 2-7](#)), a large travel search module dominates the entire screen above the fold. Below the fold, there is minimal content: A featured article and only three cards for browsing.

Epicurious ([Figure 2-8](#)) gives primacy to search as well. It occupies a screen-width panel above the fold. However, its content features and browse cards begin immediately below. Their large size and appetizing photos and titles give them almost equal weight to search. Based on your needs, you can decide which prioritized or balanced approach is right for your situation, and design your interface accordingly.



2. Organizing the Content: Information Architecture and Application Structure

The screenshot displays the British Airways website's search interface and promotional content.

Search Interface:

- Flight Options:** Flight, Flight + Hotel, Flight + Car.
- Search Form:** Flights Multi-city. Fields include: From (type 3 letters), To (type 3 letters), Outbound (28 April), Add a return (+), Passengers (1 Adult), Travel class (Economy), and a pink Search button.
- Optional extras:** A link labeled "Optional extras" is visible.

User Navigation:

- Manage My Booking
- Check in
- Check flight status

Promotional Content:

- Login Area:** Log in to see your offers and bookings. Includes fields for Login ID / Username and PIN/Password, and a red Log in button. A link for Forgotten PIN/Password? is also present.
- 100th Anniversary:** A large banner celebrating 100 years of British Airways. It features a man writing the number "100" on a wall. Text: "100 years of British Airways" and "We're celebrating our landmark birthday by celebrating Britain in our new video." A "Watch now" button is available.
- New User Registration:** Text: "New to ba.com? Register for free". A list of benefits: Get personal offers, Manage your bookings in one place, Save details for easier bookings. A "Register now" button is available.
- Vacation Offers:**
 - London vacations (image of Tower Bridge)
 - Paris vacations (image of Eiffel Tower)
 - Save on fly-drive packages (image of a road through a green landscape)
- Car Rental Offer:** Enjoy the freedom and convenience of having a car on your next vacation. Combine and save with a flight + car package and get 7 days car rental for the price of 5 in select destinations on your next trip. Book by April 20, 2010.

Figure 2-7. British Airways

2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows the homepage of the Epicurious website. At the top, there is a navigation bar with a menu icon, a heart icon for logging in or signing up, the "epicurious" logo, social media links for Follow (Facebook, Twitter, Instagram), and a search bar labeled "Search". The main header features a large, bold "Find a Recipe" text over a background image of various fresh herbs and flowers. Below the header is a search bar with a magnifying glass icon and an "ADVANCED SEARCH" link. A Williams Sonoma advertisement for "Natural Style" is visible, with a "SHOP NOW >" button and images of wreaths. The main content area includes a "Family Dinner Wins" article by Anya Hoffman, a "RECIPE ROUNDUP" section for spring side dishes, and a "TIPS AND TRICKS" section for cutting chicken. To the right, there is a teal-colored sidebar for "2 MEDICINES" and "1 PILL" related to HIV treatment, with a "LEARN MORE" button and the ViiV Healthcare logo.

Log in/Sign up

epicurious

Follow Search

Find a Recipe

ADVANCED SEARCH

WILLIAMS SONOMA Natural Style

SMALL PLATES

Family Dinner Wins: 5 Easy, Back-Pocket Meals That Save Us From Takeout

BY ANYA HOFFMAN

RECIPE ROUNDUP

63 Side Dishes That Scream Spring

BY THE EPICURIOUS EDITORS

TIPS AND TRICKS

How to Cut a Chicken in Half and Win the Crispy-Skin Game Every Time

©2019 ViiV Healthcare or its licensor. ARI0000000000 and 2019-Presence in USA.

2 MEDICINES

1 PILL

ZERO REASONS —
to wait to ask your doctor about
an HIV-1 treatment option

LEARN MORE

2. Organizing the Content: Information Architecture and Application Structure

BY JOE SEVIER



Figure 2-8. Epicurious

Search with Facets and Filters

These examples show how two different web sites use multiple categories, or facets, to help the user create a targeted search for large data sets. Each facet can have a range of values. Combinations of facets for search or filtering allows for sophisticated searching of large data sets.

Crunchbase ([Figure 2-9](#)) is unusual in that it promotes faceted filters as part of its search submittal, implying that the searcher can get better, more appropriate search results by using the facets. Featured content is below.

Epicurious ([Figure 2-10](#)) and AirBnB ([Figure 2-11](#)) both use a more traditional deployment of faceted filters on their search results screens (note the mobile-friendly grid of cards format for search results in both cases). The facets are most relevant for narrowing the search results based on their domain.

A screenshot of the Crunchbase website. At the top, there's a blue header bar with the "crunchbase" logo, navigation links for "Solutions", "Products", "Resources", and "Pricing", a search bar with the placeholder "Search Crunchbase", and "LOG IN" and "REGISTER" buttons. On the left, a sidebar titled "Crunchbase Pro" lists various search filters: "Companies", "People", "Investors", "Funding Rounds", "Acquisitions", and "Schools". The main content area has a teal background. It features the text "Discover innovative companies and the people behind them" and "Select up to two filters to start your search:". Below this are five filter buttons: "Location +", "Categories +", "Number of Employees +", "Total Funding Amount +", and "Last Funding Date +". A large green "SEARCH NOW" button is at the bottom of this section. At the very bottom, there's another section titled "Do even more with Crunchbase".

2. Organizing the Content: Information Architecture and Application Structure

[Events](#)

[Hubs](#)

[My Searches](#)

[My Lists](#)

[Marketplace](#)

[Add New Profile](#)

Find prospects

Discover the right leads and close more deals, faster

[LEARN MORE >](#)

Find investments

Identify and track innovative companies in your target sector

[LEARN MORE >](#)

Find investors

Research investors and monitor competitors all in one place

[LEARN MORE >](#)

Conduct market research

Find emerging trends and analyze changes across industries

[LEARN MORE >](#)

Elevate your search with Crunchbase Pro

Intelligently search and target the right opportunities with advanced search filters, analysis tools, and company alerts.

[START FREE TRIAL](#) [LEARN MORE](#)

Already have an account? [Log in here](#)

Crunchbase insights & analysis

Latest insights and analysis

April 26, 2019 [Last Week In Venture: Better ER Software, More AI, and More](#)

Activity this week

310 Funding rounds announced

8.9B Total Funding

[? help](#)

Figure 2-9. Crunchbase

61

2. Organizing the Content: Information Architecture and Application Structure

POPULAR ▾ MEAL & COURSE ▾ DISH TYPE ▾ DIETARY CONCERN ▾ INGREDIENT ▾ CUISINE ▾ HOLIDAY ▾ TECHNIQUE ▾

107 matching results for "enchiladas"

Vegetarian
 Quick & Easy
 Kid Friendly
 Healthy
 Wheat/Gluten-Free
 Vegan
 Low Fat
 Low/No Sugar

SHOW: All Content ▾ SORT BY: Relevance ▾

ARTICLE
Weekly Meal Plan: April 8-12
 Zingy pineapple shrimp, sheet-pan tandoori chicken, and the world's easiest...

RECIPE
Enchiladas Divas

ARTICLE
Weekly Meal Plan: March 26-30
 Cuban-style grain bowls, hearty shrimp fajitas, and the easiest chicken...

ARTICLE
Weekly Meal Plan: October 15-19
 We're keeping it vegetarian this week with big-batch chili, "nextover" tomato...

GALLERY
102 Mexican Recipes We Love
 From smoky, chile-laced enchiladas to creamy, coffee-spiked desserts, these a...

GALLERY
How to Throw a Cinco de Mayo Party
 41 recipes worthy of a Mexican fiesta.

ARTICLE
Big-Batch Ground Beef Without All the Work
 Make enough meat for a week's worth of meals in less than half an hour.

VIDEO
Cheesy Chicken Enchilada Skillet
 Learn how to make a Cheesy Chicken Enchilada Skillet! Recipe here:....

GALLERY
Our 31 Best Leftover Turkey Recipes
 These soups, enchiladas, salads, and sandwiches made from leftover turkey...

ARTICLE
PROGRESSIVE Chat. Click. Call.
Your quote, your way
 Connect with Flo on Messenger
 Get a Quote

GALLERY
35 Amazing Thanksgiving Leftover Recipe Ideas
 These recipes—from ramen and gumbo to enchiladas and pancakes—make...

ARTICLE
Cheesy Chicken Enchilada Skillet

ARTICLE
Our 31 Best Leftover Turkey Recipes

GALLERY
35 Amazing Thanksgiving Leftover Recipe Ideas

ARTICLE
Cheesy Chicken Enchilada Skillet

2. Organizing the Content: Information Architecture and Application Structure

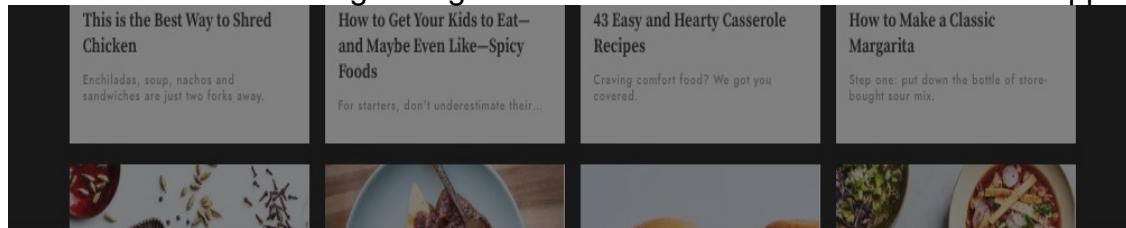


Figure 2-10. Epicurious search results with faceted filters

 Q Los Feliz, Los Angeles · Homes

Become a host Help Sign up Log in

Dates Guests Home type Price Instant Book More filters Show Map X

- Entire place
Have a place to yourself
- Private room
Have your own room and share some common spaces
- Hotel room
Have a private or shared room in a boutique hotel, hostel, and more
- Shared room
Stay in a shared space, like a common room

[Apply](#)



PLUS VERIFIED - ENTIRE APARTMENT

Elegant Design Apt with a View of the LA River

\$109 per night

★★★★★ 144



VERIFIED - ENTIRE APARTMENT

Charming Studio Retreat in West Hollywood

\$119 per night

★★★★★ 199



PRIVATE ROOM - 1 BED

theMuse Mini Studio by Hollywood

\$49 per night

★★★★★ 182



ENTIRE APARTMENT - 1 BED

Cozy, Private Single in Los Feliz

\$96 per night

★★★★★ 152

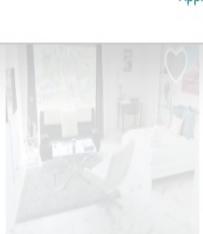


ENTIRE APARTMENT - 2 BEDS

Bright, Cozy Los Feliz Gem!

\$75 per night

★★★★★ 157

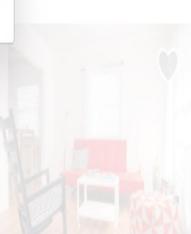


ENTIRE APARTMENT - 0 BEDS

★Stylish ★Parking ★Balcony ★Pet friendly

\$130 per night

★★★★★ 85



ENTIRE APARTMENT - 2 BEDS

Charming Los Feliz Apt w/ parking

\$125 per night

★★★★★ 144

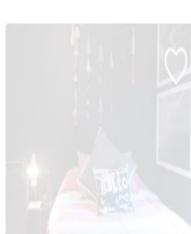


ENTIRE APARTMENT - 1 BED

Sweet Garden Studio in the heart of Silver Lake

\$110 per night

★★★★★ 140 - Superhost

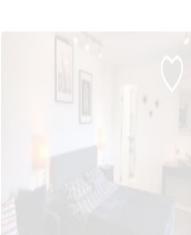


PRIVATE ROOM - 1 BED

Your own bedroom in Eastside LA

\$68 per night

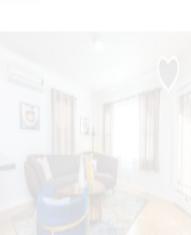
★★★★★ 220 - Superhost



ENTIRE APARTMENT - 1 BED



PLUS VERIFIED - ENTIRE APARTMENT



ENTIRE APARTMENT - 2 BEDS



ENTIRE APARTMENT - 1 BED



ENTIRE APARTMENT - 1 BED

2. Organizing the Content: Information Architecture and Application Structure

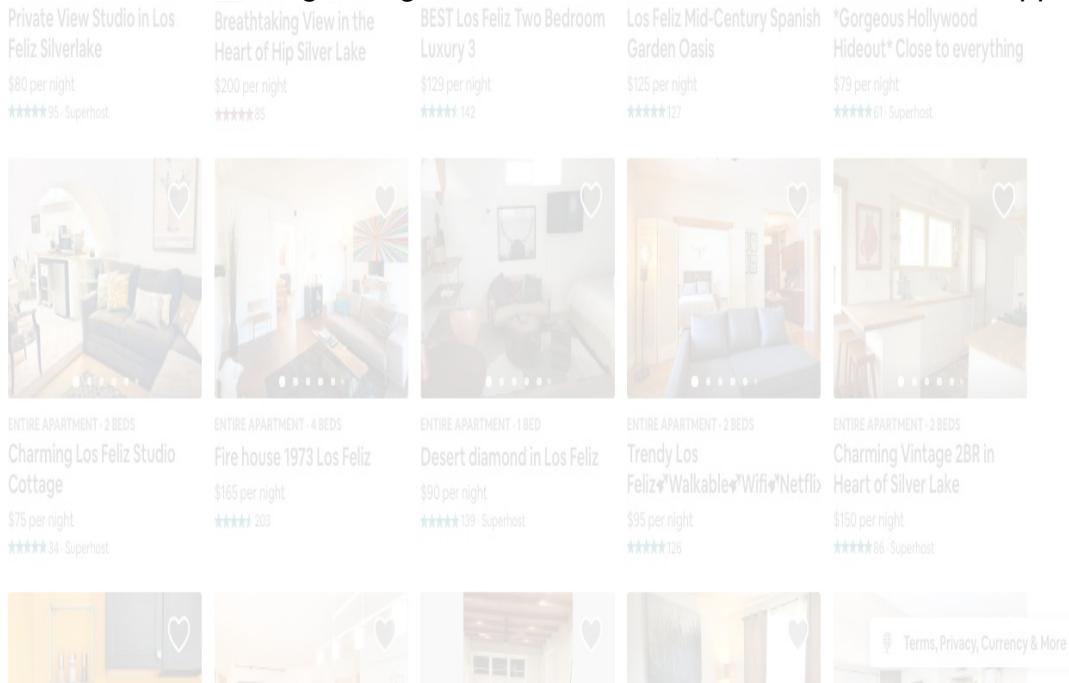


Figure 2-11. Airbnb search results with faceted filters

Mobile Direct Access

What

Launch the mobile application so that the primary function of the app is already active or the first screen presents actionable information without requiring any input or action from the user. This is often achieved by taking advantage of location and time information from the user's mobile device. Making assumptions about what the user is most likely to do also drive what appears on launch. The user is able to immediately take action without having to do any configuration or input.

Use when

Your mobile app generates value by doing one thing really well, or is used or known for one primary thing.

Why

Starting the user with an immediately actionable mode, choice or screen satisfies the need

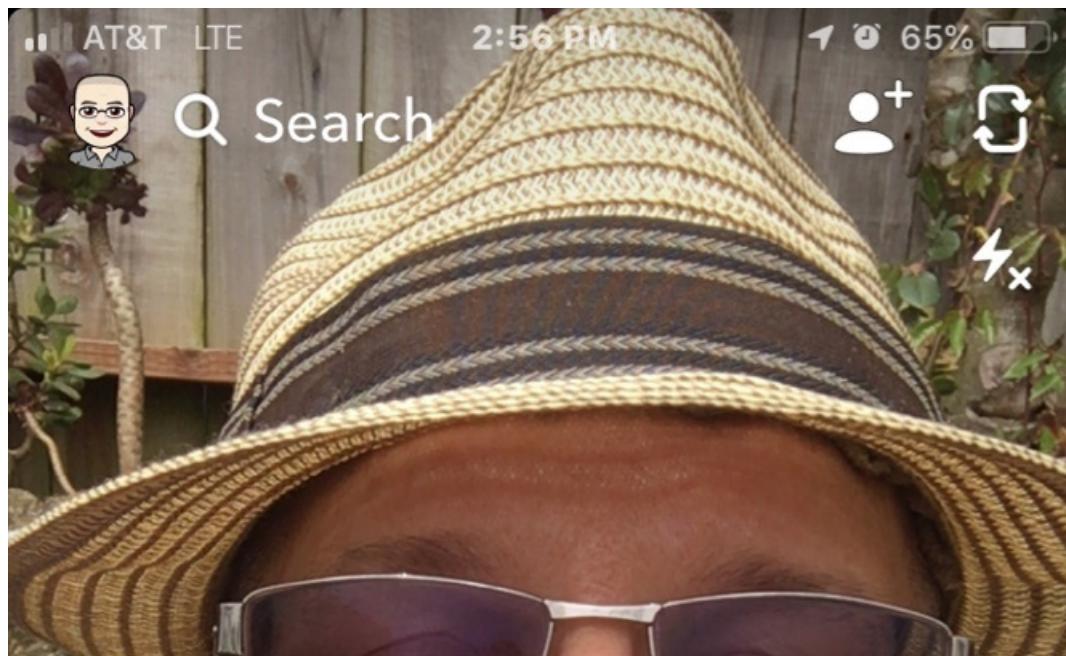
2. Organizing the Content: Information Architecture and Application Structure to get the user engaged or to provide value in the mobile context. Specifically, there is limited ability to enter search or any kind of choice or configuration. On the other hand, there is useful information available from the device and the primary use cases to make this assumption highly likely to be valuable, even expected.

How

Use live data from the user's mobile device (assuming the user has given permission in the settings). Primarily, look at location and time to generate a meaningful landing screen for the user. In addition, make assumptions about what the user is most likely to be doing with your app, and get them as close to completing the action as possible, with a minimum of input.

Examples

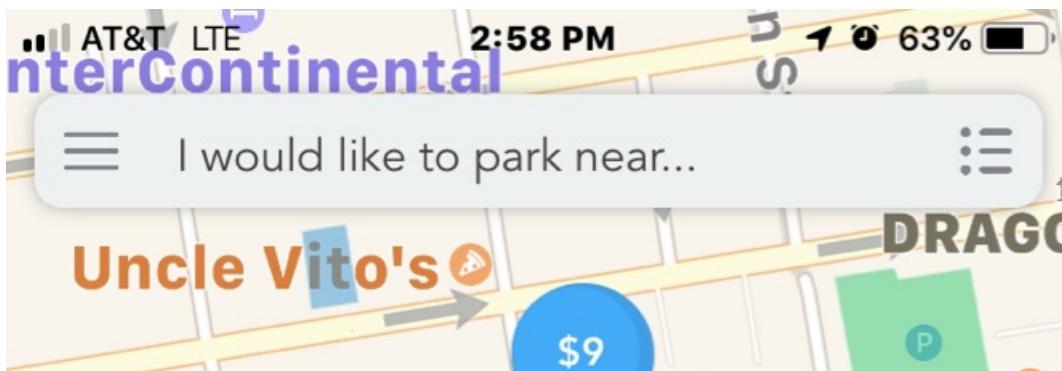
All of these examples are mobile but could be valuable in desktop settings. In the first example, Snap ([Figure 2-12](#)) lives up to its brand as a photo-centric social media and camera company. When the app is launched, the user-facing camera is automatically turned on, ready to take a selfie. The next three examples all show how the apps use location and time data to return meaningful results with no input from the customer. Inrix ParkMe ([Figure 2-13](#)), Eventbrite ([Figure 2-14](#)) and Weatherbug ([Figure 2-15](#)) give useful results this way. ParkMe makes some smart prefill assumptions (park for 1 hour) to get price results by default.



2. Organizing the Content: Information Architecture and Application Structure



Figure 2-12. Snap start screen



2. Organizing the Content: Information Architecture and Application Structure

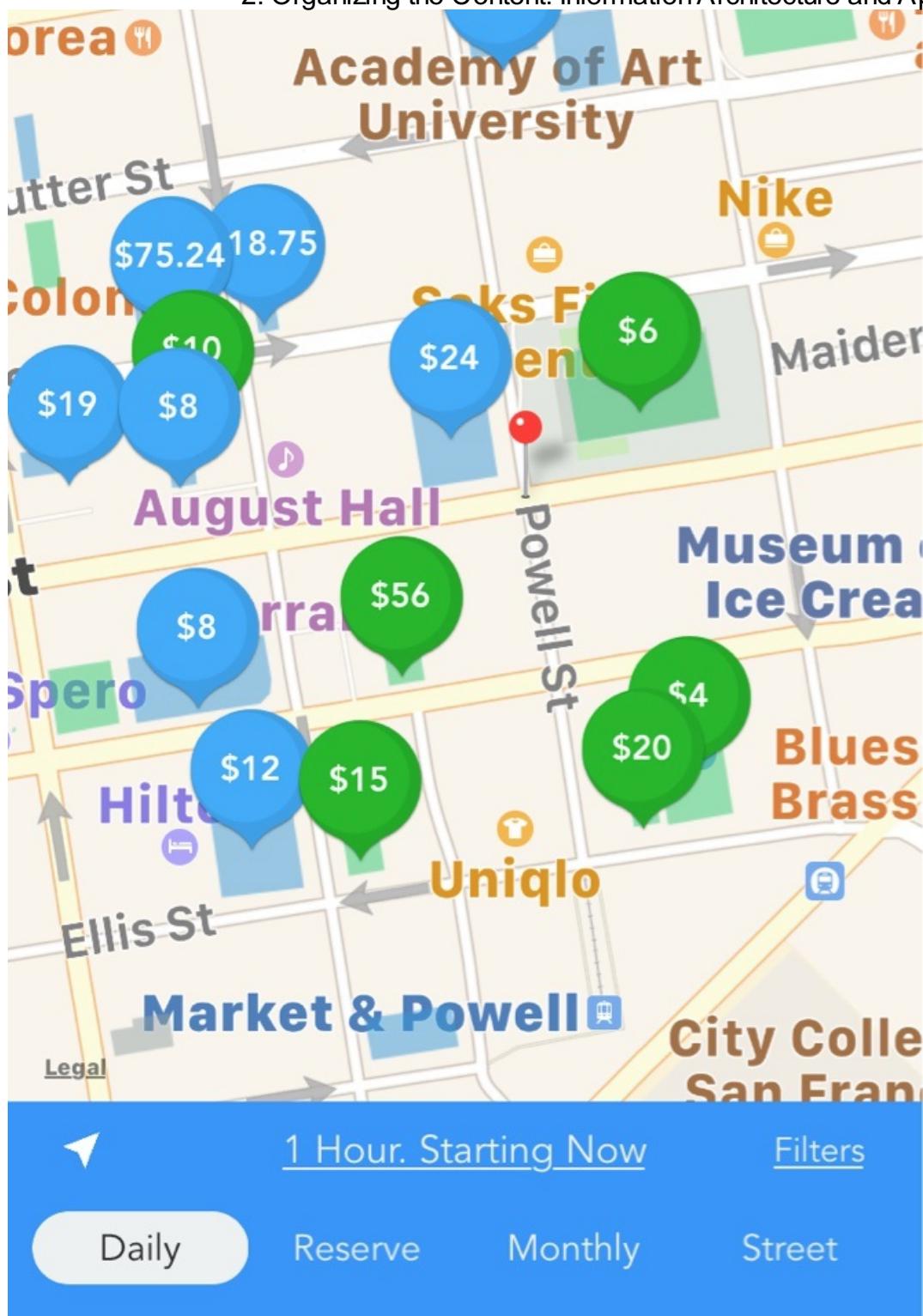


Figure 2-13. Inrix ParkMe start screen



What's good in
San Franci...



Editor's Picks





APR 26 stARTup Art Fair SF 2019
Hotel del Sol



2019 Lexus
Culinary Classic

April 26–28 • 6:00 PM PDT
Cavallo Point Lodge



2. Organizing the Content: Information Architecture and Application Structure



Figure 2-14. Eventbrite start screen





Figure 2-15. Weatherbug start screen

Streams and Feeds

What

A continuously updated series of images, news stories, web articles, comments or other content presented in a scrollable, vertical (sometimes horizontal) strip or ribbon. The items in the feed are usually presented as “cards,” with an image from the article, a headline, introductory copy, and the name of the source with a link. Comments will feature the name and avatar or photo of the poster. From its origins as a social media innovation, this pattern has become a common one for any company, app or site that publishes or aggregates large amounts of content continuously, and for social media, and for business collaboration software.

News/Content Streams

Show time-stamped items in a reverse chronological list, with the latest items at the top. Update it dynamically. Content can be first party (publish your own content), or aggregated from third parties. This news stream approach, inspired by social media, is an established, mobile-friendly pattern for publishers.

Social Streams

Show time-sensitive items in a reverse chronological list, with the latest items at the top. Update it dynamically. Content comes from other members that the user follows (plus sponsored content).

Business Collaboration

Social media technology has now integrated into the way we work online. It is common for companies of all types to use online tools to collaborate. This allows employees and teams from any location to still come together online for discussion and feedback. Social media style comment feeds are one of the most common patterns,

Use when

Your site or app has frequently updated content or and the user checks it often, often dozens of times a day. Also use this pattern when you have multiple collaborators on a project and you need to stay on top of comments and feedback from multiple people. This feedback often comes asynchronously, meaning different people give feedback at different times. This is especially common in distributed or remote teams. For news publishers and aggregators, use one or more source channels, such as own original content, blogs, major news sites, other social site updates, and content partners to deliver timely content to users.

The pure social version may be personal—a user “owns” it, as for social media sites, like Twitter or Facebook friends list.

For business collaboration software, use this pattern to allow people to view, comment and edit a document. The document and the discussion are presented together. Employees can scroll through the comment feed to see the history of the discussion.

Why

Ensures that new content is always appearing at the top of the user’s feed. This makes each visit a reward, where there is something new to see and to scroll through. People can keep up with a news stream easily, since the latest items reliably appear on top with no effort on the part of the user. This promotes the habit of checking back frequently and spending lots of time reading, following and interacting with their stream.

People go to many sites or apps each day to keep up with their friends’ activities, engage in conversations, or follow topics or blogs of interest. When multiple “news” sources can be blended in one place, it’s easier to keep track of it all.

From the perspective of a publisher, such as a news website, publishing own content in

2. Organizing the Content: Information Architecture and Application Structure

the feed or stream format promotes engagement, return visits, and interaction.

From the perspective of a business, social collaboration software allows employees to be more efficient, and to save time. Remote workers and employees in different locations and time zones can still come together asynchronously to get work done.

This pattern supports the “Microbreaks” behavior pattern in Chapter 1. A glance at a “News Stream” application can give a user lots of useful information (or entertainment) with very little time or effort.

How

List incoming items in reverse chronological order. Display newest items onto the top of the list without waiting for the user to request an update. Older items get pushed down by the newer comments or entries. Offer a way for the user to get an immediate update or refresh. Also, they need to be able to scroll or review through the list to get to the older, unreviewed items.

Offer publisher-curated streams that the user can view in addition to their own social stream. Offer advanced users the ability to create custom streams based on topics or curated lists of other members. Others, such as Tweetdeck, use “Many Workspaces” to show multiple parallel panels of incoming content.

Information shown with each item might include:

What

For short micro-updates, show the whole thing. Otherwise, show a title, a teaser that’s a few words or sentences long, and a thumbnail picture if one is available.

Who

This might be the person who wrote an update, the blog where an article was posted, the author of said article, or the sender of an email. It could be the coworker who posted a comment or document. Actual person names humanize the interface, but balance this against recognition and authoritativeness—the names of news outlets, blogs, companies, and so forth are important, too. Use both if that makes sense.

2. Organizing the Content: Information Architecture and Application Structure

When

Give a date or timestamp; consider using relative times, such as “Yesterday” and “Eleven minutes ago.” As the post ages, switch to the traditional date & time stamp.

Where

If an item’s source is a website, link to that website. If it comes from one of your organization’s blogs, link to that.

When there’s more to an item than can be shown easily in the list display, show a “More” link or button. This is a good pattern for long comments. For news or story cards, allow the reader to click on the card to load the full story as a new screen. You might design a way to show the entire contents of an item within the “News Stream” window. The “News Stream” is a list, so you can choose among “Two-Panel Selector”, “One-Window Drilldown”, and “List Inlay”. Examples abound of each model.

Give the user ways to respond immediately to incoming items. Stars, thumbs-up, liking, and favoriting are available in some systems—these all provide low-effort feedback and “handshaking” among people who don’t have time to write out long replies. But allow those long replies to be written, too! By placing controls and text fields immediately next to an item in a “News Stream”, you encourage responsiveness and interaction. This is usually a good thing in social systems.

Sharing of items, either privately via email or semi-publicly via a provided social service, is also common in these interfaces. See the “Sharing Widget” pattern in Chapter 9.

“News Stream” designs for mobile devices are fairly straightforward as of this writing. Almost all of them devote the full screen to a single list—often a “Thumbnail-and-Text List” (Chapter 10) with richly formatted text—and users can drill down to an item by simply tapping or clicking it in the list.

Many “News Stream” services, including Twitter and Facebook, use the “Infinite List” pattern (see Chapter 10) for both their mobile and full-screen designs. This pattern lets users load a page or two of the most recent updates, and gives the option of loading more to go “backward in time.”

Activity History

2. Organizing the Content: Information Architecture and Application Structure

Some resources use the term activity stream for a very closely related concept: the time-ordered stream of actions (usually social actions) performed by a single entity such as an individual, system, or organization. It is a history of their actions. This is a useful concept, and it doesn't really conflict with the "News Stream" pattern, which talks about the stream of activities that are of interest to an individual or group of users, not generated by them. "News Stream" will usually have multiple diverse sources.

Examples

News/Content Streams



Startups

Apps

Gadgets

Videos

Podcasts

Extra Crunch^{NEW}

Events

Advertise

Crunchbase

More

Tesla

Fundings & Exits

Google

tc sessions robotics 2019

Search

Week-in-Review: Tesla's losses and Elon Musk's new promises

10 hours ago Lucas Matney

What a complicated week for Tesla. The electric car-maker announced this week that it had lost more than \$700 million in the first quarter of 2019, an unpleasant surprise for investors that came du...



Sponsored Content Results Are In: Best Travel Credit Cards Of 2019

Sponsored by CompareCards.com

Take advantage of no annual fees + travel rewards. Get up to 10x miles, 20,000 bonus points, all with no annual fee.



Meet the tech boss, same as the old boss

11 hours ago Jon Evans

"Power corrupts, and absolute power corrupts absolutely." It seems darkly funny, now, that anyone ever dared to dream that tech would be different. But we did, once. We would build new ...



Bad PR ideas, esports, and the Valley's talent poaching war

1 day ago Danny Crichton

Sending severed heads, and even more PR DON'Ts I wrote a "master list" of PR DON'Ts earlier this week, and now that list has nearly doubled as my fellow TechCrunch writers continued to experience e...



2. Organizing the Content: Information Architecture and Application Structure

E Extra Crunch

Sending severed heads, and even more PR DON'Ts

1 day ago Danny Crichton

This week, I published a piece called the "The master list of PR DON'Ts (or how not to piss off the writer covering your startup)." The problem, of course, with writing a "master list" is that as s...



Sponsored Content

Results Are In: Best Travel Credit Cards Of 2019

Sponsored by CompareCards.com

Take advantage of no annual fees + travel rewards. Get up to 10x miles, 20,000 bonus points, all with no annual fee.



Original Content podcast: 'Game of Thrones'

We're barely more than 24 hours away from what's widely expected to be the most spectacular and



Figure 2-16. Techcrunch

TC

Avengers: Endgame becomes the first film to break \$1 billion in an opening weekend

Login

Jonathan Shieber @jshieber / 6 hours ago

Comment

Startups

Apps

Gadgets

Videos

Podcasts

Extra Crunch NEW

Events

Advertise



2. Organizing the Content: Information Architecture and Application Structure

Crunchbase

More

Tesla

Fundings & Exits

Google

tc sessions robotics 2019

Search 

In its opening weekend, "Avengers: Endgame" made [breaking box office records](#) look like a snap.

The last film in what [Marvel Studios](#) dubbed phase three of its rollout of characters and plots in an ever-expanding cinematic universe is a box office marvel raking in an estimated \$1.2 billion at the box office.

Thanos will snap away your Google search results

Go to Google (don't worry, I'll wait). Type in "Thanos." Click the little cartoon Infinity Gauntlet on the right side. Now sit back in horror as Evil Space Grimace snaps half of your search results into dust in the wind. Call it an Easter egg or Snap Engine Optimization. It's a fun little promotion for the ... [Continue reading](#)



 TechCrunch

1 ↗

Benefiting from a \$350 million domestic debut and another \$859 million in global box office receipts, it's clear that the [Marvel Studios](#) franchise has achieved [super heroic returns for Disney](#) since its 2009 acquisition for [\\$4 billion](#).

"Avengers: Endgame" hit the billion-dollar mark in five days, faster than its predecessor [Avengers: Infinity War](#), which held the previous record at 11 days (but still not faster than a speeding bullet).

Starring deep breath): Robert Downey Jr., Chris Evans, Mark Ruffalo, Chris Hemsworth, Scarlett Johansson, Brie Larson, Jeremy Renner, Don Cheadle, Paul Rudd, Brie Larson, Karen Gillan, Danai Gurira, Chadwick Boseman (fleetingly), Bradley Cooper and Josh Brolin; the film was an exercise in fan service, but also [a thrilling and moving way to say goodbye to the current crop of Earth's mightiest heroes](#), according to our reviewer, Anthony Ha



Figure 2-17. Techcrunch Detail / Individual Article screen

Techcrunch ([Figure 2-16](#) and [Figure 2-17](#)) is a great example of a news stream publisher. The main mobile app and web site is a scrollable stream of stories, with the most recent stories at the top. At this level, the reader is provided with just enough information to get the main idea: A photo, a headline, and some introductory copy. If the reader selects the story, they go to the full version, with larger images and full text. This detail page is where the social sharing feature is available, promoting distribution to the reader's own

2. Organizing the Content: Information Architecture and Application Structure

social networks.

BuzzFeed News

REPORTING TO YOU

ABOUT US GOT A TIP? SUPPORT US BUZZFEED.COM SECTIONS ▾

TRENDING Synagogue Shooting Avengers: Endgame Taylor Swift Climate Change Game Of Thrones 2020 Election Measles

Science

Reckoning With Personal Responsibility In The Age Of Climate Change

As someone who loves traveling and going outdoors, I struggle with balancing my hopefulness and my despair — and my culpability — regarding an imperiled earth.

Shannon Keating • 1 day ago

Can You Work Out How To Spend Your Money To Slow Global Warming?

It's easy to feel helpless in the face of climate change, but how you spend your money can make a difference.

Peter Aldhous • 1 day ago

How Much Do You Know About Climate Change?

When Yale University quizzed US adults on the facts, most of them got a failing grade. How do you compare?

Peter Aldhous • 2 days ago

17 Books That Will Change The Way You Think About The World

From Naomi Klein to Barbara Kingsolver, these authors explain the consequences of our warming planet — and imagine its future.

Arianna Rebolini • 2 days ago

These Haunting Pictures Show How Chernobyl Has Aged Over The Years

ADVERTISEMENT

Skip the trip to the store
Save \$10 on your first order

amazonfresh

Restrictions apply

SCIENCE MASTHEAD

Virginia Hughes	Twitter Email
Science Editor	
Peter Aldhous	Twitter Email
Science Reporter	
Zahra Hirji	Twitter Email
Science Reporter	
Stephanie M. Lee	Twitter Email
Science Reporter	
Nidhi Subbaraman	Twitter Email
Science Reporter	
Dan Vergano	Twitter Email
Science Reporter	

ADVERTISEMENT

77

2. Organizing the Content: Information Architecture and Application Structure

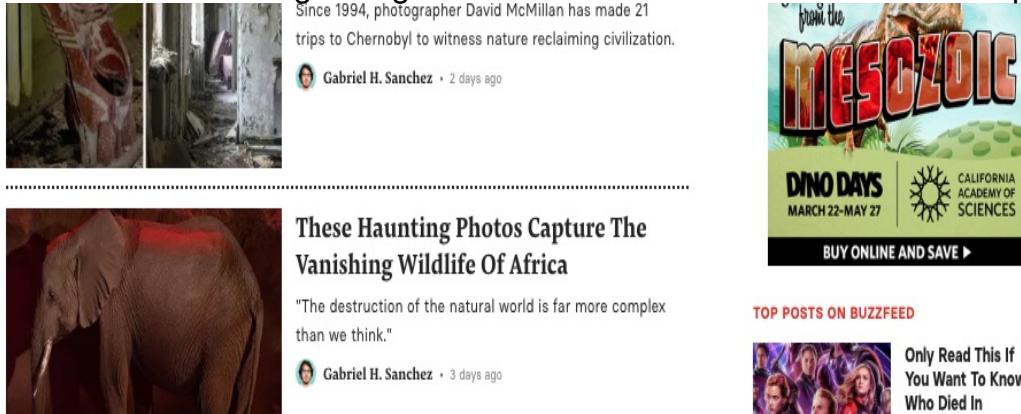


Figure 2-18. BuzzFeedNews

This screenshot shows the full BuzzFeed website. The top navigation bar includes links for News, Quizzes, Avengers, GoT, Mother's Day, Shopping, Videos, and Newsletters. Below the navigation is a horizontal banner of various trending images. A central feature is a quiz titled 'Can You Get 4/7 On This Week's Pop Culture Quiz?' with 28,381 views. To the right, there's an advertisement for the musical 'Beautiful: The Carole King Musical' with a poster for the show. The bottom of the page features social sharing icons for Facebook, Twitter, Email, Pinterest, Tumblr, and Google+, along with a green footer bar with various icons.

2. Organizing the Content: Information Architecture and Application Structure

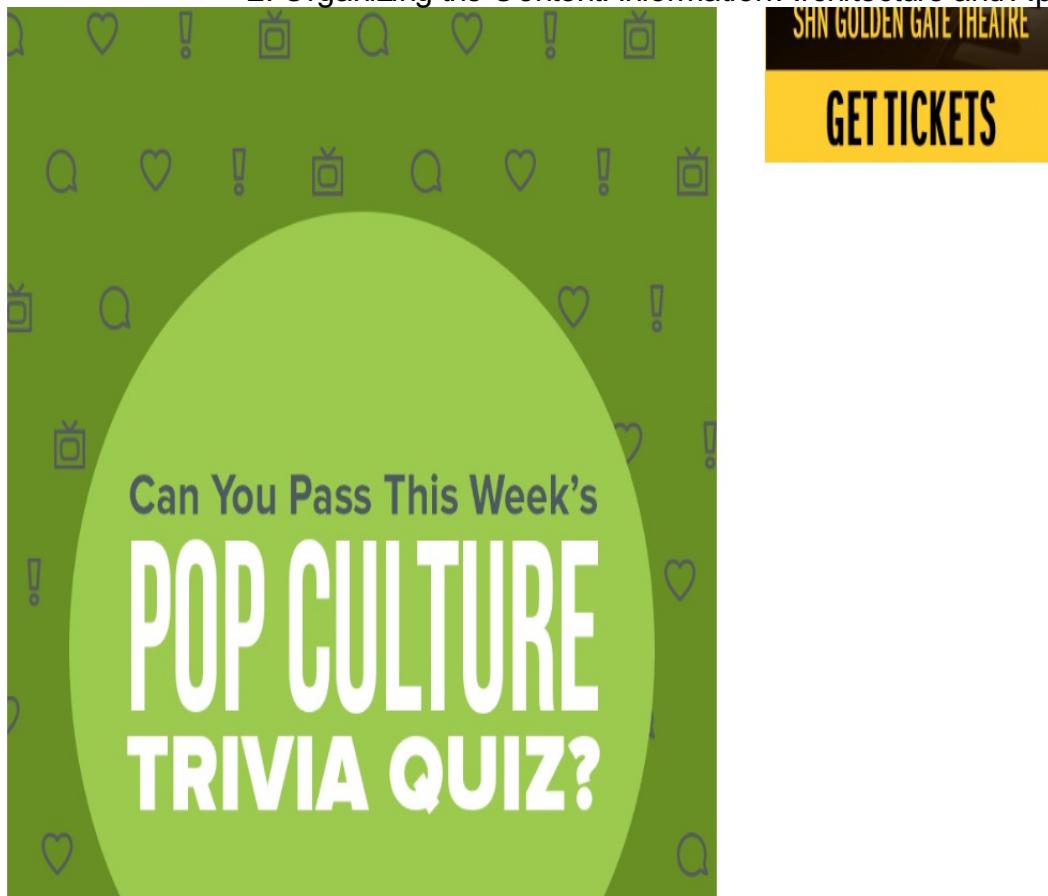
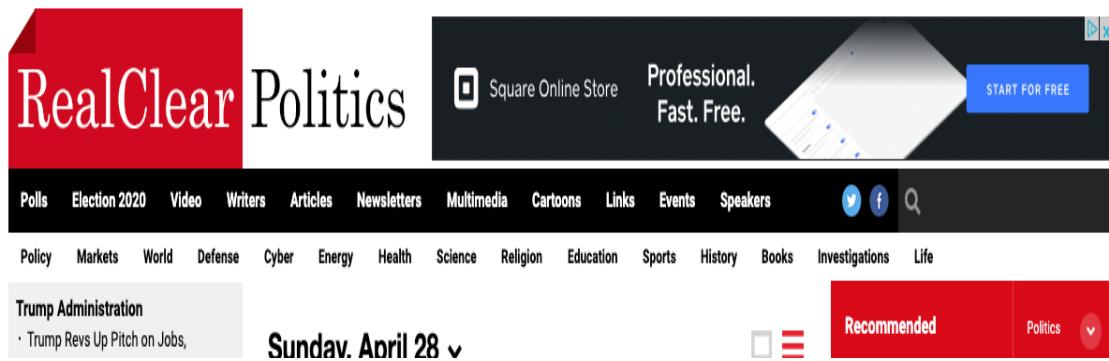


Figure 2-19. BuzzFeedNews Detail/Individual Article screen

BuzzFeedNews ([Figure 2-18](#) and [Figure 2-19](#)) follows exactly the same pattern. Note that this company uses the word “feed” in its name. This shows how important this pattern is to the company’s identity and purpose. Again we see a scrollable stream of stories, with the most recent stories at the top. BuzzFeed’s strong editorial voice is clear in the enticing headlines and reader-challenging questions. Selecting a story loads the detail screen, with the full story, quiz or gallery of images. The social sharing widget is even more prominent.



2. Organizing the Content: Information Architecture and Application Structure

Economy to Key Wisconsin Voters

- Replay: Rally in Green Bay, WI
- On Infrastructure, Trump and Dems May See Common Ground
- Trump Debates Waiving Jones Act to Ship Natural Gas to NE, Puerto Rico
- Charles Barkley Sneaked Into White House by Kushner for Meeting

Washington

- Barr Warns House Democrats He Might Not Appear at Hearing
- WHCD Focuses on Press Freedom and History, Not Stars and Comedy

Video Highlights

- Trump on Border Crossings: "It's Like Disneyland...It's a Disaster"
- Chernow Praises Media for "Noble Work" at Correspondents Dinner
- Moore Apologizes for Bad Jokes, Says Smear Campaign Should Stop
- Maher: Should Those Who Don't Go to College Pay for Those Who Do?

2020 Democratic Nomination

- Democratic Candidates Seek Union Support at Workers' Forum
- Why the Biden-Union Bond May Slacken in the Age of Trump
- How Biden, Obama's "Odd Couple" Relationship Aged Into Family Ties
- Can a Woman Win in 2020? Some Democrats Wonder If It's Risky

U.S. Economy

- Overcoming Doubts, Economy Finds Way Forward With Strong Growth
- GDP Data Illustrate Both the Good and Bad of Trump's Trade Battles

Political Landscape

- FL Gov. DeSantis Defies Republican Orthodoxy w/Drug Importation Plan
- NY Attorney General Investigates the NRA's Tax-Exempt Status
- Oliver North Steps Down as NRA President After Leadership Dispute

Media & Politics

- NY Times Apologizes for Cartoon Depicting Anti-Semitic Tropes
- Trump: Fox's Napolitano Wanted High Court Appt., Pardon for Friend

REAL CLEAR POLITICS FOR IPAD



The Media Has	Saturday, April 27	Action Joel Kotkin, New Geography
Trump Has Son		Lead of 2020 Sabato & Kondik, Post & Courier
Biden Is Normal	Friday, April 26	Good to Me Ana Navarro, CNN
Can a Woman Win	Thursday, April 25	Under If It's a Risk Lauren Egan, NBC News
An Indictment in	Wednesday, April 24	New York Review of Books
Charlottesville		Albert E. Lee Nicole Hemmer, CNN
Why the Dem M	Tuesday, April 23	Is as Racist Karl Notturno, Am. Greatness
A Despicable C	Monday, April 22	Stephens , New York Times
Sunday Panels:	Sunday, April 21	Nation, This Week , FOX News, CNN

Biden-Union Bond May Slacken in the Age of Trump

Trump Cheers Economy, Criticizes Democrats at Wisconsin Rally

Chernow Praises Media for "Noble Work" at Correspondents Dinner

U.S. Economy Grew at Strong 3.2% Rate in First Quarter

The Free-Stuff Primary: What Democrats' Promises Will Cost

Presidential Job Approval

Cartoons of the Week

RCP Afternoon Edition

20 Dems Are Vying for 2020. Can One Beat Trump? Amy Sorkin, The New Yorker

Will Barr Keep His Promise to Investigate Spying? Michael Goodwin, New York Post

Mueller Proved Comey Told the Truth About Trump William Saletan, Slate

What I Saw at Middlebury College Dominic Aiello, Quillette

Democrats Hunt for an Economic Argument to Counter Trump David Siders, Politico

Will Biden's History Lift Him Up or Weigh Him Down? Evan Osnos, The New Yorker

Biden's Big Lie About Charlottesville Will Backfire Jeffrey Lord, The American Spectator

A Long Talk With Cory Booker Gabriel DeBenedetti, New York Magazine

China, the U.S. and Trade in a Dog-Eat-Dog World Edward Luce, Financial Times

Obsession With 'Whiz Kids' Is Harming U.S. Teens Eric Spitznagel, New York Post

Just Don't Call Them UFOs Marina Koren, The Atlantic

RCP Morning Edition

Trump Will Relish His Fight With the Democratic House Rich Lowry, New York Post

Trump Should Be Indicted for Obstructing Russia Probe Rocah & Mariotti, Daily Beast

Collusion Conspiracy Theory Was an Attack on Democracy Aaron Mate, The Nation

Families Need Defending—Conservative Women Must Step Up Helen Andrews, NYT

Will Trump and Supreme Court Tear the Country Apart? E.J. Dionne, Washington Post

A Democratic Party Obsessed With Skin Color Karin McQuillan, American Greatness

Why Biden Will Be Tough to Beat in the Dem Primary Josh Kraushaar, National Journal

Democrats Would Be Mad to Nominate Biden in 2020 Mehdi Hasan, The Intercept

Mick Mulvaney on the White House, 2020 and Mueller Plott & Nicholas, The Atlantic

Want to Fix the Universities? Here Are 2 Options Roger Kimball, American Greatness

We're Standing Up to Gun Lobby and the NRA Is Losing Rep. Eric Swalwell, USA Today

Black Conservatives Debate Black Liberals on Trump, Democrats Lee Adams, Vice

Trump Made the Right Call on How to Win the 5G Race Mark Mills, New York Post

Au Revoir, Trump and Washington, D.C. Maureen Dowd, New York Times

Meet the Photographer Trump Can't Shut Up About Christopher Cadelago, Politico

vast.soptim.market... [Phuc Tran](#) [Get Addicted To](#) [Sam Quinones](#) [The Atlantic](#)



ENJOY YOUR AD AGE
SUBSCRIPTION BENEFITS TODAY,
PERFECTLY PACKAGED TO GET
YOU STARTED

[Subscribe now](#)

Favorability Ratings: U.S. Political Leaders

	Favorable	Unfavorable	Spread
Donald Trump	41.3	53.6	-12.3
Nancy Pelosi	36.5	50.3	-13.8
Mitch McConnell	23.5	46.3	-22.8
Chuck Schumer	28.0	39.7	-11.7

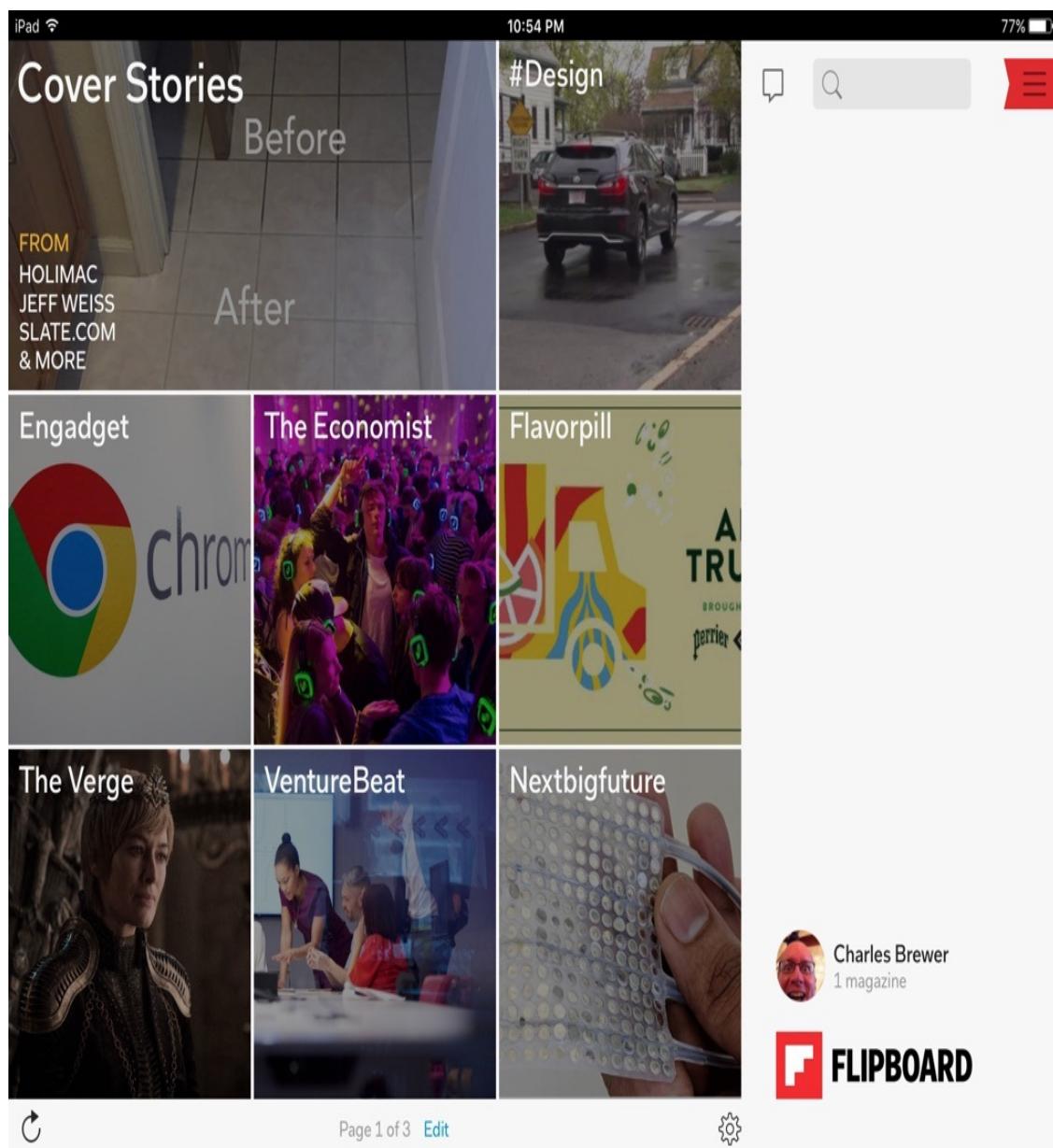
Looking for a career shift or to expand your skill set?
Learn the language of FM with the [Essentials of Facility](#)

80

2. Organizing the Content: Information Architecture and Application Structure

Figure 2-20. Real Clear Politics

The Real Clear family of sites is based around aggregating links to stories published elsewhere on the web. The most active is Real Clear Politics (Figure 2-20). While this looks like a simple, flat list that one might find in a wiki or encyclopedia, it is a feed. The links are in chronological publishing order, multiple times a day. Note how Real Clear Politics mimics the newspapers of the past, with a “morning edition” of curated links, followed by an “evening edition” of new links. The reader can scroll or use a menu selector to review previous days’ links. It’s an endless feed of a curated, time-based list of stories.



2. Organizing the Content: Information Architecture and Application Structure

Figure 2-21. Flipboard start screen

Flipboard looks like a magazine or picture viewer but is actually a feed reader. It can link to and aggregate from your social media accounts or from popular publishers with feeds. You can also use hashtag keywords to create feeds of matching articles from across all feeds.

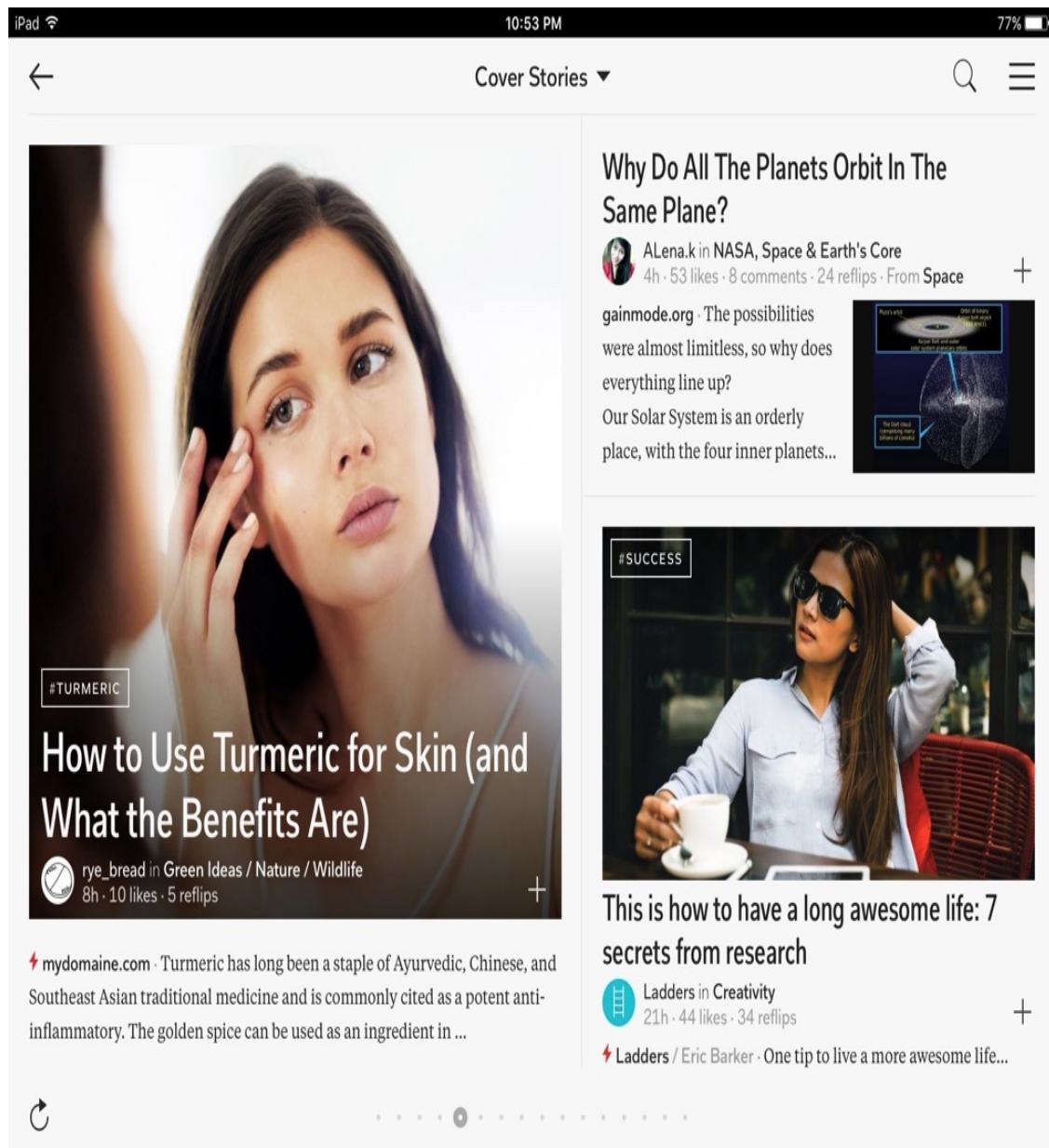


Figure 2-22. Flipboard news stream

This is actually a feed that has rendered the content cards in a variety of sizes, organized

2. Organizing the Content: Information Architecture and Application Structure into pages like a book. The user swipes left to right just like browsing a magazine.

Social Streams

The screenshot displays the Twitter mobile interface. At the top, there are navigation tabs: Home, Moments, Notifications, and Messages. The main header includes a search bar and a 'Tweet' button. Below the header, the user's profile picture and name (Charles Brewer) are shown, along with statistics: 5,947 Tweets, 1,122 Following, and 470 Followers.

Trends for you (Change):

- #LivePDMarathon: The 11-day Ultimate #LivePDMarathon continues TONIGHT at 9pm w/ PD Cam
- #BattleOfWinterfell: Preparing for the emotional massacre of the Battle of Winterfell
- #JackStopTheHate: Steve Portigal 🇺🇸 is Tweeting about this
- Richard Lugar: Former Indiana senator Richard Lugar dies at 87
- #NationalSuperheroDay: 18.9K Tweets
- Harden: NBA fans are calling for fouls on Klay Thompson's closeouts
- Rockets: NBA fans are calling for fouls on Klay Thompson's closeouts
- #MIGlobal: Tim Ferriss is Tweeting about this
- Rabbi Yisroel Goldstein: 5,900 Tweets
- Chris Paul: 11.5K Tweets

What's happening?

See 3 new Tweets

Charles Adler @cadler · 5h
OMG! 10 yrs ago today a small group of us launched @kickstarter ❤️❤️❤️
100 100 100 100 100

Been an epic decade. Excited for the next! 🎉🎉🎉🎉🎉

/cc @waxpancake @perrychen @ystrickler @cainlevy

Charles Adler @cadler · 5h
And thanks to @waxpancake for the reminder! Totally forgot. Lol.

The New York Times @nytimes · Mar 17
Here are some recent Netflix originals that are worth checking out

10 Recent Netflix Originals Worth Your Time
Whether you're looking for a new reality dating show, an irreverent comedy or a broody British thriller, here's our pick of the best recent titles.
nytimes.com

Sneak a peek at the new Twitter

Bookmarks, dark mode, data saver, and more – see all the new features coming to the web.

Take a look

Who to follow · Refresh · View all

Followed by Twitter TV 📺 and others

Amazon Prime Video US @primevideo · Follow
Promoted

Graeme Devine @zap... · Follow

Conrad Wadowski @conr... · Follow

Find people you know
Import your contacts from Gmail

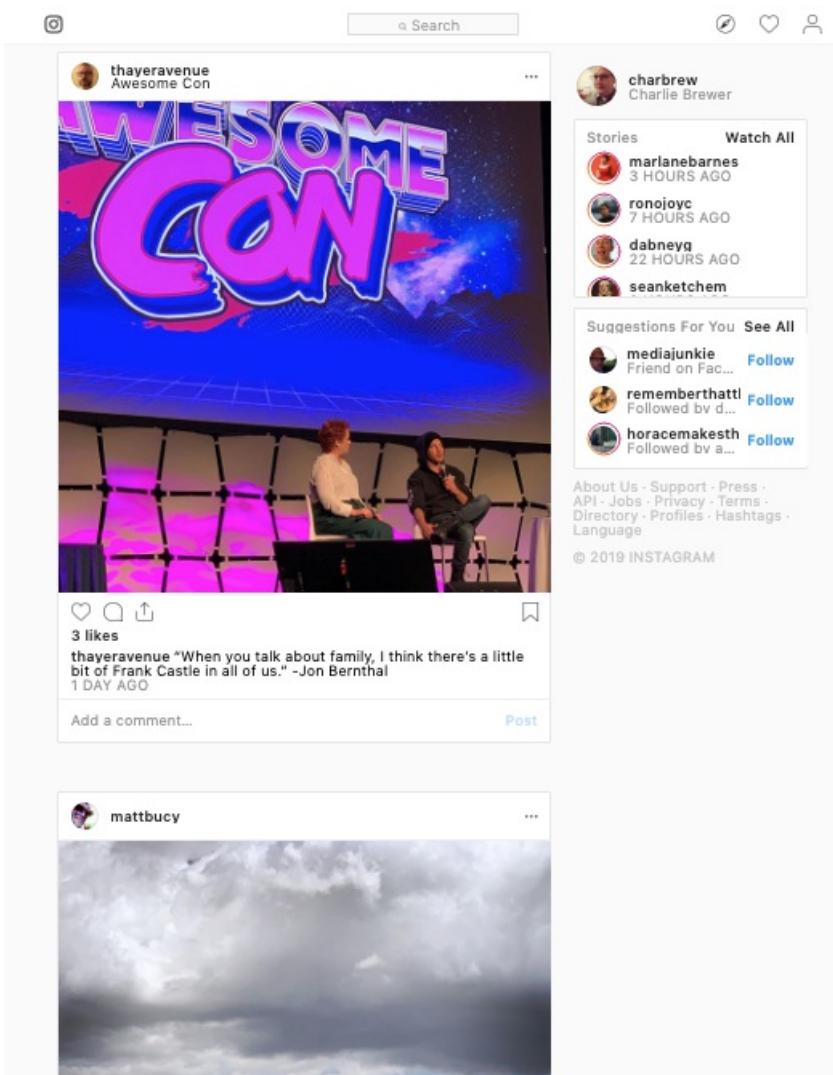
Connect other address books

© 2019 Twitter About Help Center Terms
Privacy policy Cookies Ads info Brand
Blog Status Apps Jobs Marketing
Businesses Developers

2. Organizing the Content: Information Architecture and Application Structure



Figure 2-23. Twitter



2. Organizing the Content: Information Architecture and Application Structure

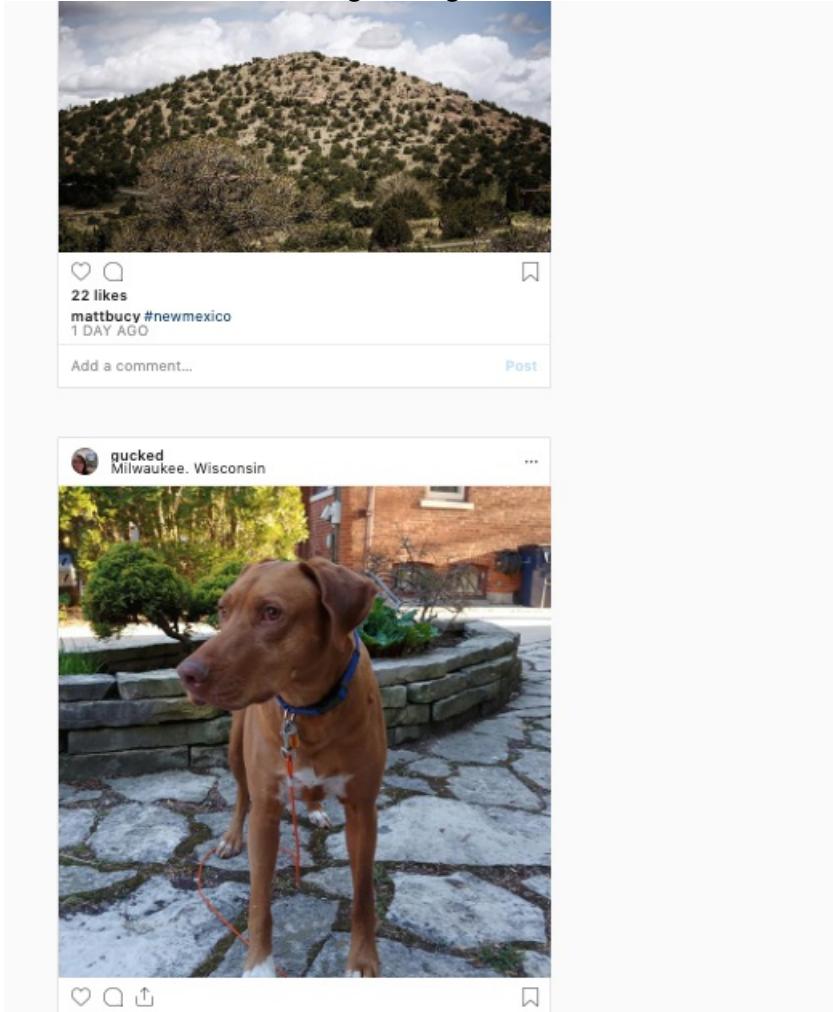


Figure 2-24. Instagram

Social streams dominate the consumer Internet experience. This doesn't show any signs of ending anytime soon. This is because streams drive engagement. As shown with Twitter ([Figure 2-23](#)) and Instagram ([Figure 2-24](#)), the social stream is alive and well. The feed can be a mix of content or posts or cards from business and personal networks, as with Twitter. Or it can be (almost) purely a personal social feed, as with Instagram. The feed is the consumption format here: View the image, the comment, and then use the social feedback features to like, share or comment in turn.

Social networking services, news aggregators, and private communications (such as email) provide plenty of examples of personal "News Stream".

Facebook automatically (and unpredictably) switches between a filtered "Top Stories" view, and a "Most Recent" view that shows everything. However, Facebook excels at

2. Organizing the Content: Information Architecture and Application Structure
the immediate response. Posting a short comment to a Facebook entry is almost as easy as thinking about it.

Business Collaboration

The screenshot shows a mobile application interface for business collaboration. On the left is a vertical sidebar with a dark purple header containing three colored dots (red, yellow, green) and a small profile icon. Below the header, the sidebar lists several channels with small icons and numbers indicating unread messages:

- #area-san-francisco (blue background, selected channel)
- #area-losangeles
- #enterprisesoftware
- #freshcontent
- #general
- #jobs
- #jtbd
- #media-pm

On the right is the main feed area. At the top, there's a search bar with a magnifying glass icon and the word "Search". To the right of the search bar are icons for a user profile, a star, and a more options menu. The feed displays two posts:

Thursday, April 25th

#area-san-francisco (by charbrew) 1

much. I was hoping i could meet some feedback/validation on the product and the strategy ahead. Any help would be greatly appreciated.

Thanks!

Friday, April 26th

#area-san-francisco (by charbrew) 1 3:25 PM

Hey guys! I'm looking to reach out to some senior level mobile developers in the Bay Area regarding a job opening in Redwood City. I'm working with a partner on a recruitment platform and I'd like some feedback on a job opening we have.

It would also be great to get a referral to fill that job! And, the position is offering a referral bonus, so reach out if you know a qualified mobile developer in/around Redwood City, CA!

2. Organizing the Content: Information Architecture and Application Structure

Saturday, April 27th

The screenshot shows a Slack workspace interface. On the left, there's a sidebar with various icons and channel names:

- # mtpcon
- # pm-and-design
- # pmmentoring
- # pmtraining
- # producttank
- # random
- # reading-list
- # saas

Below the channels, there's a list of Direct Messages:

- Slackbot
- charbrew (you)
- camassey
- housssem
- James
- saxenashobbit

At the bottom, there are buttons for "+ Invite people" and "Announcements".

A message from a user (@) at 5:28 PM reads: "You can try <https://www.facebook.com/groups/sanfranciscostartupjobs/>".

Next to the message is a link to a Facebook group page for "San Francisco Startup Jobs". The page includes a description: "San Francisco Startup Jobs Guidelines (MANDATORY READING): Looking to hire or be hired in San Francisco / Bay Area startup world? Jobs for developers, designers, product managers, growth, digital PR...". To the right of the description is a small image of the Golden Gate Bridge.

Below this, another message from "Albert's Job Listings & Referrals - San Francisco Bay Area" is shown. The message discusses A-List jobs, No MLM, No Exceptions, and California laws. It also mentions being SF Bay Area Focused. This message is accompanied by a small image of the Golden Gate Bridge.

Text below the messages states: "There are tons of active recruiters and job seekers on those groups who might be able to help".

At the bottom right, there's a message input field with a placeholder "Message #area-san-francisco" and a send button with an '@' icon and a smiley face.

2. Organizing the Content: Information Architecture and Application Structure

Figure 2-25. Slack

Social feeds and streams have made the jump from consumer experiences to business. They are they key component for enabling online, distributed and remote work collaboration. They allow employees to work together in a number of ways. They can start or add to threaded discussions organized by topic. Or they can congregate around a digital document that is generated collaboratively. This work can happen in real time or asynchronously. Employees can be in the same geographic location, or distributed across time zones. In Slack ([Figure 2-25](#)), the whole platform is built around discussion topics. Within the company “space,” employees can start or contribute to group discussions, or start private chat sessions with one or more coworkers. Files can be shared directly in the feed. In Quip a digital document is the anchor. Multiple collaborators work on this document. The social and comment feed next to the document gives a history of the discussion around the document. Both of these approaches are now a standard part of many business applications.

Grids

What

Use thumbnails, item views, and a browsing interface to create a “grid of objects” structure for browsing and selecting objects. This pattern is very common. For content-centric sites and applications, it allows an overview of the files, stories or documents available to read. Grids are also common for managing and editing photos, videos, and other pictorial items.

People use your software to work with lists or collections of pictorial things: photos, drawings, video clips, and so on. The list might be in a web page, or in an application, or both. It might allow editing by the owner of the content, or it might simply show the content to the public for browsing, viewing, and comments.

Why

This is a distinct style of application that is commonly used for mobile and desktop. It is also a guild of patterns—a set of patterns linked together and supporting each other in predictable ways. Once someone sees a “Thumbnail Grid” of images or videos in the right context, she knows what to expect: browse, click to view, set up slideshows or playlists, and so on.

2. Organizing the Content: Information Architecture and Application Structure
Patterns and other components that often play parts in this guild include:

- “Thumbnail Grid”
- “One-Window Drilldown”
- “Two-Panel Selector”
- “Pyramid”
- Tabs and “Collapsible Panels”
- “Button Groups”
- Trees or outlines
- “Keyboard Only”
- “Sharing Widget”
- Search box
- Social comments and discussion

How

Set up two principal views: a “Thumbnail Grid” of the items in the list, and a large view of a single item. Users will go back and forth between these. Design a browsing interface and associate it with the “Thumbnail Grid” to let users explore a large collection easily.

The Thumbnail Grid

Use this pattern to show a sequence of items. Many “Picture Manager” show a small amount of metadata with each item, such as its filename or author, but do this with care, as it clutters the interface. You might offer a control to adjust the size of the thumbnails. There may also be a way to sort the items by different criteria, such as date, label, or rating, or to filter it and show only the starred items (for instance).

When a user clicks on an item, show it immediately in the single-item view. Applications often let the user traverse the grid with the keyboard—for example, with the arrow keys

2. Organizing the Content: Information Architecture and Application Structure
and space bar. (See the “Keyboard Only” pattern in Chapter 1.)

If the user owns the items, offer ways to move, reorder, and delete items at this level in the interface. This implies having a multiple-selection interface, such as Shift-select, checkboxes, or lassoing a group of items with the pointer. Cut, copy, and paste should also work in applications.

You can offer slideshow or playlist functionality to all users at the “Thumbnail Grid” level.

The browsing interface

The contents of the “Thumbnail Grid” should be driven by a browsing interface that might be complex, simple, or nearly nonexistent, depending on the nature of the application. If needed, interfaces should offer a search box, either to search an individual user’s items or to search all public items (or both). Alternately, just present a scrollable grid.

Websites that host public collections, such as YouTube and Vimeo use the entire home page as a browsing interface. Sites such as these often present a balanced view with user-owned content and also public or promoted content.

Private photo and video management interfaces—especially desktop apps such as iPhoto or iMovie—should let the user browse the filesystem for images stored in different directories. If users can group items into albums, sets, projects, or other types of collections, these should be available in a browsing interface, too. Most also permit favoriting or starring of items.

Adobe Bridge puts filters into its browsing interface; more than 10 properties can be used to slice through a large collection of items, including keywords, modification date, camera type, and ISO.

The single-item view

Display a full story or document or image using the whole screen so the user can read it, edit it, or comment/share. This is the detail or full view screen as a screen on its own. Alternately, show a large view of the selected image (or a player, for a video). This view can be next to the “Thumbnail Grid” if the window is large, or it might replace the area used by the grid. Display metadata—information about the item—next to it. In practice,

2. Organizing the Content: Information Architecture and Application Structure
this means choosing between a “Two-Panel Selector” and a “One-Window Drilldown”.
See Chapter 5 for these list-related patterns.

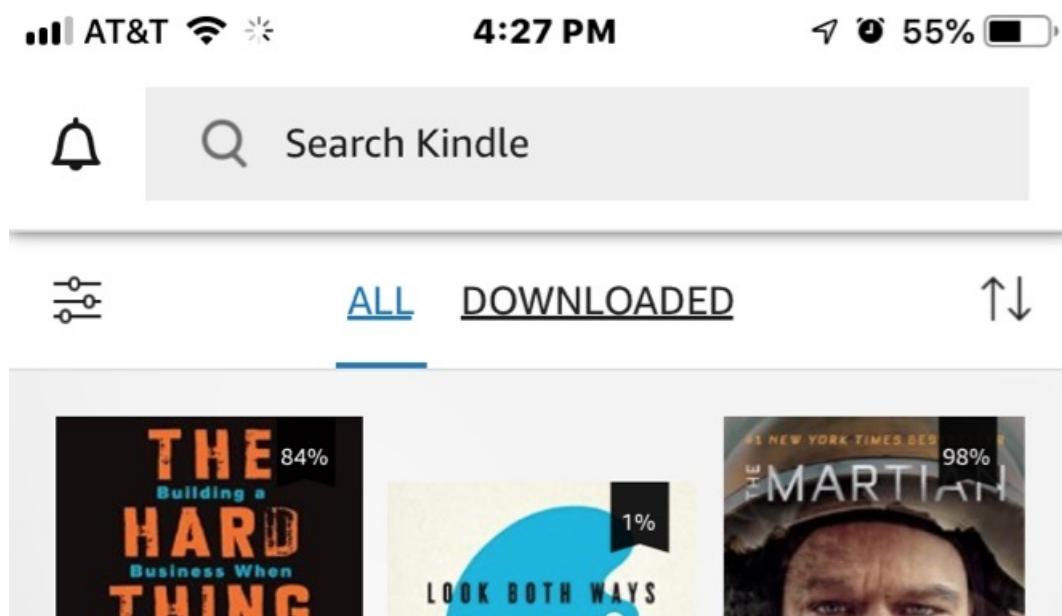
If the interface is a website or is otherwise web-connected, you might choose to offer social features at this level. Comments, liking or thumbs-up, and sharing might be here; see the “Sharing Widget” and other patterns in Chapter 9. Likewise, tagging or labeling can also be done here, either privately or publicly. An “other items you may like” feature is sometimes found in web-based public collections.

Editing features for individual items will live here, also. For instance, a photo manager might offer simple functionality such as cropping, color and brightness adjustment, and red-eye reduction. Metadata properties could be edited here, too. If a full editor is too complex to present here, give the user a way to launch a “real” editor. (Adobe Bridge, for example, lets the user launch Photoshop on a photo.) Use “Button Groups” to maintain a simple, comprehensible visual grouping of all these features.

Link the item to the previous and next items in the list by providing “previous” and “next” buttons, especially if you use “One-Window Drilldown” to display the single-item view (which also requires a “back” button). See the “Pyramid” navigational pattern in Chapter 3.

Examples

Browse a Collection of Objects



2. Organizing the Content: Information Architecture and Application Structure



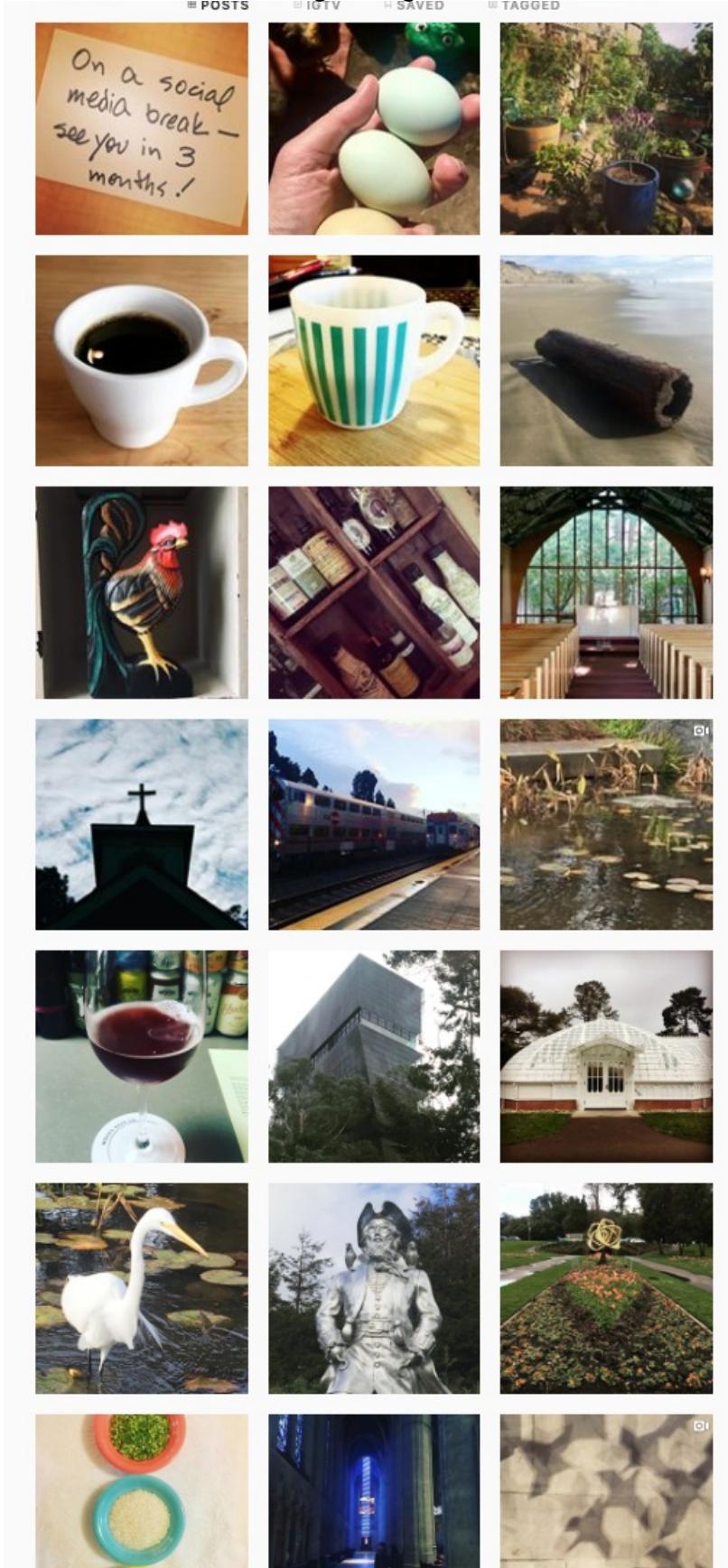
Figure 2-26. iOS Kindle App

charbrew [Edit Profile](#)

579 posts • 193 followers • 121 following

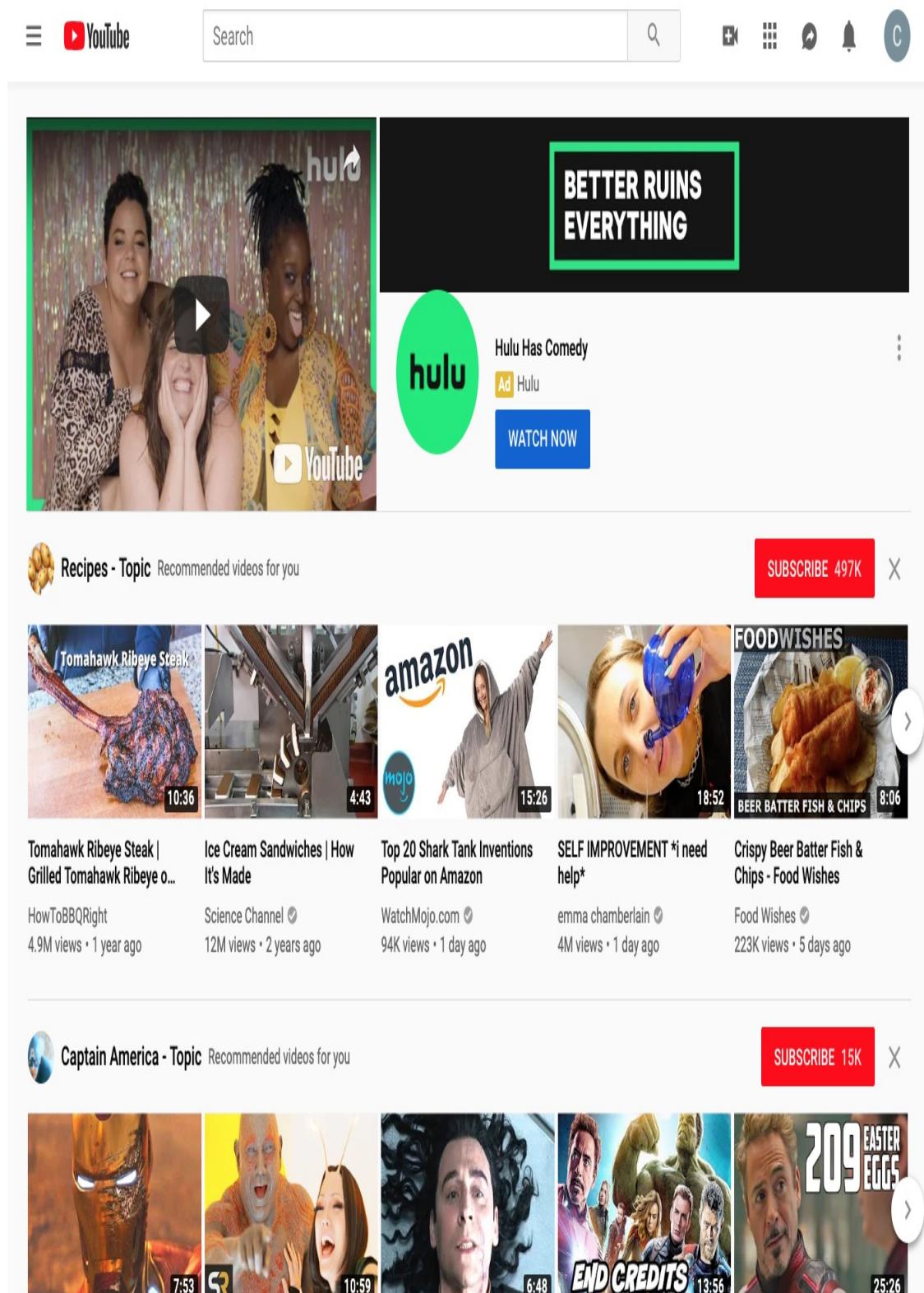
Charlie Brewer
Visible Innovation: UX & Product. Customer-focused Director of UX. Complicated products, design systems, leading teams. Photographer, foodie. LA & SF
www.linkedin.com/in/thecharbrew

2. Organizing the Content: Information Architecture and Application Structure



2. Organizing the Content: Information Architecture and Application Structure

Figure 2-27. Instagram Profile Screen



2. Organizing the Content: Information Architecture and Application Structure

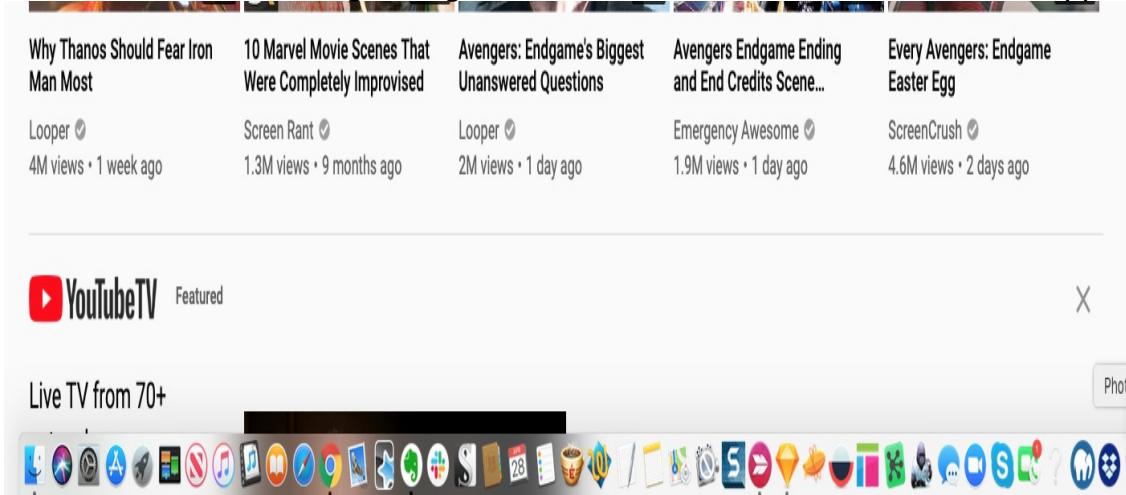


Figure 2-28. YouTube

2. Organizing the Content: Information Architecture and Application Structure

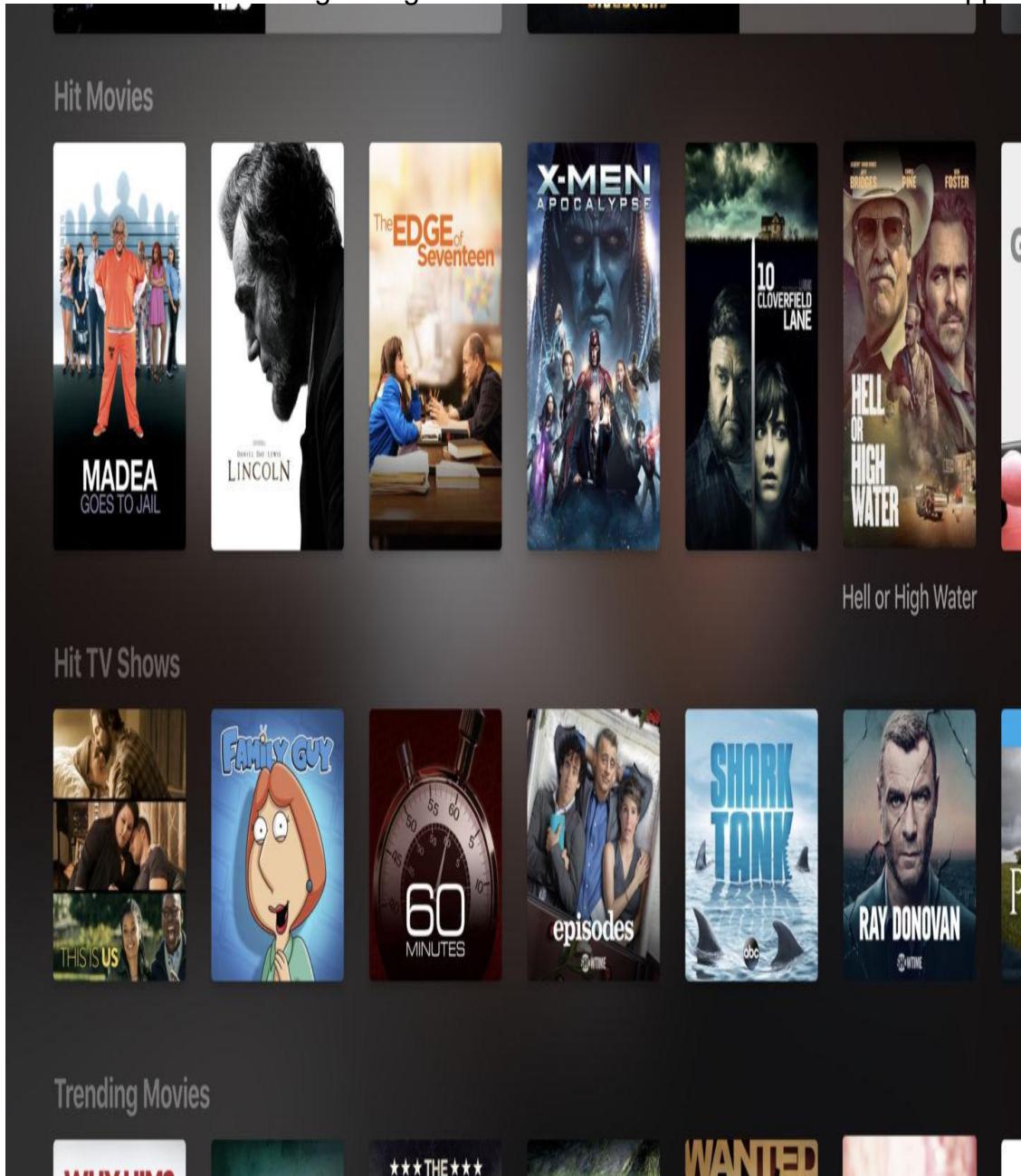


Figure 2-29. Apple TV



2. Organizing the Content: Information Architecture and Application Structure

Start free month
Need to train your team? Learn More.

IN PROGRESS
Having Difficult Conversations
1h 7m

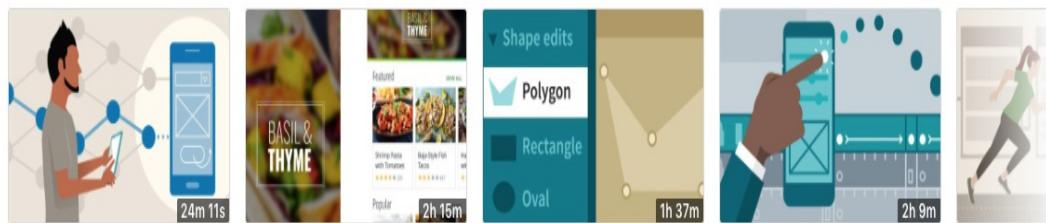
See all (20)

You have no saved content

SKILLS YOU FOLLOW
7

Stay sharp on Interaction Design

< Previous Next >



Designing for Neural Networks
and AI Interfaces

Sketch for UX Design

Sketch: Beyond the Basics

Adobe Animate: Designing
Interactive Experiences

UXPin: Design
Reviews

Top liked by your connections

< Previous Next >



Developing Executive Presence

Lead Like a Boss

Shane Snow on Storytelling

Transitioning from Manager to
Leader

InVision Essential
Design System

Trending on LinkedIn Learning

< Previous Next >



Becoming Indistractable

Leadership: Practical Skills

Leading with Vision

Motivating and Engaging
Employees

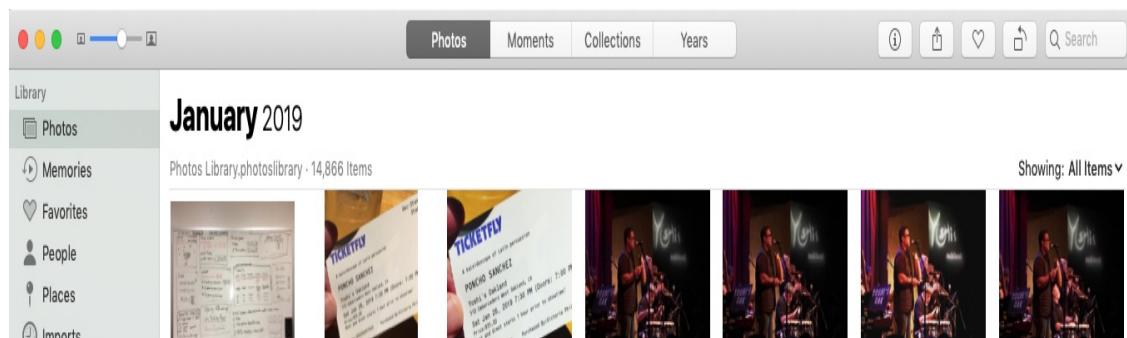
Transformation

Figure 2-30. LinkedIn Learning

The power of images is that they can carry a huge amount of information and can be recognized quickly. This is why they are so frequently used to represent a collection of objects for browsing and selecting. A grid of images presented with or without written descriptions is a compact, useful pattern for selecting a single item from a large collection. This pattern is universal across mobile, desktop and large screen UIs like Apple TV. Selecting a single item or card from the grid will load the object for direct consumption, or it will load a detail screen with a description. In [Figure 2-26](#), the Kindle reading app for iOS, the grid is nothing more than images of the book covers. In Instagram ([Figure 2-27](#)), one's profile has very little metadata in favor of a scrolling grid of previously posted images. The grid of images is your Instagram identity and personality. In the YouTube ([Figure 2-28](#)), Apple TV ([Figure 2-29](#)) and LinkedIn Learning ([Figure 2-30](#)), grids allow users to browse a large number of video assets quickly. Seen here also is a common variation on the square grid layout. It is a scrolling ribbon: a single height row that scrolls. Because of the huge number of items to browse, all of them group the images into categories for easier comprehension and review. Apple TV takes a minimalist approach. YouTube provides the most information, with each item becoming a “card” with image, title, author, and popularity metrics (not surprising since YouTube is built on social media dynamics).

Even video sites fit this pattern. When you view someone's YouTube channel, you can choose to see either a “Thumbnail Grid”, or in the single video view, a list beside a video player (the default). (Both options are shown in [Figure 2-14](#).) Clicking a thumbnail brings you to the page for that video, where detailed information and discussion are shown. Visitors can browse by looking at playlists, the latest videos added, the most-viewed videos, and the top-rated videos; a search box is also provided, as it is everywhere.

Manage and Edit Media Assets



2. Organizing the Content: Information Architecture and Application Structure

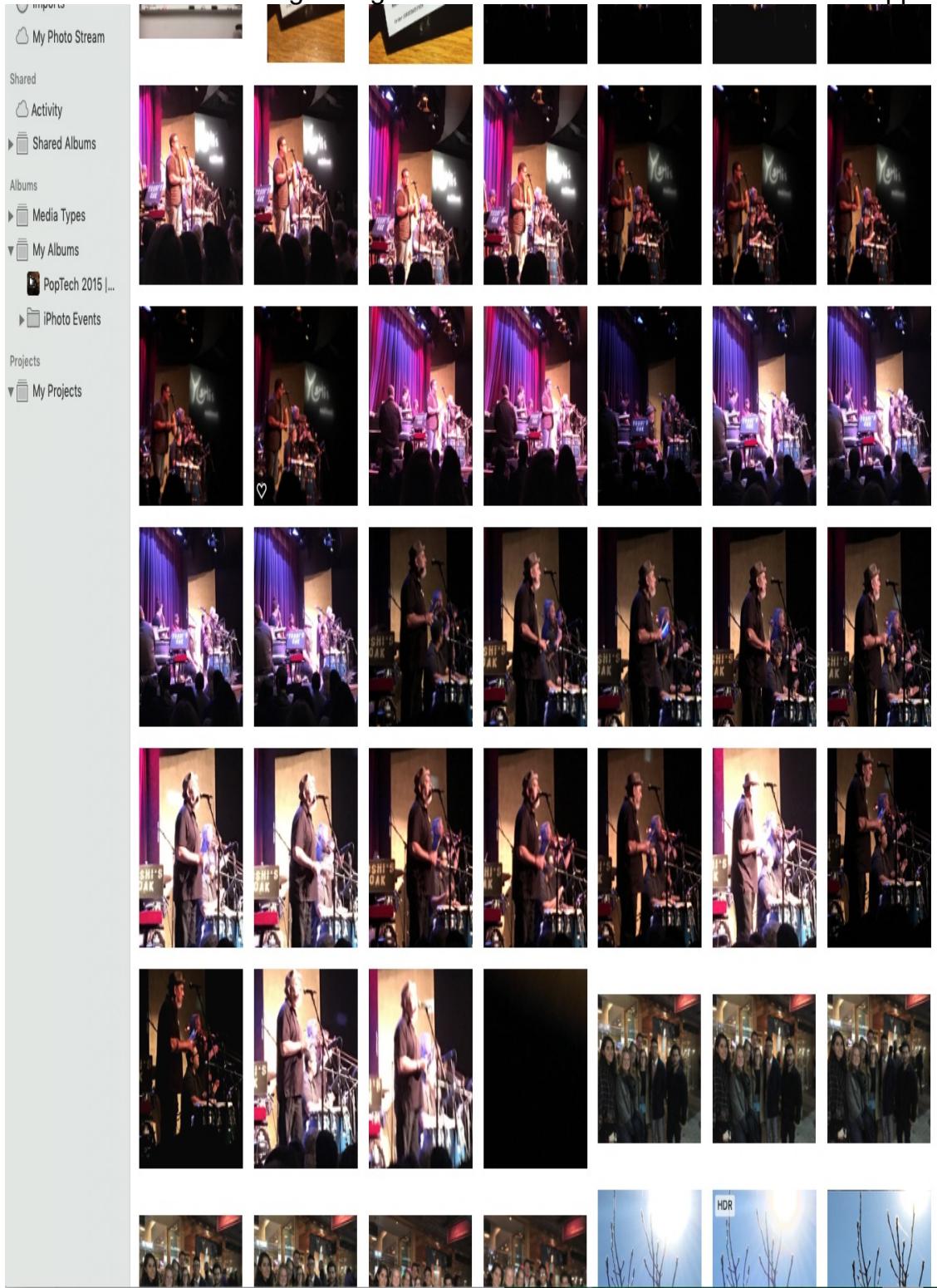


Figure 2-31. Apple Photos

2. Organizing the Content: Information Architecture and Application Structure

LA Dec 08 Folder - Adobe Bridge

Computer > Media & Projects > PICS_ARCHIVE > LA Dec 08 Folder

Sort by Filename

Preview Publish

Favorites Folders Content

LA Dec 08 Folder

Lightroom Lingba Louts NC_July_Sept_2004 New_apt_&_LA_2007 peeps Pencam Photo Booth Pics_Bos-NY-NC-UK-Pink-SF Polga scans PS source files Savanna_Jazz

Filter Collections

Keywords No Keywords (17)

Date Created 12/14/08 (5) 12/13/08 (1) 12/10/08 (11)

Date Modified 12/14/08 (5) 12/13/08 (1) 12/10/08 (11)

Camera Raw No Settings (17) Un-cropped (17)

PIC-0549.jpg PIC-0550.jpg PIC-0551.jpg PIC-0552.jpg

PIC-0553.jpg PIC-0554.jpg PIC-0555.jpg PIC-0556.jpg

PIC-0557.jpg

PIC-0558.jpg PIC-0559.jpg PIC-0560.jpg

PIC-0561.jpg PIC-0562.jpg

Metadata ISO--

1280 x 960 306 KB 131 ppi Untagged RGB

File Properties

Filename	PI
Document Type	J
Date Created	12
Date File Modified	12
File Size	3
Dimensions	12
Dimensions (in inches)	9.
Resolution	13
Bit Depth	8
Color Mode	R
Color Profile	U
> IPTC Core	
> IPTC Extension	

17 items, 1 selected - 306 KB

2. Organizing the Content: Information Architecture and Application Structure

Figure 2-32. Adobe Bridge

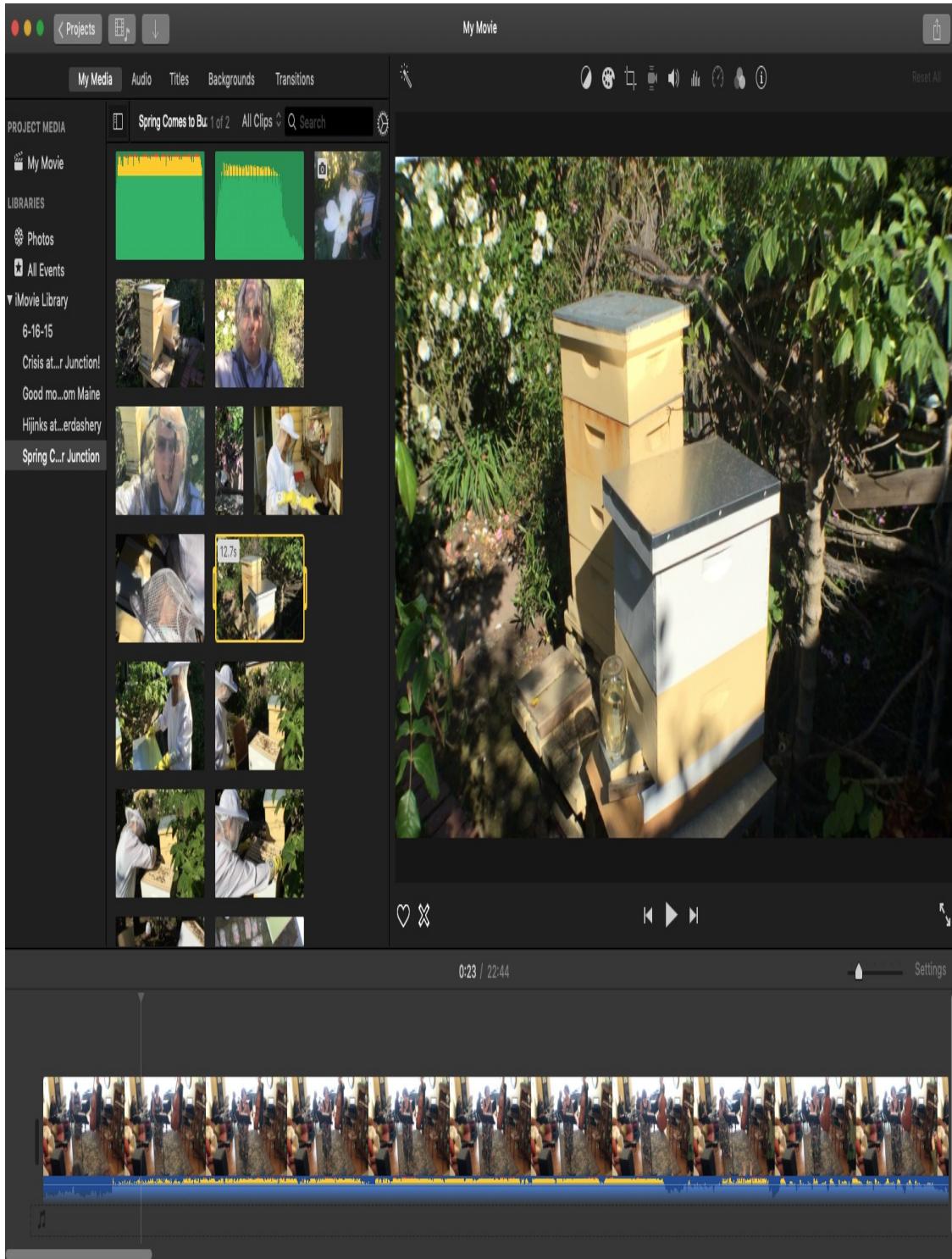


Figure 2-33. Apple iMovie



2. Organizing the Content: Information Architecture and Application Structure

Home Tools All.pdf x

Organize Pages 1 Close

Share

1 2 3 4 5 6 7

8 9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

Proposal:
Lease Notifications
in the App

Functional
Specifications:
Tables

Functional
Specifications:
Costs, Projections, Reports, Integrations,
Leases, Employees, Tickets

2. Organizing the Content: Information Architecture and Application Structure

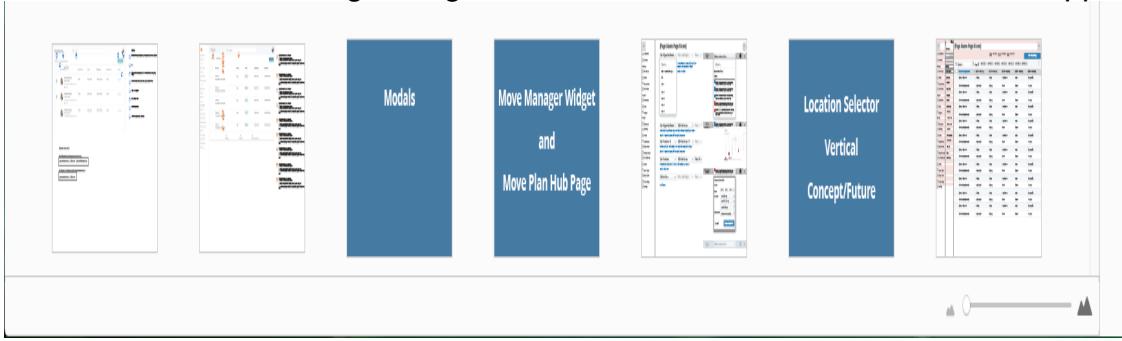


Figure 2-34. Adobe Acrobat

Media and document creators also use this grid layout to manage assets that are being assembled or processed. Apple iPhoto (shown in [Figure 2-31](#)), Adobe Bridge ([Figure 2-32](#)) and Apple iMovie ([Figure 2-33](#)) are mobile desktop applications for managing personal collections of images. Their browsing interfaces—all “Two-Panel Selector”—vary in complexity from iPhoto’s very simple design to Adobe Bridge’s numerous panels and filters. iPhoto uses “One-Window Drilldown” to reach the single-item view, while Adobe Bridge ([Figure 2-32](#)) and Apple iMovie ([Figure 2-33](#)) put all three views together on one page. A common variation on the square grid layout is a single height row that scrolls: a ribbon. In the case of iMovie, it is a timeline. It is the central working palette for creating time-based media such as videos.

Adobe Acrobat ([Figure 2-34](#)), the reader/editor for the popular PDF document format, offers a grid view of document pages. This is used in the edit mode. It allows for rapid reorganization of pages, or selecting pages for deletion, or selecting an insertion point for adding screens.

In other libraries

The Image Browser pattern at Welie.com describes some aspects of a Picture Manager:

<http://welie.com/patterns/showPattern.php?patternID=image-browsing>

Dashboard

What

Arrange data displays into a single (sometimes multiple) information-dense page, updated

2. Organizing the Content: Information Architecture and Application Structure regularly. Show users relevant, actionable information, and let them customize the display as necessary. A dashboard is often the first screen a customer will see when logging in to a consumer or business platform. They are a very common pattern for business information software.

Use when

Dashboards are very popular because they solve the need to get a quick update on status, on key information, and on tasks to be done. Your site or application deals with an incoming flow of information from something—web server data, social chatter, news, airline flights, business intelligence information, or financials, for example. Your users would benefit from continuous monitoring of that information.

Why

This is a familiar and recognizable page style. Dashboards have a long history, both online and in the physical world, and people have well-established expectations about how they work: they show useful information, they update themselves, they usually use graphics to display data, and so on.

A dashboard is also a guild of interlocking patterns and components. Many online dashboards use these in predictable ways:

- “Titled Sections”
- Tabs and “Collapsible Panels”
- “Movable Panels”
- “One-Window Drilldown”
- Lists and tables of various kinds (see Chapter 5)
- “Row Striping”
- Information graphics (see Chapter 7)
- “Datatips”

How

Determine what information users need or want to see, or what tasks they need to stay on top of. This isn't as simple as it sounds, because you need a researcher's or editor's eye—eliminate confusing or unimportant data, or people won't be able to pick out the parts that matter. Remove, or at least deemphasize, information that doesn't help the user. Feature the most important information or next steps.

Use a good visual hierarchy (see Chapter 4) to arrange lists, tables, and information graphics on the page. Try to keep the main information on one page, with little or no scrolling, so people can keep the window on-screen and see everything at a glance. Group related data into “Titled Sections”, and use tabs only when you're confident that users won't need to see the tab contents side by side.

Use “One-Window Drilldown” to let users see additional details about the data—they should be able to click on links or graphics to find out more. “Datatips” work well to show individual data points when the pointer rolls over an information graphic.

Choose appropriate and well-designed information graphics for the data you need to show. Gauges, dials, pie charts, and 3D bar charts look nice, but they are rarely the best way to show comparative information at a glance—simple line and bar charts express data better, especially time-based data. When numbers and text are more relevant than graphics, use lists and tables. “Row Striping” is a common pattern for multicolumn data tables.

People will try to get actionable information from the dashboard at a glance, without looking hard at every element on the page. So, when you show text, consider highlighting keywords and numbers so that they stand out from surrounding text.

Should your users be able to customize their dashboard displays? Many dashboards do offer customization, and your users may expect it. One way to customize a dashboard page is to rearrange the sections—Salesforce offers “Movable Panels” to users, in addition to choosing which gadgets get shown.

Examples

Salesforce ([Figure 2-35](#)) has built a huge software business around answering the need for enterprises large and small to monitor and manage all forms of business processes. Custom-built and customizable dashboards are a central part of this strategy. Some

2. Organizing the Content: Information Architecture and Application Structure examples of purpose-built dashboards are shown here. Users can build or configure modules according to their needs, and then arrange the standard-sized modules into a grid that suits them best. These can be saved and shared.

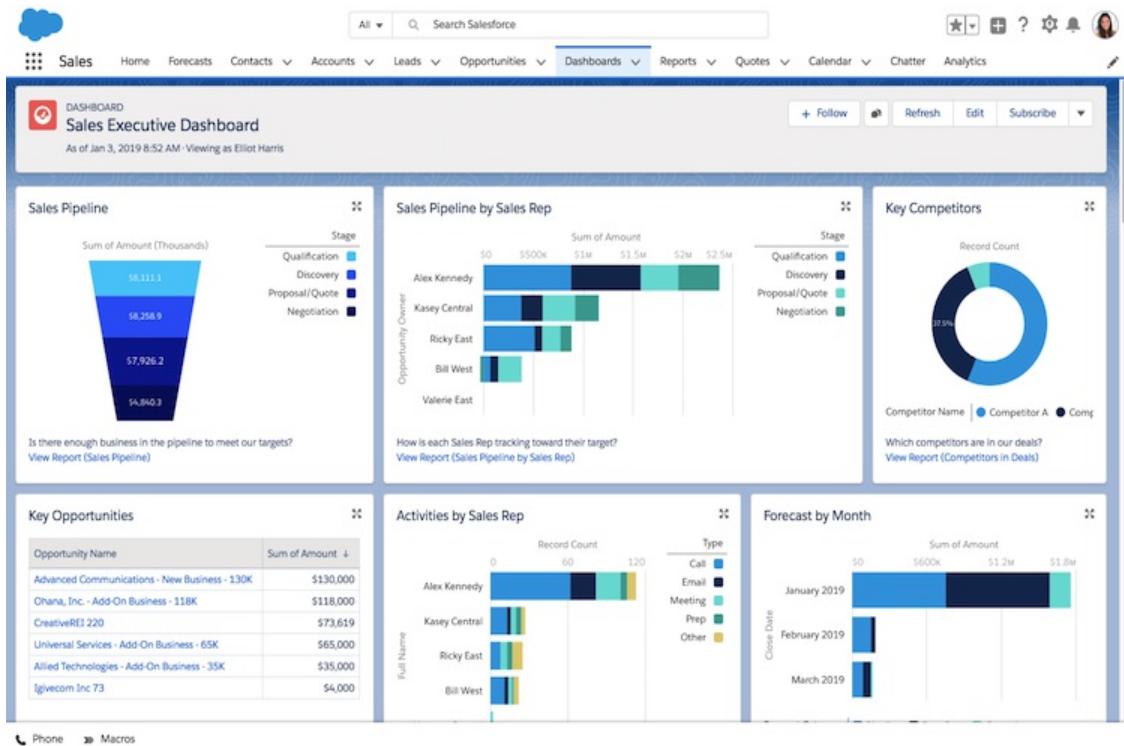


Figure 2-35. Salesforce Dashboard, one example

2. Organizing the Content: Information Architecture and Application Structure

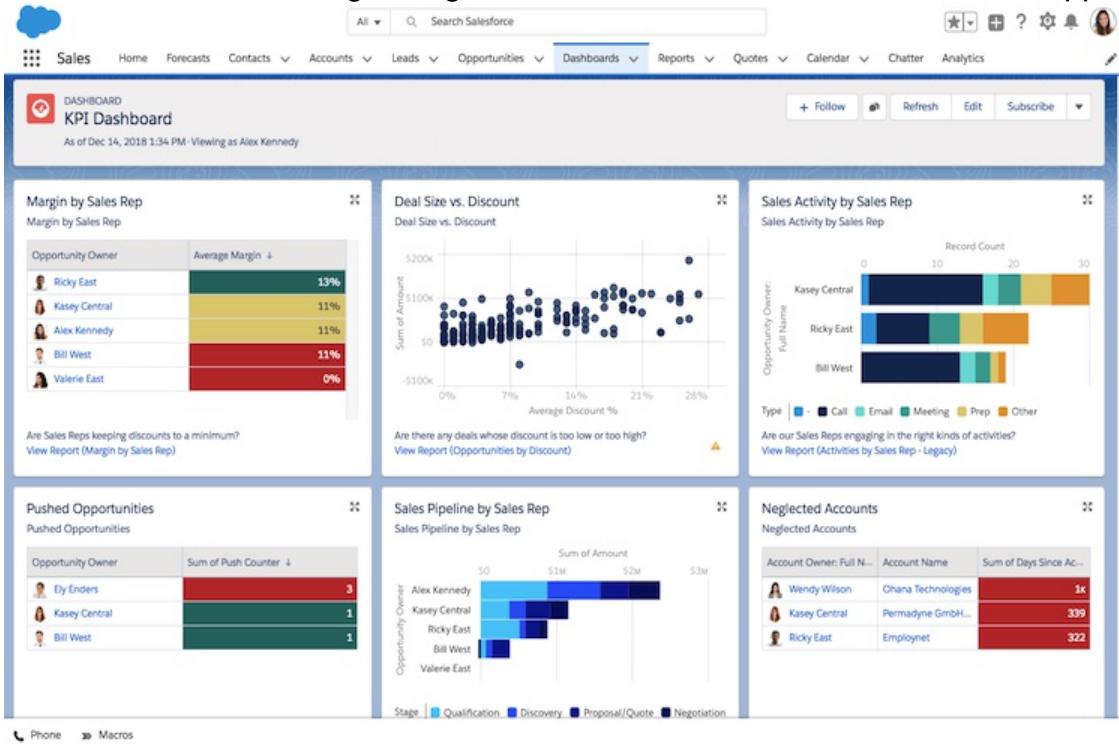
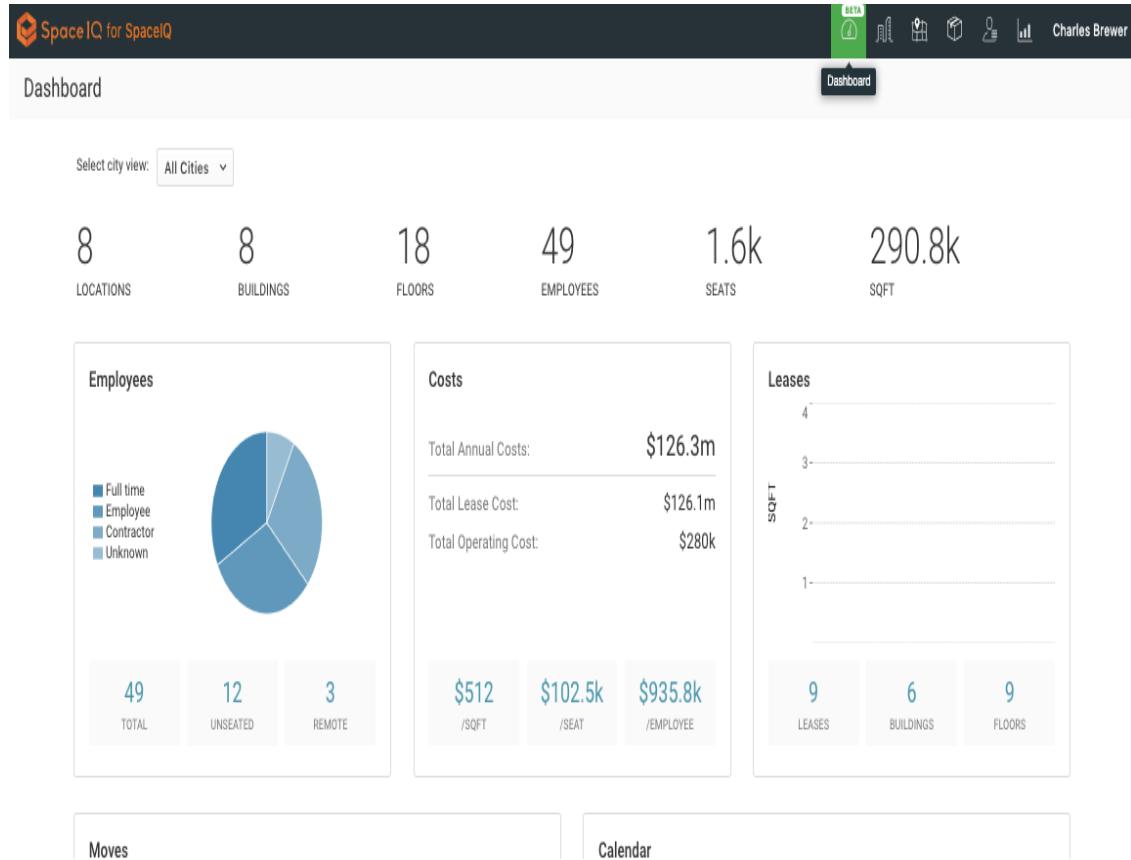


Figure 2-36. Salesforce Dashboard, another example



2. Organizing the Content: Information Architecture and Application Structure

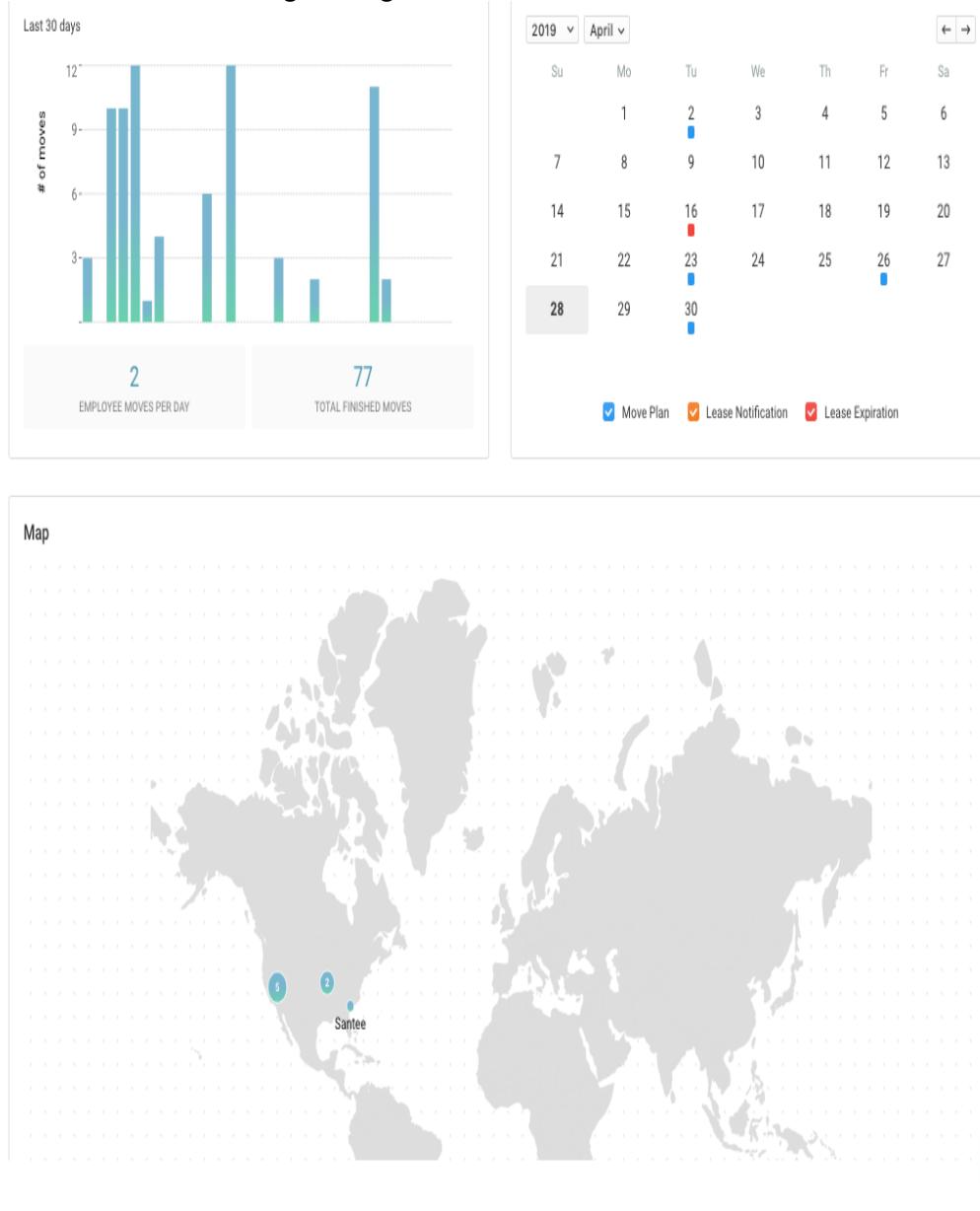


Figure 2-37. SpaceIQ

SpaceIQ ([Figure 2-37](#)) offers a dashboard on login. This startup follows a classic design pattern of offering key performance indicators, an overview of what is happening now or needs attention, and quick navigation to other key parts of the platform.

In other libraries

- <http://quince.infragistics.com/Patterns/Dashboard.aspx>

2. Organizing the Content: Information Architecture and Application Structure

- <http://patternry.com/p=information-dashboard/>
- Dashboard is one of the canonical RIA screen layouts described by Bill Scott and Theresa Neil. An article in UX Magazine explains these layouts:
- <http://www.uxmag.com/design/rich-internet-application-screen-design>
- Finally, you may be interested in Stephen Few's book on information dashboards.
- Few, Stephen. *Information Dashboard Design: Displaying Data for at-a-Glance Monitoring*. Analytics Press, 2013.

Canvas Plus Palette

What

Place an iconic palette next to a blank canvas; the user clicks on the palette buttons to create objects on the canvas.

Use when

You're designing any kind of graphical editor. A typical use case involves creating new objects and arranging them on some virtual space.

Why

This pair of panels—a palette with which to create things, and a canvas on which to put them—is so common that almost every user of desktop software has seen it. It's a natural mapping from familiar physical objects to the virtual on-screen world. And the palette takes advantage of visual recognition: the most common icons (paintbrush, hand, magnifying glass, etc.) are reused over and over again in different applications, with the same meaning each time.

How

Present a large empty area to the user as a canvas. It might be in its own window, as in Photoshop, or embedded in a single page with other tools. The user just needs to see the canvas side by side with the palette. Place additional tools—property panels, color

2. Organizing the Content: Information Architecture and Application Structure swatches, and so on—to the right or bottom of the canvas, in small palette-like windows or panels.

The palette itself should be a grid of iconic buttons. They can have text in them if the icons are too cryptic; some GUI-builder palettes list the names of GUI components alongside their icons, for instance. So does Visio, with its palettes of complex visual constructs tailored for specific domains. But the presence of icons is necessary for users to recognize the palette for what it is.

Place the palette to the left or top of the canvas. It can be divided into subgroups, and you may want to use “Module Tabs” or “Collapsible Panels” to present those subgroups.

Most palette buttons should create the pictured object on the canvas. But many builders have successfully integrated other things, such as zoom mode and lassoing, into the palette.

The gestures used to create items on a palette vary from one application to another. Some use drag-and-drop only; some use a single click on the palette and a single click on the canvas; and some use One-off Modes, Spring-Loaded Modes (see the previous edition of this book for both of these patterns), and other carefully designed gestures. I have always found that usability testing in this area is particularly important, since users’ expectations vary greatly.

Examples

2. Organizing the Content: Information Architecture and Application Structure

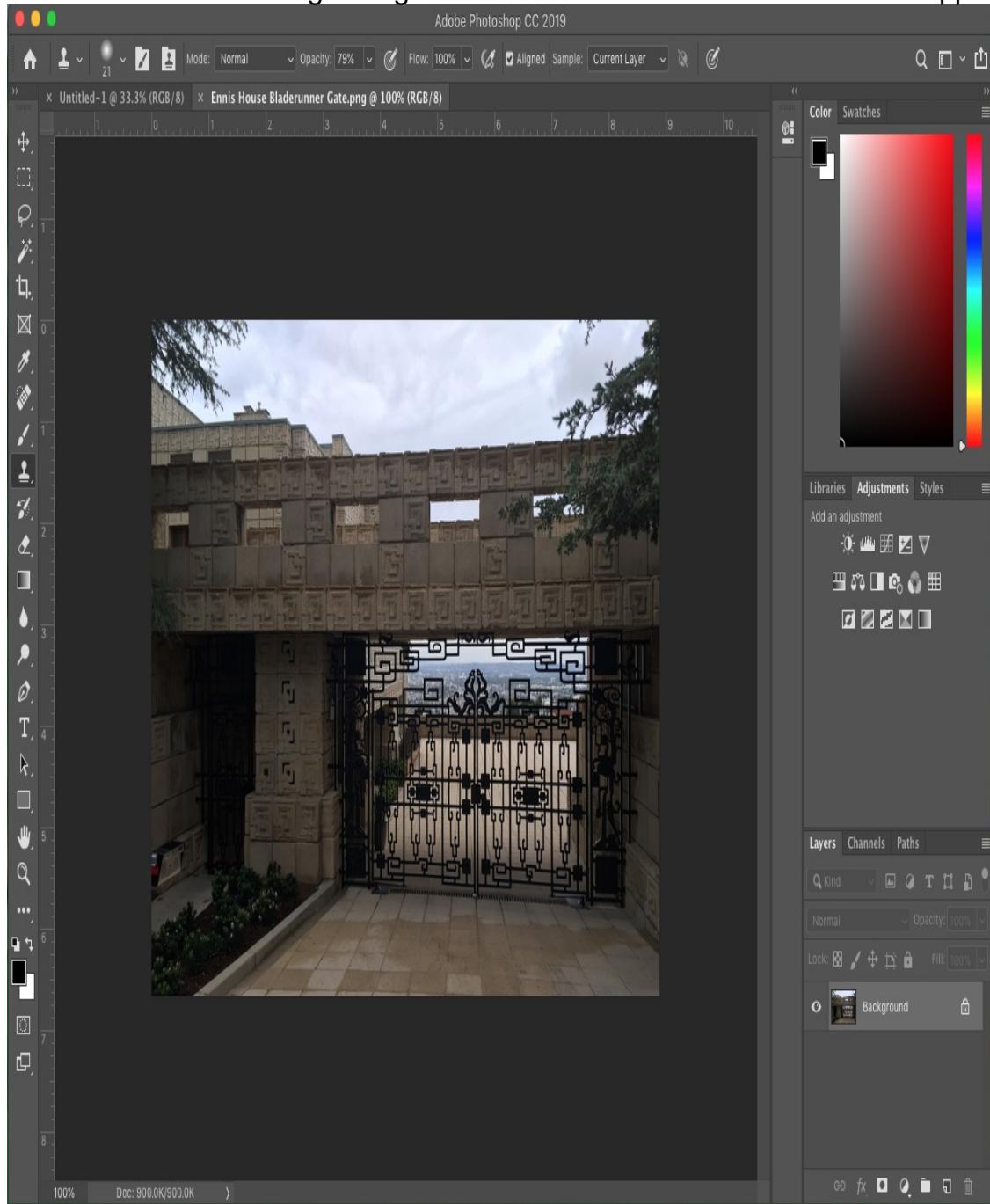


Figure 2-38. Adobe Photoshop

2. Organizing the Content: Information Architecture and Application Structure

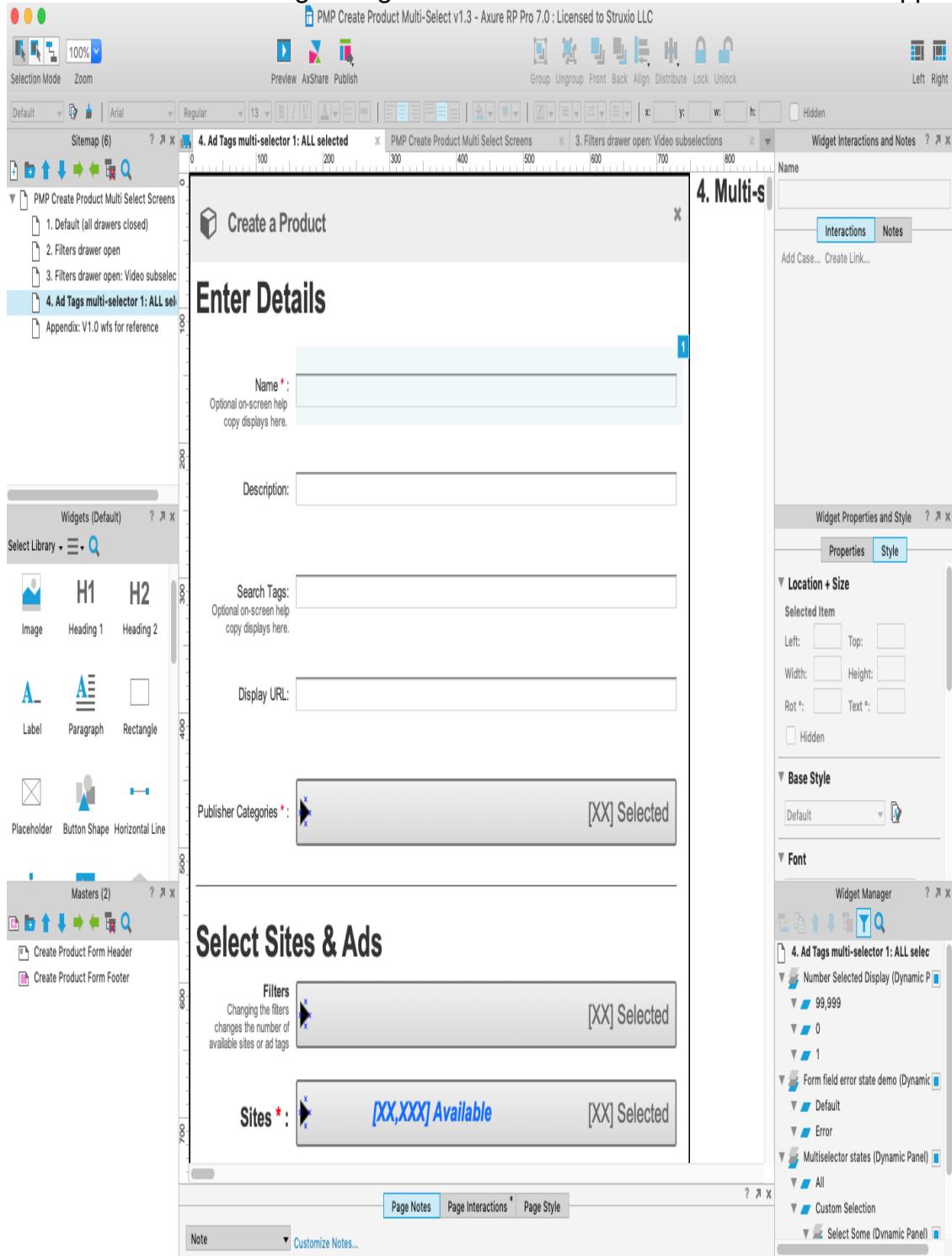


Figure 2-39. Axure RP Pro

2. Organizing the Content: Information Architecture and Application Structure

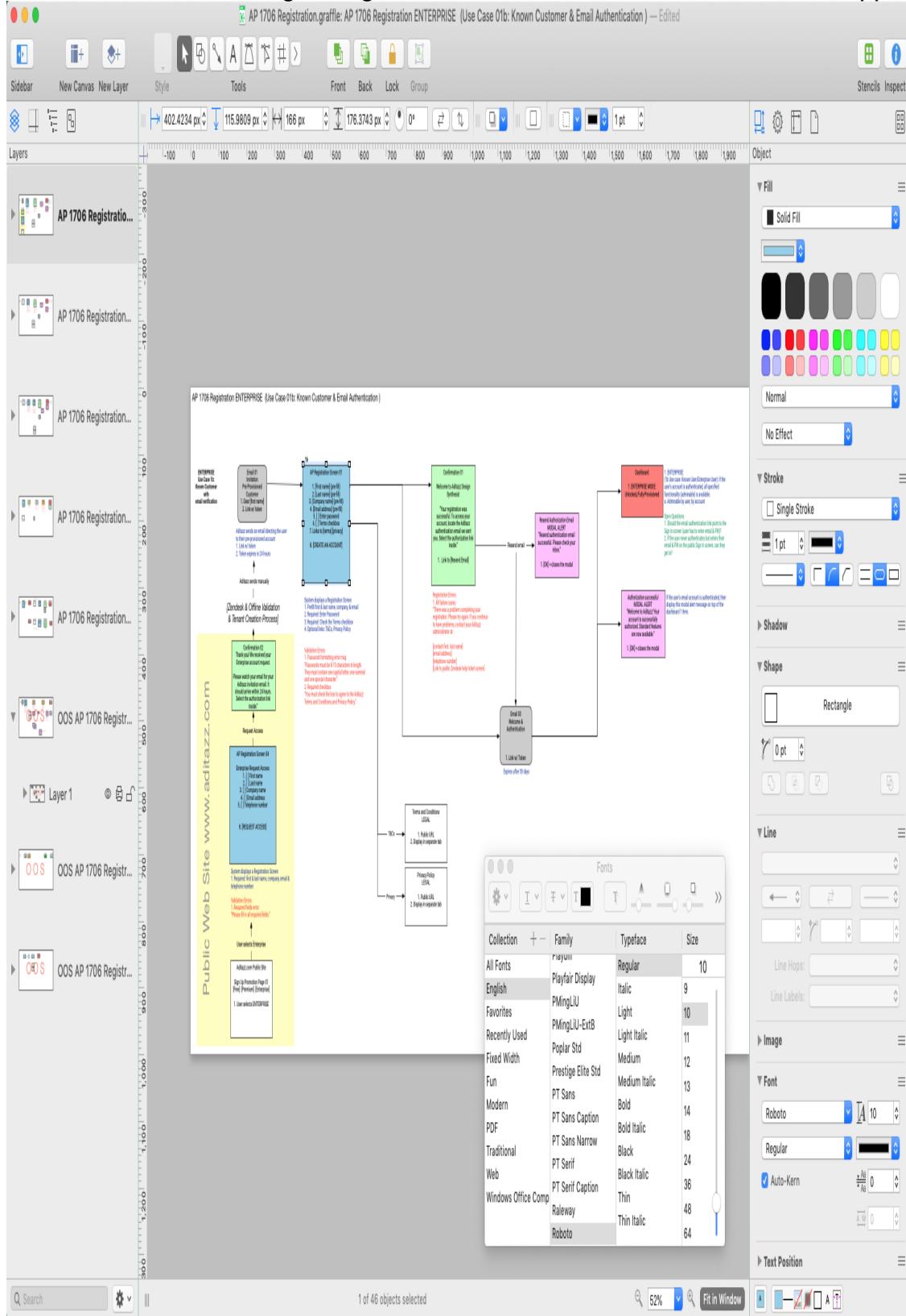


Figure 2-40. OmniGraffle

2. Organizing the Content: Information Architecture and Application Structure



 **Light**



 **Color**



 **B&W**





Figure 2-41. iOS Photos

In other libraries

Palette/Canvas is one of the canonical RIA screen layouts described by Bill Scott and Theresa Neil. [An article in UX Magazine](#) explains these layouts.

Wizard

What

Lead the user through the interface step by step to do tasks in a prescribed order.

Use when

You are designing a UI for a task that is long or complicated, and that will usually be novel for users—not something that they do often or want much fine-grained control over (such as the installation of a software package). You’re reasonably certain that the designer of the UI will know more than the user does about how best to get the task done.

Tasks that seem well suited for this approach tend to be either branched or very long and tedious—they consist of a series of user-made decisions that affect downstream choices.

The catch is that the user must be willing to surrender control over what happens when. In many contexts, that works out fine, since making decisions is an unwelcome burden for people doing certain things: “Don’t make me think, just tell me what to do next.” Think about moving through an unfamiliar airport—it’s often easier to follow a series of signs than it is to figure out the airport’s overall structure. You don’t get to learn much about how the airport is designed, but you don’t care about that.

But in other contexts, it backfires. Expert users often find “Wizard” frustratingly rigid and limiting. This is particularly true for software that supports creative processes such as writing, art, or coding. It’s also true for users who actually do want to learn the software;

2. Organizing the Content: Information Architecture and Application Structure
“Wizard” don’t show users what their actions really do, or what application state gets changed as choices are made. That can be infuriating to some people. Know your users well!

Why

Divide and conquer. By splitting up the task into a sequence of chunks, each of which can be dealt with in a discrete “mental space” by the user, you effectively simplify the task. You have put together a preplanned road map through the task, thus sparing the user the effort of figuring out the task’s structure—all he needs to do is address each step in turn, trusting that if he follows the instructions, things will turn out OK.

But the very need for a “Wizard” indicates that a task may be too complicated. If you can simplify a task to the point where a short form or a few button clicks can do the trick instead, that’s a better solution. (Keep in mind, too, that “Wizard” are considered a bit patronizing in some Asian cultures.)

How

“Chunking” the task

Break up the operations constituting the task into a series of chunks, or groups of operations. You may need to present these groups in a strict sequence, or not; sometimes there is value in breaking up a task into steps 1, 2, 3, and 4 just for convenience.

A thematic breakdown for an online purchase may include screens for product selection, payment information, a billing address, and a shipping address. The presentation order doesn’t much matter because later choices don’t depend on earlier choices. Putting related choices together just simplifies things for people filling out those forms.

You may decide to split up the task at decision points so that choices made by the user can change the downstream steps dynamically. In a software installation “Wizard”, for example, the user may choose to install optional packages that require yet more choices; if she chooses not to do a custom installation, those steps are skipped. Dynamic UIs are good at presenting branched tasks such as this, because the user never has to see anything that’s irrelevant to the choices she made.

In either case, the hard part of designing this kind of UI is striking a balance between the

2. Organizing the Content: Information Architecture and Application Structure sizes of the chunks and the number of them. It's silly to have a 2-step "Wizard", and a 15-step "Wizard" is tedious. On the other hand, each chunk shouldn't be overwhelmingly large, or you've lost some benefits of this pattern.

Physical structure

"Wizard" that present each step in a separate page, usually navigated with Back and Next buttons, are the most obvious and well-known implementation of this pattern. They're not always the right choice, though, because now each step is an isolated UI space that shows no context—the user can't see what went before or what comes next. But an advantage of such "Wizard" is that they can devote each page to that step completely, including illustrations and explanations.

If you do this, allow the user to move back and forth at will through the task sequence. Offer a way for the user to step backward, or to otherwise change her mind about an earlier choice. Additionally, many UIs show a selectable map or overview of all the steps, getting some of the benefits of a "Two-Panel Selector". (In contrast to that pattern, a "Wizard" implies a prescribed order—even if it's merely suggested—as opposed to completely random access.)

If you instead choose to keep all the steps on one page, you could use one of several patterns from Chapter 4:

- "Titled Sections", with prominent numbers in the titles. This is most useful for tasks that aren't heavily branched, since all steps can be visible at once.
- "Responsive Enabling", in which all the steps are present on the page, but each one remains disabled until the user has finished the previous step.
- "Responsive Disclosure", in which you wait to show a step on the UI until the user finishes the previous one. Personally, I think this is the most elegant way to implement a short "Wizard". It's dynamic, compact, and easy to use.
- "Good Defaults" (from Chapter 8) are useful no matter how you arrange the steps. If the user is willing to turn over control of the process to you, odds are good she's also willing to let you pick reasonable defaults for choices she may not care much about, such as the location of a software installation.

Examples

2. Organizing the Content: Information Architecture and Application Structure

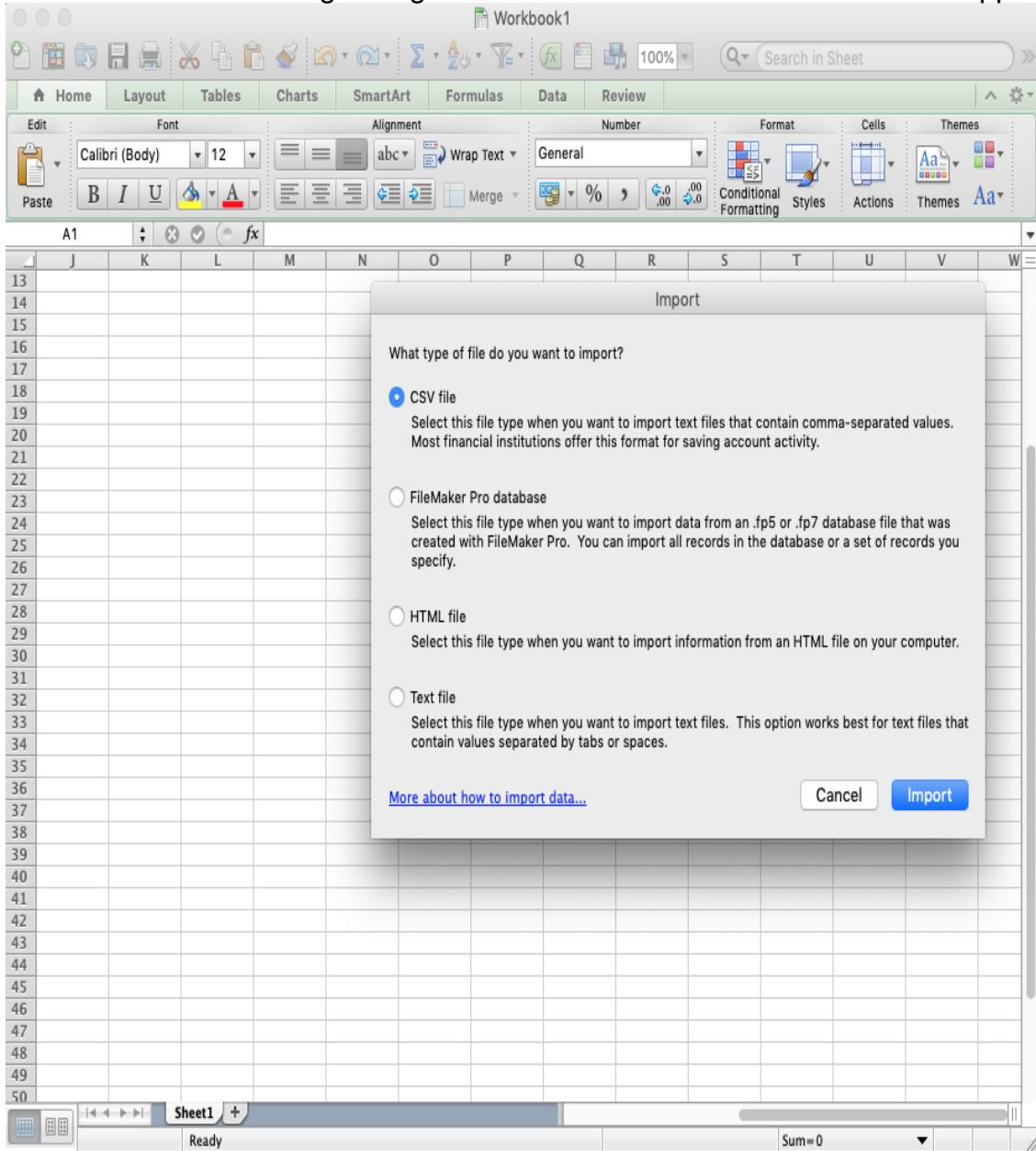


Figure 2-42. Microsoft Excel import wizard

The Microsoft Office designers have done away with many of its “Wizard”, but a few remain—and for good reason. Importing data into Excel is a potentially bewildering task. The Import Wizard (see [Figure 2-42](#)) is an old-school, traditional application “Wizard” with that guides the user step by step through the import process. It uses Back/Next buttons, branching, and no sequence map, but it works. Each screen lets you focus on the step at hand, without worrying about what comes next.

In other libraries

- <http://ui-patterns.com/patterns/Wizard>
- <http://www.welie.com/patterns/showPattern.php?patternID=wizard>
- <http://patternry.com/p=one-page-wizard/>
- <http://patternry.com/p=multiple-page-wizard/>
- <http://quince.infragistics.com/Patterns/Wizard.aspx>

Wizard is one of the canonical RIA screen layouts described by Bill Scott and Theresa Neil. [An article in UX Magazine](#) explains these layouts.

Settings Editor

What

Provide an easy-to-find, self-contained page or window where users can change settings, preferences, or properties. Divide the content into separate tabs or pages, if you need to manage large numbers of settings.

Use when

You are designing any of the following applications or tools, or something similar:

- An application that has app-wide preferences.
- An operating system, mobile device, or platform that has system-wide preferences.
- A site or app for which a user must sign in—users will need to edit their accounts and profiles.
- An open-ended tool to create documents or other complex work products. Users may need to change a document's properties, an object within a document, or another item.
- A product configurator, which allows people to customize a product online. (This is

2. Organizing the Content: Information Architecture and Application Structure
really a different pattern, however, with slightly different requirements and
constraints. See the Product Configurator pattern at
[http://www.welie.com/patterns/showPattern.php?patternID=product-configurator. \)](http://www.welie.com/patterns/showPattern.php?patternID=product-configurator.)

Why

Though both use forms, a “Settings Editor” is distinct from a “Wizard”, and it has very particular requirements. A user must be able to find and edit a desired property without being forced to walk through a prescribed sequence of steps—random access is important.

To aid findability, the properties should be grouped into categories that are well labeled and make immediate sense.

Another important aspect of “Settings Editor” design is that people will use it for viewing existing settings, not just changing them. The design needs to communicate the values of those settings at a glance.

Experienced users have strong expectations for preference editors, account settings, and user profiles being in familiar places and behaving in familiar ways. Break these expectations at your own peril!

How

First, make it findable. Most platforms, both mobile and desktop, have a standard place to find application-wide preferences—follow the conventions, and don’t try to be overly clever. Likewise, websites where people sign in usually put links to account settings and profiles where the username is shown, often in the upper-right or -left corner.

Second, group the properties into pages, and give those pages names that make it easy to guess what’s on them. (Sometimes all the properties or settings fit on one page, but not often.) Card-sorting exercises with representative users can help you figure out the categories and their names. An outrageously large number of properties may require a three- or four-level hierarchy of groups, but be careful that users don’t get frustrated at having to click 53 times to reach commonly needed properties.

Third, decide how to present these pages. Tabs, “Two-Panel Selector”, and “One-Window Drilldown” (Chapter 5) with an extensive page “menu” on the top page seem to be the most common layouts for “Settings Editor”.

2. Organizing the Content: Information Architecture and Application Structure

The design of the forms themselves deserves an entire chapter. See Chapter 8 for patterns and techniques used in forms.

Finally, should you immediately apply changes that the user makes, or offer Save and Cancel buttons? That may depend on the type of settings you're working with. Platform-wide settings seem to be applied immediately when changed; settings on websites mostly use Save buttons; and application settings and preferences can go either way. It may not be a huge usability issue in any case. Follow an established convention if there is one, or see what the underlying technology requires; test it with users if you still have open questions.

Examples

The screenshot shows the Gmail Settings page. At the top, there's a navigation bar with 'Gmail' and a search field. On the left, a sidebar lists inbox categories like 'Compose', 'Inbox' (23,875), 'Starred', 'Snoozed', 'Important', 'Sent', 'Drafts' (2), '[Imap]/Drafts', 'Brilliant Forge Invit...', 'Career Coach', 'Deleted Messages', 'DESIGNING INTERFACES', 'Junk' (18), and 'Notes'. Below this is a blurred contact section. The main area is titled 'Settings' and contains several tabs: General (selected), Labels, Inbox, Accounts and Import, Filters and Blocked Addresses, Forwarding and POP/IMAP, Add-ons, Chat, Advanced, Offline, and Themes. Under the 'General' tab, there are various configuration options:

- Language:** Gmail display language: English (US) (dropdown menu) | Change language settings for other Google products | Show all language options
- Phone numbers:** Default country code: United States (dropdown menu)
- Maximum page size:** Show 50 conversations per page
- Undo Send:** Send cancellation period: 5 seconds
- Default reply behavior:** Reply (radio button selected) | Reply all (radio button)
- Images:** Always display external images (radio button selected) | Ask before displaying external images (radio button) - This option also disables dynamic email.
- Dynamic email:** Enable dynamic emails (checkbox checked) - Display dynamic email content when available. ("Images" setting must be set to "Always display external images") | Dynamic email development
- Smart Compose:** Writing suggestions on (radio button selected) | Writing suggestions off (radio button) | Feedback on Smart Compose suggestions
- Smart Compose personalization:** Personalization on (radio button selected) | Personalization off (radio button)
- Default text style:** Sans Serif (dropdown menu) | A (font size dropdown) | X (bold dropdown) | This is what your body text will look like.

2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows the Gmail settings interface. On the left, there's a sidebar with blurred contact names like 'sof', 'yfb', 'hy', 'l sh', and 'l'. Below the sidebar are icons for people, a gear, and a phone. The main content area has several sections:

- Experimental Access:** A section with a greyed-out background. It includes a link "Learn more" and a note: "Enable experimental access - Gain early access to features still under development. Google may email you to request feedback on these features. Experimental features (indicated by ⓘ) may be removed without notice."
- Conversation View:** A section with a greyed-out background. It includes a link "Learn more" and two options: "Conversation view on" (selected) and "Conversation view off". A note says "(sets whether emails of the same topic are grouped together)".
- Nudges:** A section with a greyed-out background. It includes a link "Learn more" and two options: "Suggest emails to reply to" (selected) and "Suggest emails to follow up on". A note says "- Emails you might have forgotten to respond to will appear at the top of your inbox" and "- Sent emails you might need to follow up on will appear at the top of your inbox".
- Hover actions:** A section with a greyed-out background. It includes a link "Learn more" and two options: "Enable hover actions" (selected) and "Disable hover actions". A note says "- Quickly gain access to archive, delete, mark as read, and snooze controls on hover."
- Send and Archive:** A section with a greyed-out background. It includes a link "Learn more" and two options: "Show 'Send & Archive' button in reply" (selected) and "Hide 'Send & Archive' button in reply".
- Smart Reply:** A section with a greyed-out background. It includes a link "Learn more" and two options: "Smart Reply on" (selected) and "Smart Reply off". A note says "(Show suggested replies when available.)"
- Desktop notifications:** A section with a greyed-out background. It includes a link "Learn more" and three options: "New mail notifications on" (selected), "Important mail notifications on", and "Mail notifications off". A note says "(allows Gmail to display popup notifications on your desktop when new email messages arrive)".
- Stars:** A section with a greyed-out background. It includes a note: "Drag the stars between the lists. The stars will rotate in the order shown below when you click successively. To learn the name of a star for search, hover your mouse over the image." Below it are "Presets": "1 star", "4 stars", and "all stars".

Figure 2-43. Google

2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows the 'General' section of the Facebook account settings. The left sidebar lists various categories: General, Security and Login, Your Facebook Information, Privacy, Timeline and Tagging, Location, Blocking, Language, Face Recognition, Notifications, Mobile, Public Posts, Apps and Websites, Instant Games, Business Integrations, Ads, Payments, Support Inbox, and Videos. The 'General' tab is selected. The main content area displays account details:

Name	Charles Brewer	Edit
Username	https://www.facebook.com/charles.brewer	Edit
Contact	Primary: [REDACTED]	Edit
Ad account contact	[REDACTED]	Edit
Temperature	Fahrenheit	Edit
Manage Account	Modify your legacy contact settings or deactivate your account.	Edit
Identity Confirmation	Confirm your identity to do things like run ads related to politics and issues of national importance.	View

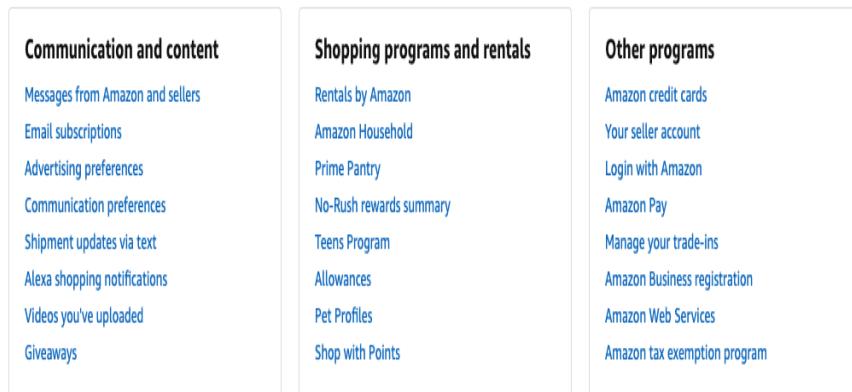
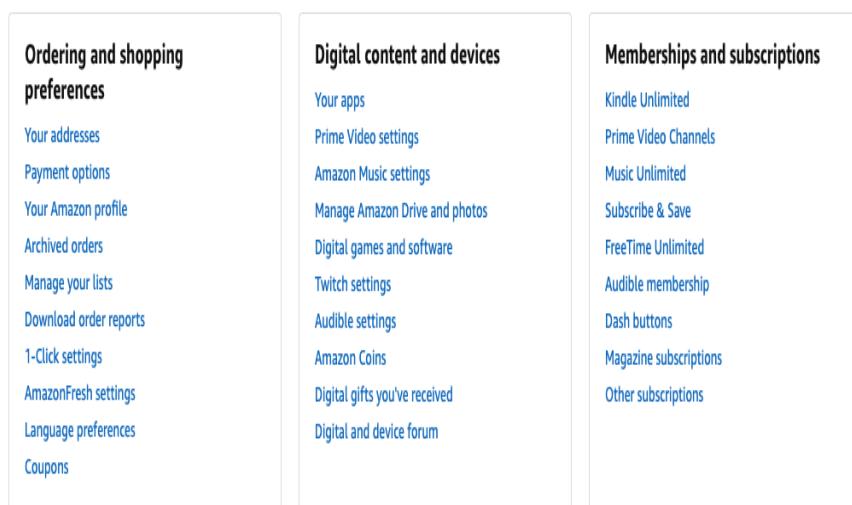
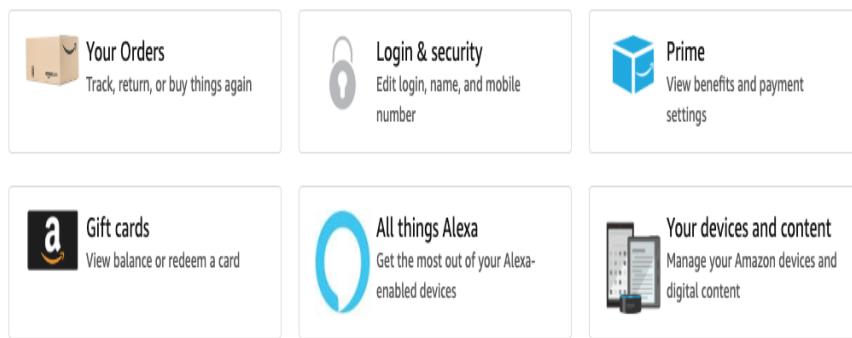
Below the table, there are links for About, Create Ad, Create Page, Developers, Careers, Privacy, Cookies, Ad Choices, Terms, Account Security, Login Help, and Help. At the bottom, it says Facebook © 2019 and provides language options: English (US), Español, Français (France), 中文(简体), العربية, Português (Brasil), Italiano, 한국어, Deutsch, हिन्दी, 日本語, and a plus sign for more.

Figure 2-44. Facebook

The screenshot shows the top navigation bar of an Amazon Prime account page. It includes the Amazon logo, a search bar, user information (Hello, Charles), account links (Account & Lists, Orders, Prime), and a shopping cart icon showing 4 items. Below the navigation, there are links for Deliver to Charles, Buy Again, Browsing History, Charles's Amazon.com, Today's Deals, Gift Cards, Whole Foods, Sell, Prime Video (Stream movies & TV shows), and Prime Video (Stream movies & TV shows).

Your Account

2. Organizing the Content: Information Architecture and Application Structure



Your recently viewed items and featured recommendations

Selected items

Page 1 of 2



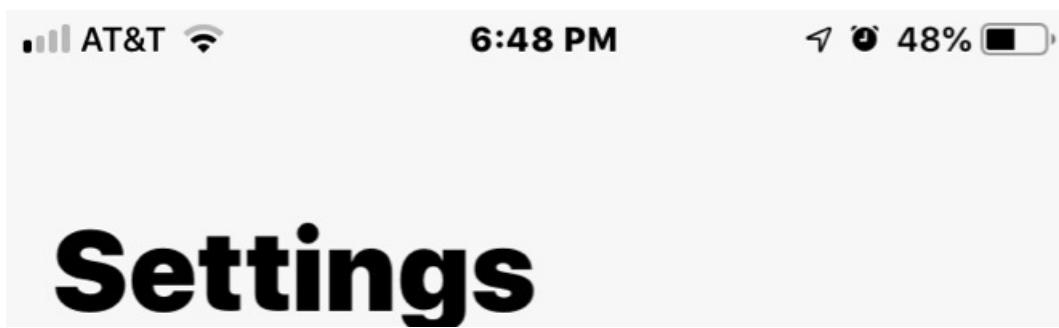


Figure 2-45. Amazon

Amazon offers an outrageously large number of properties that require a deep hierarchy of pages. The designers mitigated some of the problems, however. For instance, they put a list of shortcuts on the top-level page; these are probably the items users look for most often. They put a search box on the top. And by putting lists of items, they show users which items fall into which categories.

Google ([Figure 2-43](#)) and Facebook ([Figure 2-44](#)) both use tabs to present the pages of their profile editors. Amazon has one single link for all account-related information: “Your Account” (see [Figure 2-45](#)). This “Menu Page” lists account settings alongside order information, credit card management, digital content, and even community and wish-list activity. The clean, tight page organization is terrific—if I have any questions about what’s going on with my relationship to Amazon.

Google, Facebook, and Amazon have huge settings, preferences and configuration management issues related to their services. Customers must access these settings from time to time in order to review or change them. All have opted for a strong organization system to categorize their settings and preferences. Google and Facebook use tabs to organize the settings into major categories, with screens in each that are in turn sectioned out with titles and groups of controls to allow for comprehension and relatively easy access. Amazon places its most frequently used settings and configurations at the top of the settings screen, with special formatting as giant buttons. Selecting one allows the user to drill down into the appropriate category of settings. Below this is a grid of cards, each labeled with its settings category name, and displaying a list of links to each subcategory within. All three use strong information architecture and navigation to bring some understandable structure to a complicated part of their platform. While it’s not painless, the user has a good chance of eventually finding and changing the setting they seek.



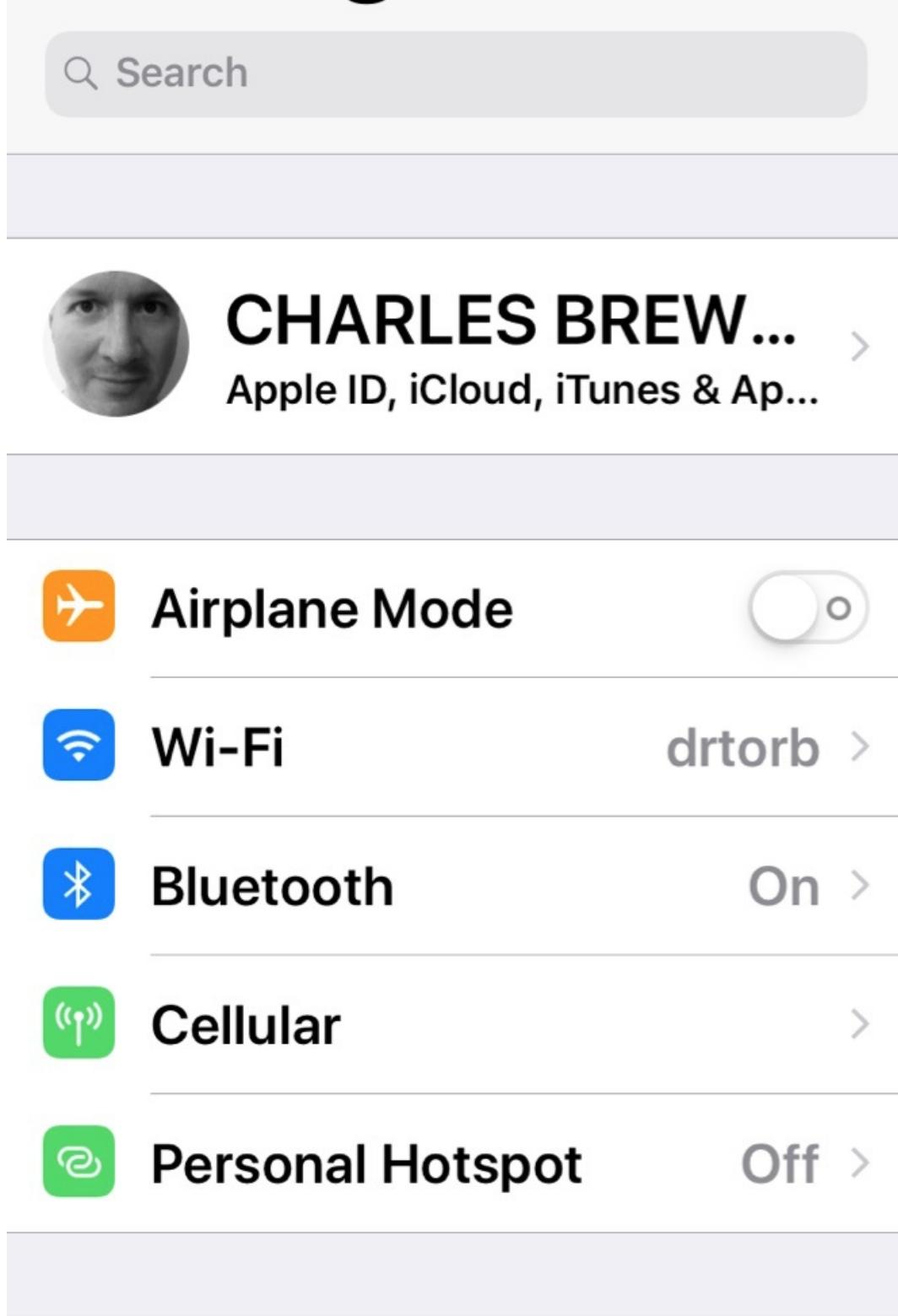


Figure 2-46. Settings

2. Organizing the Content: Information Architecture and Application Structure

In Apple's mobile OS, iOS, there are many settings. Some are for the whole device, and some are for individual apps on the iPhone. Apple has opted for a single scrolling list. Some order is provided by putting the most critical and frequently used settings at the top. The items in the long list are also grouped to help with navigation and selection.

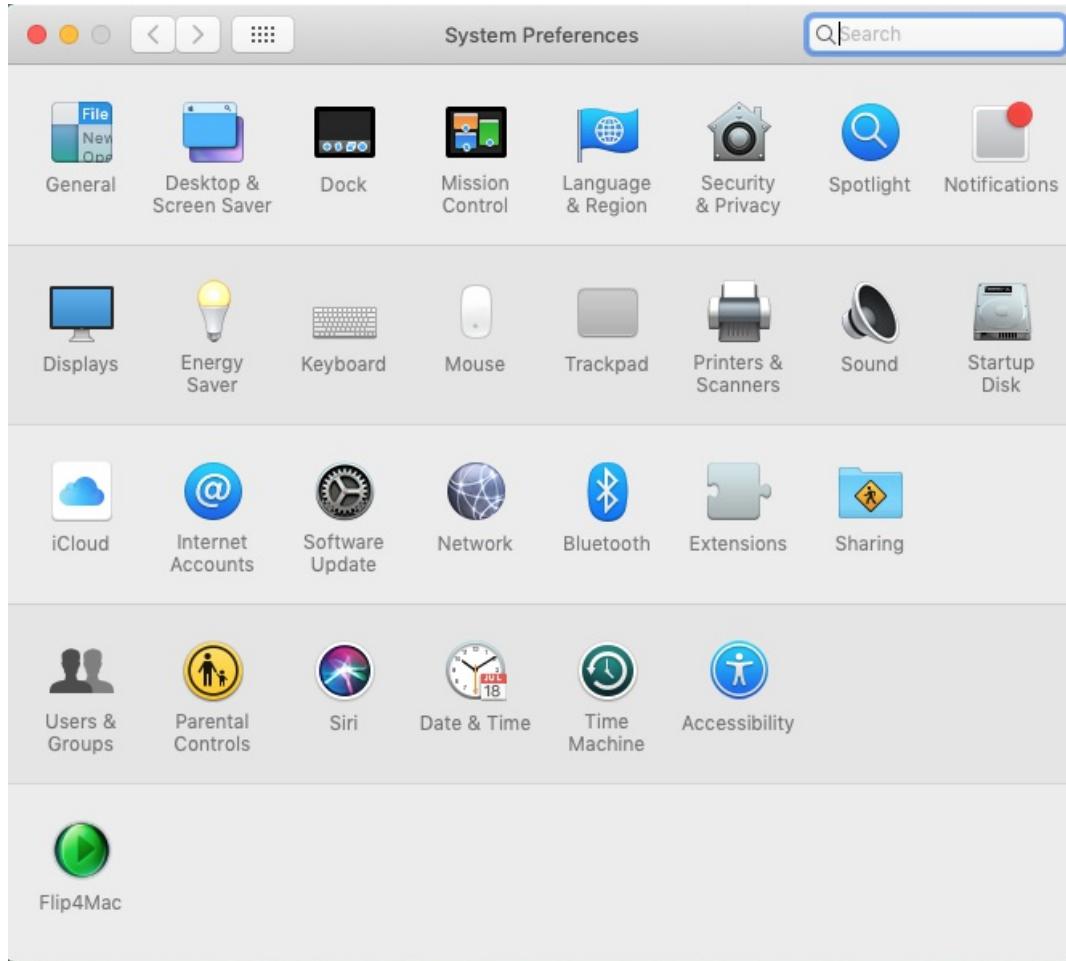


Figure 2-47. MacOS

For its desktop OS, MacOS, Apple has opted for a panel of categories for its system settings. These are marked by sections, icons and labels to help with understanding the categories and selecting the right one.

Alternative Views

What

2. Organizing the Content: Information Architecture and Application Structure

Let the user choose among alternative views to the same information that are substantially different.

Use when

You’re building something that views or edits a complex document, list, website, map, or other content. You may face design requirements that directly conflict with each other. You can’t find a way to show both feature set A and feature set B at the same time, so you need to design both separately and let the user choose between them.

Why

Try as you might, you can’t always accommodate all possible usage scenarios in a single design. For instance, printing is typically problematic for websites because the information display requirements differ—navigation and interactive gizmos should be removed, for instance, and the remaining content reformatted to fit the printer paper.

There are several other reasons for “Alternative Views”:

Users have preferences with regard to speed, visual style, and other factors.

A user might need to temporarily view data through a different “lens” or perspective in order to gain insight into a problem. Consider a map user switching between views of street information and topographic information (see [Figure 2-48](#) at the top of the pattern).

If a user is editing a slideshow or website, for instance, he may do most of his editing while using a “structural” view of the document, containing editing handles, markers for invisible content, layout guides, private notes, and so on. But sometimes he will want to see the work as an end user would see it.

How

Choose a few usage scenarios that cannot easily be served by the application’s or site’s normal mode of operation. Design specialized views for those scenarios, and present them as alternatives within the same window or screen.

In these alternative views, some information might be added and some might be taken away, but the core content should remain more or less the same. A common way to

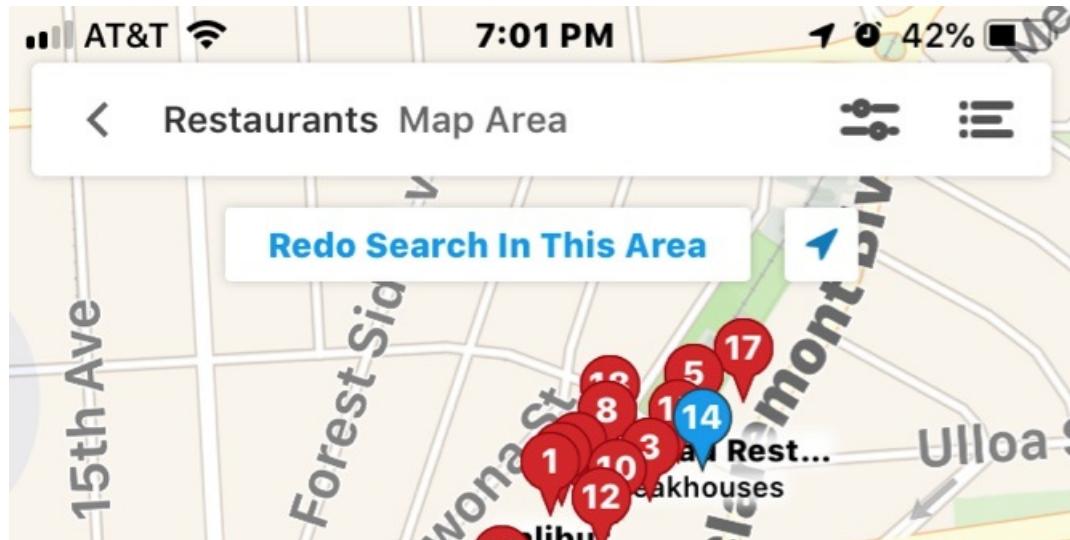
2. Organizing the Content: Information Architecture and Application Structure
switch views is to change the rendering of a list; file finders in both Windows and Mac OS let users switch from lists to “Thumbnail Grid” to “Tree Table” to “Cascading Lists” to “Carousel”, for instance.

If you need to strip down the interface—for use by a printer or screen reader, for instance—consider removing secondary content, shrinking or eliminating images, and cutting out all navigation but the most basic.

Put a “switch” for the mode somewhere on the main interface. It doesn’t have to be prominent; PowerPoint and Word used to put their mode buttons in the lower-left corner, which is an easily overlooked spot on any interface. Most applications represent the alternative views with icons. Make sure it’s easy to switch back to the default view, too. As the user switches back and forth, preserve all of the application’s current state—selections, the user’s location in the document, uncommitted changes, undo/redo operations, and so on—because losing them will surprise the user.

Applications that “remember” their users often retain the user’s alternative-view choice from one use to the next. In other words, if a user decides to switch to an alternative view, the application will just use that view by default next time. Websites can do this by using cookies; desktop applications can keep track of preferences per user; an app on a mobile device can simply remember what view it used the last time it was invoked. Web pages may have the option of implementing “Alternative Views” as alternative CSS pages. This is how some sites switch between ordinary pages and print-only pages, for example.

Examples



2. Organizing the Content: Information Architecture and Application Structure

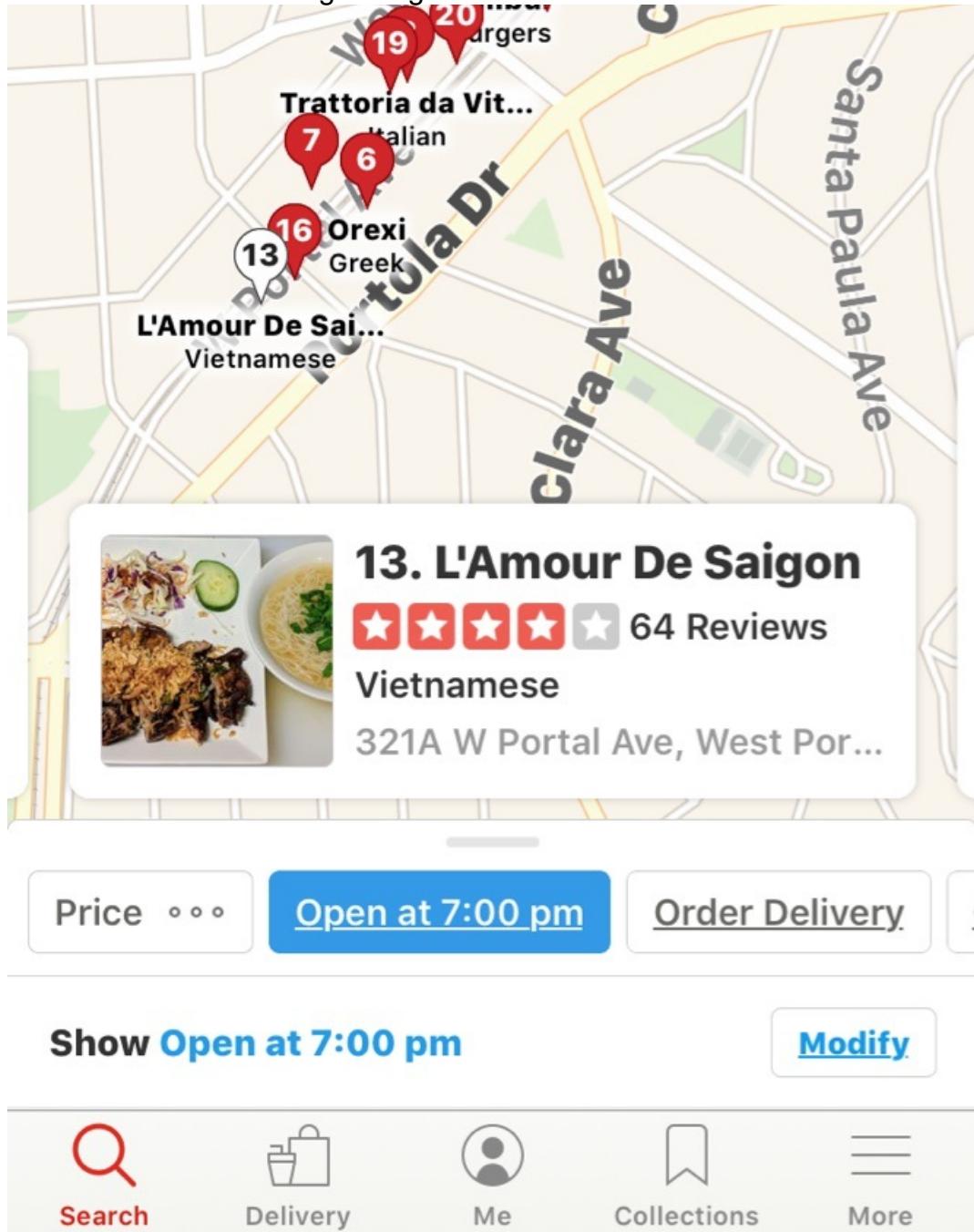
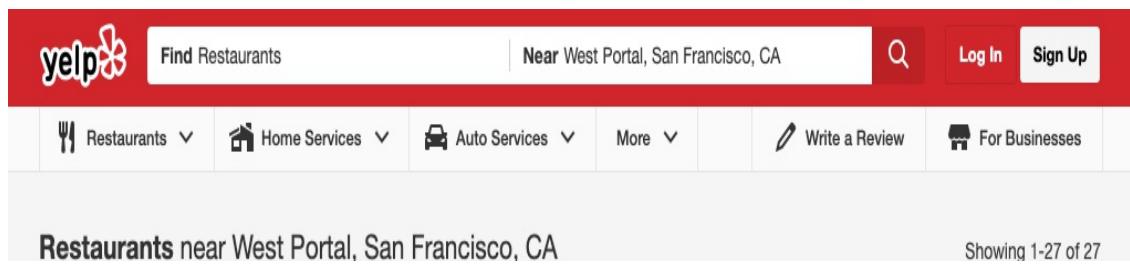


Figure 2-48. Yelp iOS Map and List screens



2. Organizing the Content: Information Architecture and Application Structure

San Francisco, CA > West Portal

All Filters \$ \$\$ \$\$\$ \$\$\$\$ Open Now Delivery Takeout Good for Dinner

1. Trattoria da Vittorio



★★★★★ 654 reviews
\$\$ - Italian, Pizza
6 Most viewed Italian place in West Portal

Offers takeout and delivery Start Order

2. Orexi



★★★★ 340 reviews
\$\$ - Greek, Mediterranean, Desserts
🍴 Popular for its Moussaka

3. Bursa



★★★★★ 751 reviews
\$\$ - Middle Eastern, Turkish, Mediterranean
6 Most viewed dinner spot in West Portal

4. Xiao Loong Restaurant



★★★★★ 265 reviews
\$\$ - Chinese
🍴 Popular for its Mongolian Beef

Offers takeout Start Order

5. Calibur



★★★★★ 354 reviews
\$\$ - Burgers, American (Traditional)
🍴 Popular for its Veggie Burger

Offers takeout and delivery Start Order

Bubba Gump San Francisco - Seafood Restaurant

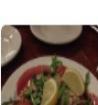
bubbagump.c... (415) 781-4867 [Visit Website](#)

Seafood Restaurants - McCormick and Kuleto's

mccormickan... (415) 929-1730 [Visit Website](#)

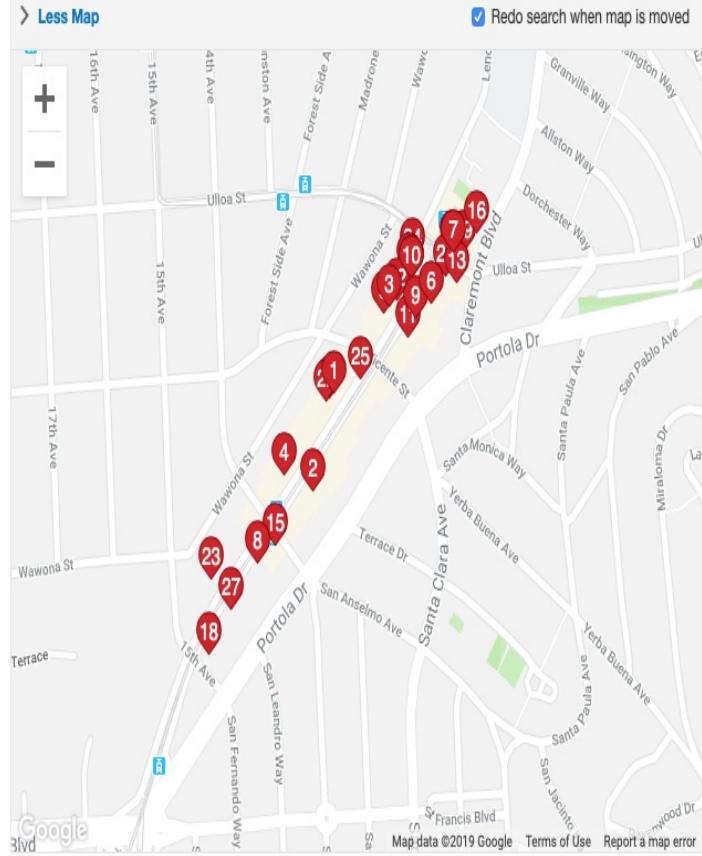
Ad A local favorite with fresh, seasonal menus, private rooms & more. Reserve now. Near Maritime Museum. Private Banquet Rooms. Signature Cocktails. Relaxed Atmosphere. In Ghiradelli Square. Famous Happy Hour. Reserve a Table Online. Fresh Seafood. Decadent Desserts. Amenities: Reserve Your Table Online, Send an e-Gift Card, Award-Winning Happy Hour, Seasonal Dining Menu.
900 North Point St, San Francisco, CA

6. Spiazzo Ristorante



★★★★★ 629 reviews

Less Map Redo search when map is moved



Map data ©2019 Google Terms of Use Report a map error

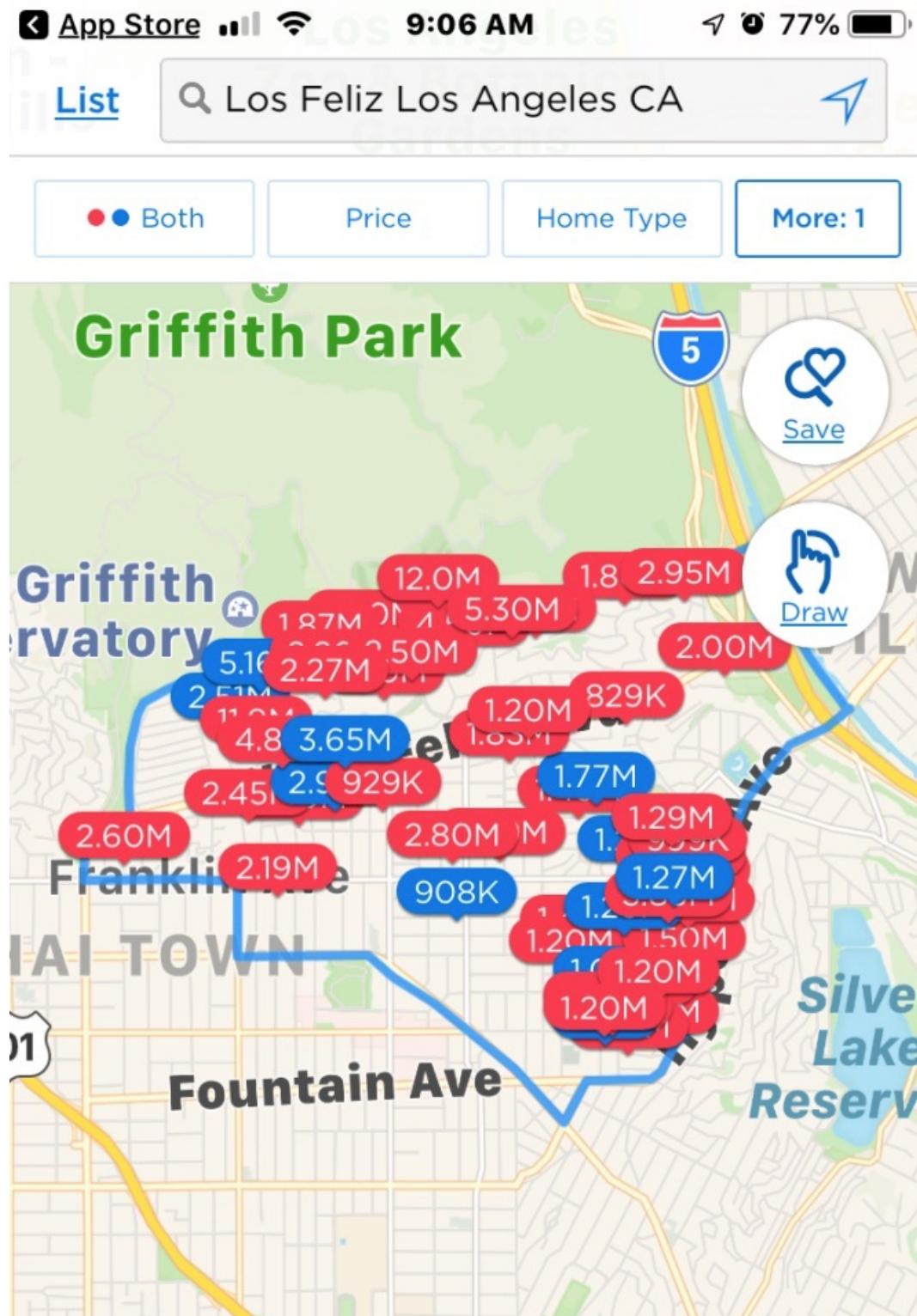
2. Organizing the Content: Information Architecture and Application Structure



\$\$ • Italian, Seafood

Great spot for a date

Figure 2-49. Yelp Desktop Map and List combination screen



2. Organizing the Content: Information Architecture and Application Structure

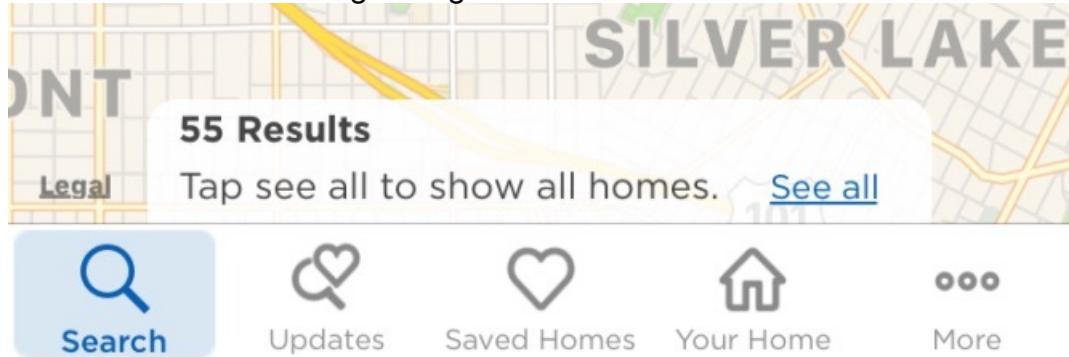


Figure 2-50. Zillow iOS Map and List screens

The screenshot shows the Zillow mobile website for Los Feliz, Los Angeles, CA. At the top, there are links for "Buy", "Rent", "Sell", "Home Loans", and "Agent finder". The Zillow logo is in the center, with "List your rental", "Advertise", "Sign in or Join", and "Help" links to its right. Below the header is a search bar with "Los Feliz Los Angeles CA" and a "Save Search" button. To the left is a map of the Los Feliz area with price markers for various properties. To the right is a grid of real estate listings:

Los Feliz Los Angeles Real Estate & Homes For Sale	
75 results	
	\$1,199,000 3 bds 1 ba 1,111 sqft 2000 Mayview Dr, Los Angeles, CA House for sale
	\$829,000 2 bds 1 ba 1,000 sqft 3317 Wood Ter, Los Angeles, CA House for sale
	\$18,895,000 6 bds 7 ba 15,391 sqft 7405 Glandowner Ave, Los Angeles, CA
	\$23,000,000 3 bds 4 ba 8,000 sqft 7607 Glandowner Ave, Los Angeles, CA

2. Organizing the Content: Information Architecture and Application Structure

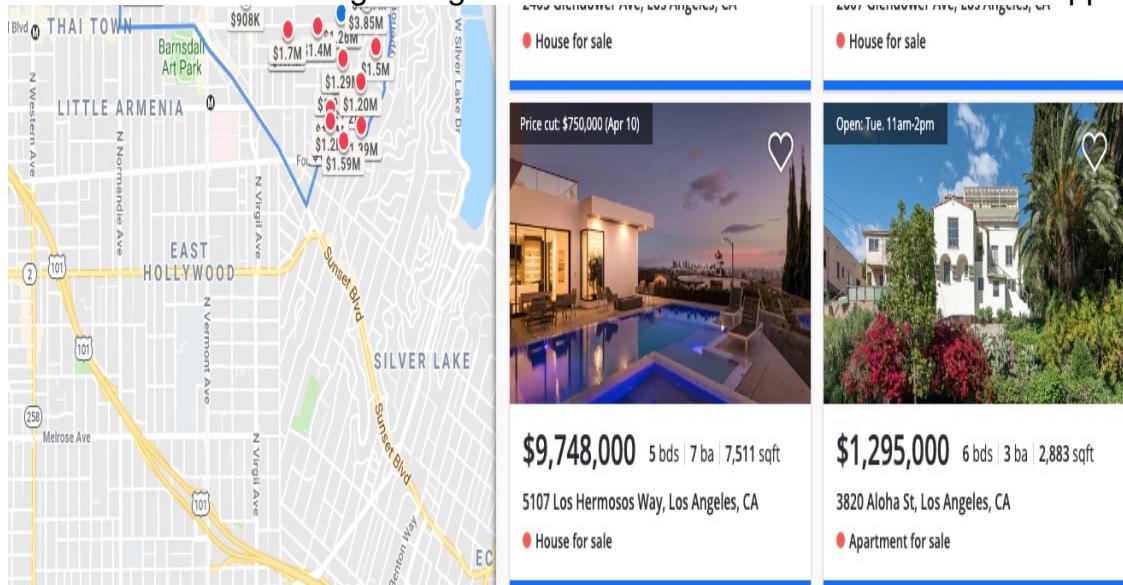


Figure 2-51. Zillow desktop Map and List combination screen

2. Organizing the Content: Information Architecture and Application Structure



Figure 2-52. Apple Keynote

2. Organizing the Content: Information Architecture and Application Structure

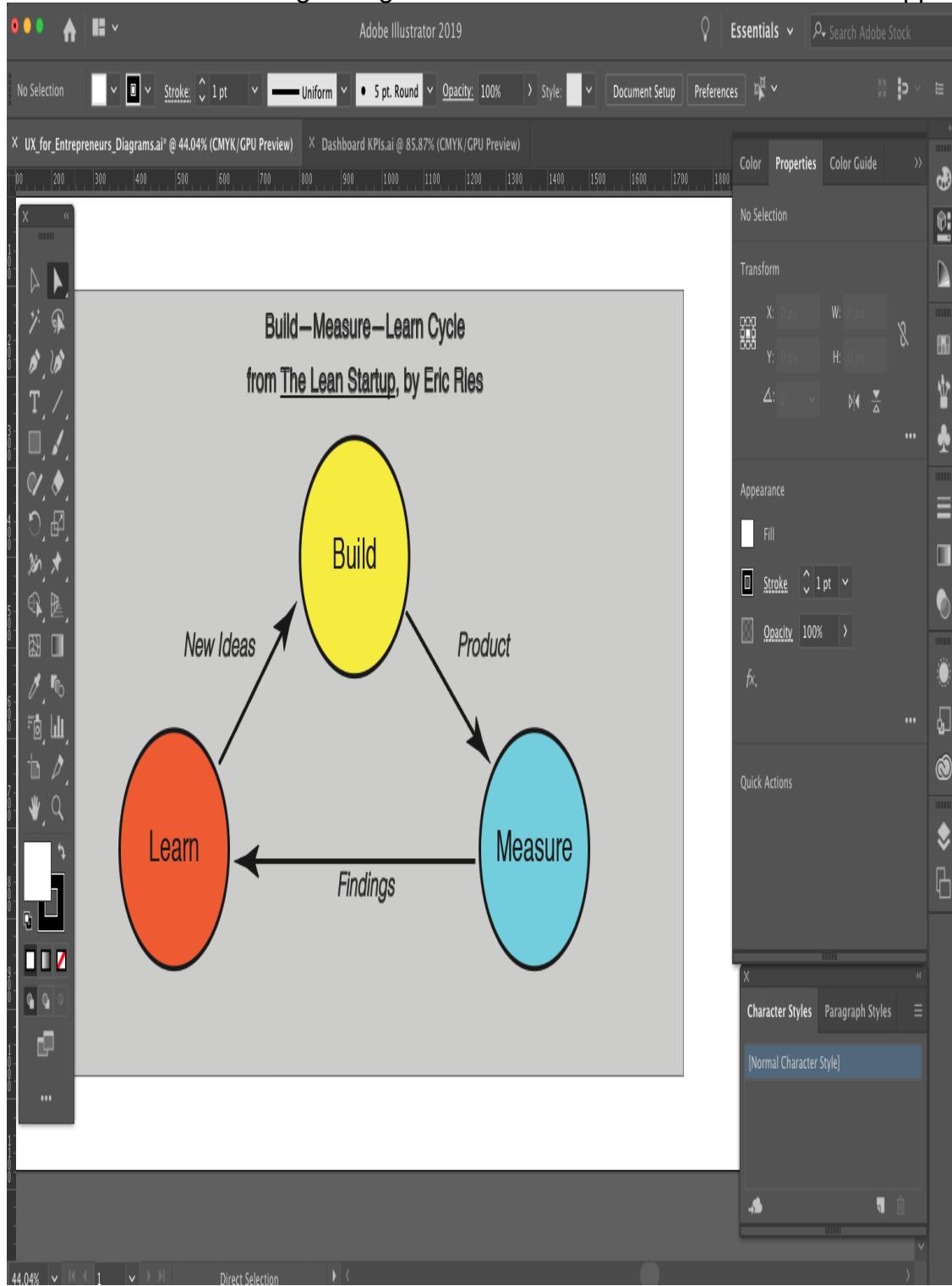


Figure 2-53. Adobe Illustrator

In [Figure 2-51](#) and [Figure 2-52](#), two graphic editors, Apple Keynote and Adobe Illustrator, show different views of a work product. In the slideshow, the user normally

2. Organizing the Content: Information Architecture and Application Structure edits one slide at a time, along with its notes, but sometimes the user needs to see all the slides laid out on a virtual table. (Not shown is a third view, in which Keynote takes over the screen and actually plays the slideshow.)

Adobe Illustrator shows an “outline” view of the graphic objects in the document—most useful if you have a lot of complex and layered objects—and the normal, fully rendered view of the artwork. This mode is explicitly for performance reasons. The outline view puts much less demand on the computer processor and so speeds up work considerably.

In other libraries

- <http://quince.infragistics.com/Patterns/Alternative%20Views.aspx>

Many Workspaces

What

Use multiple top-level tabs, tab groups, streams/feeds, panels or windows so that users can view more than one page, project, file, or context at a time. Let users place these workspaces side by side if possible.

Use when

You’re building an application that views or edits any type of content—websites, documents, images, or entire projects that include many files. A major aspect of choosing this pattern is needing to have different views or task “modes” available at the same time. For example, web surfers often keep many tabs open at the same time so they can switch between various websites, or compare them. Application developers and media creators often need to see and adjust code or controls in an editor window, and at the same time see the output of their work to see if they are getting the desired outcomes--either as compiled and executed code, or as a rendered media object.

Designers of conventional websites don’t generally need to think about this. All the major browsers supply perfectly good implementations of this pattern, using tabs and browser windows. Spreadsheet software such as from Microsoft or Google allow tabbed workspaces to separate a complicated workbook into individual calculation sheets.

Applications whose central organizing structure is a personal “News Stream” may not

2. Organizing the Content: Information Architecture and Application Structure need “Many Workspaces”, either. Email clients, personal Facebook pages, and so forth only show the one “News Stream” that matters to the user; multiple windows don’t add much value. That being said, email clients often let a user launch multiple email messages in different windows. Some Twitter applications can show several filtered streams side by side—they might show a search-based feed, then a feed from a custom list, then a feed of popular retweets, for instance. (See the TweetDeck example in [Figure 2-54](#).)

Why

People sometimes need to switch rapidly between different tasks in the same project or file, or monitor activity across a large number of real time feeds.

People multitask. They go off on tangents, abandon trains of thought, stop working on task A to switch to task B, and eventually come back to something they left hanging. One way or the other, they will multitask, so you might as well support it directly with a well-designed interface for doing so.

Side-by-side comparisons between two or more items can help people learn and gain insight. Let users pull up pages or documents next to each other without having to laboriously switch context from one to another.

This pattern directly supports some Chapter 1 patterns, such as “Prospective Memory” (a user may leave a window open as a self-reminder to finish something) and “Safe Exploration” (because there’s no cost in opening up an additional workspace while leaving the original one where it is).

How

Choose one or more ways to show multiple workspaces. Many well-known applications use the following:

- Tabs
- Separate operating-system windows
- Columns or panels within a window
- Split windows, with the ability to adjust the splitters interactively

2. Organizing the Content: Information Architecture and Application Structure

If you deal with fairly simple content in each workspace—such as text files, lists, or “News Stream”—split windows or panels work fine. More complex content might warrant entire tab pages or windows of their own so that a user can see a larger area at once.

The most complicated cases that I’ve seen involve development environments for entire coding projects. When a project is open, a user might be looking at several code files, stylesheets, command windows (where compilers and other tools get run), output or logfiles, or even visual editors. This means that many, many windows or panels can be open at once.

When users close some web browsers, such as Chrome, the set of workspaces (all open web pages, in tabs and windows) gets automatically saved for later use. Then when the user restarts the browser, her entire set of previously opened web pages is restored, almost as she left it. This is especially nice when the browser or machine has crashed. Consider designing in this feature, as it would be a kindness to your users.

Examples

2. Organizing the Content: Information Architecture and Application Structure

Home @charbrew

Notifications @charbrew

Messages @charbrew

Trending

MediaREDEF @MediaREDEF 1m
On The Future of Tesla and Full Self Driving Cars (@stevecheney)
redef.it/2wT7MF
...

Archillect @archillect 2m

...

Medium @Medium 3m
Interestingly, it's belief that may determine how effective aromatherapy is on a person's mental or emotional state
read.medium.com/eUQZhp
...

Co.Design @FastCoDesign 3m
The Academy of Motion Picture and Arts and Sciences has ruled that films from streaming services like Netflix and Amazon Prime will still be eligible to win Academy Awards st.co/XQiC0ff
...

Unbounce @unbounce 4m
We've said it a million times, but using **#video** on your **#landingpage** is a great way to boost **#engagement** and crank up your **#conversionrate**
...

Notifications @charbrew

Messages @charbrew

Trends for @charbrew

1 #GameofThrones
Game of Thrones: The living and dead face off in the Battle of Winterfell

2 Arya
1.96M Tweets

3 #BattleOfWinterfell
Game of Thrones: The living and dead face off in the Battle of Winterfell

4 #DemThrones
188K Tweets

5 Bran
368K Tweets

6 Night King
703K Tweets

7 Cersei
224K Tweets

8 Lyanna Mormont
145K Tweets

9 Sansa
131K Tweets

10 Theon
220K Tweets

Figure 2-54. Twitter Tweetdeck

Dashboard x **images** x

New Post **Upgrade my plan**

Home charbrew

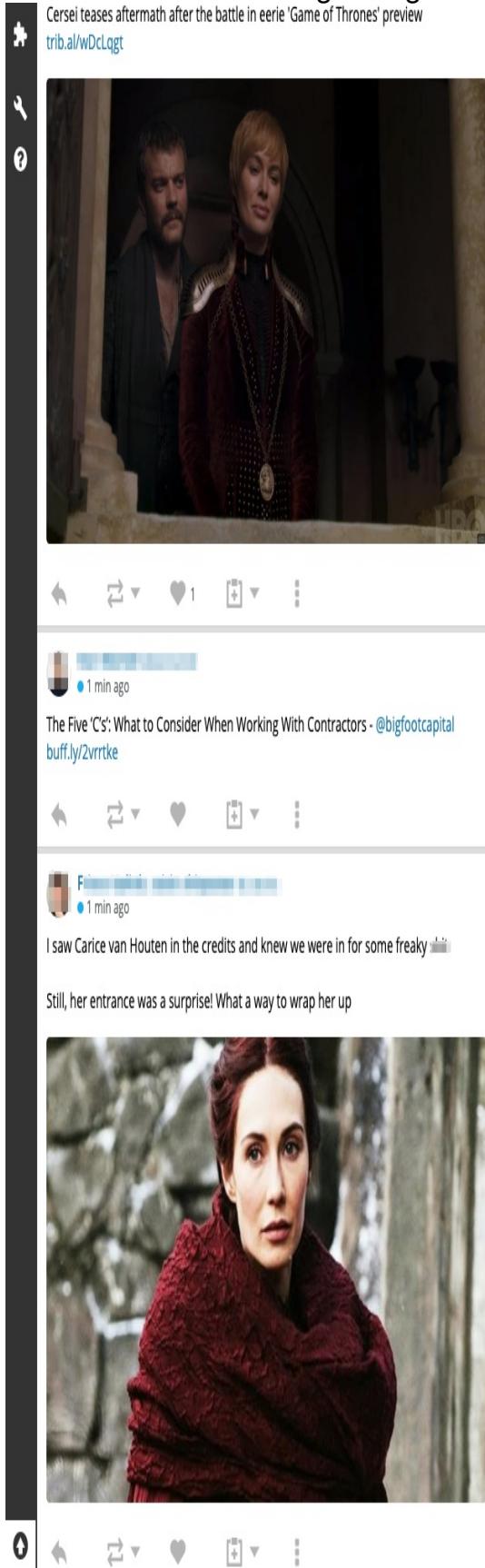
Home Charles Brewer

Mashable @mashable just now

Private LinkedIn Profile 2 hours ago

2. Organizing the Content: Information Architecture and Application Structure

Cersei teases aftermath after the battle in eerie 'Game of Thrones' preview
trib.al/wDcLqgt



Be here, right now. Pay attention to where you are and leave distractions behind.

How do you keep the mind blinkers on?
99 ways to be #ThoughtfullyRuthless

[Read More](#)

Write a comment...

Private LinkedIn Profile
3 hours ago

Ripple's Asheesh Birla Explains How Blockchain Can "Save" Facebook - Potential Ripple-Facebook Collaboration In The Cards? | Oracle Times



The Five 'C's': What to Consider When Working With Contractors - @bigfootcapital
buff.ly/2vrirkE

I saw Carice van Houten in the credits and knew we were in for some freaky 😱

Still, her entrance was a surprise! What a way to wrap her up



T... 3 hours ago

Have you heard of Guide?

Guide is a new social learning app focused on helping high school students learn critical life skills like...

[Read More](#)

2. Organizing the Content: Information Architecture and Application Structure

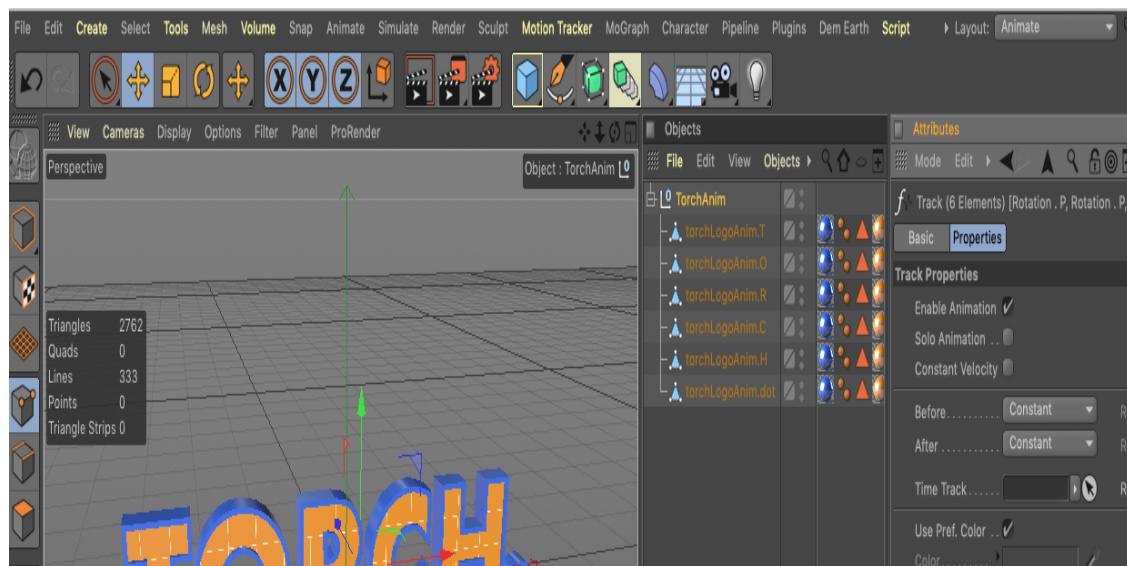
Figure 2-55. Hootsuite

Both Tweetdeck ([Figure 2-54](#)) and Hootsuite ([Figure 2-55](#)) take a multi-panel or multi-stream approach to managing social media feeds.

TweetDeck is a “News Stream”–type application that can show many streams at once: filtered Twitter feeds, non-Twitter sources, and so on. The example in [Figure 2-54](#) shows several typical TweetDeck columns. This maintains the spirit of a “News Stream” by keeping all the updates visible at once; had these columns been in different tabs or windows, a user wouldn’t be able to see all the updates as they happen.

Tweetdeck by default allows the user to see multiple streams side by side from within their own account. In this example, the user can see their main feed, notifications, and their messages at the same time. Normally these are hidden behind various navigation tabs and can only be viewed one at a time. Note also that the fourth panel displays a Twitter-curated list of trending hashtags. Tweetdeck supports many feed panels open at the same time, which is useful for monitoring other Twitter accounts at the same time.

Hootsuite is a social media postings management platform. It’s valuable for individuals, businesses and publishers who want to manage and coordinate their social media accounts in one place. This is useful for pushing out new content or increasing follower and reader interaction across their entire social media ecosystem. In this example, the Hootsuite user has set up their Twitter and LinkedIn accounts. With this side by side feed view, the user can keep track of activity in both accounts (and many more). Posting and responding to each separate account can be carried out from one multi-panel Hootsuite view.



2. Organizing the Content: Information Architecture and Application Structure

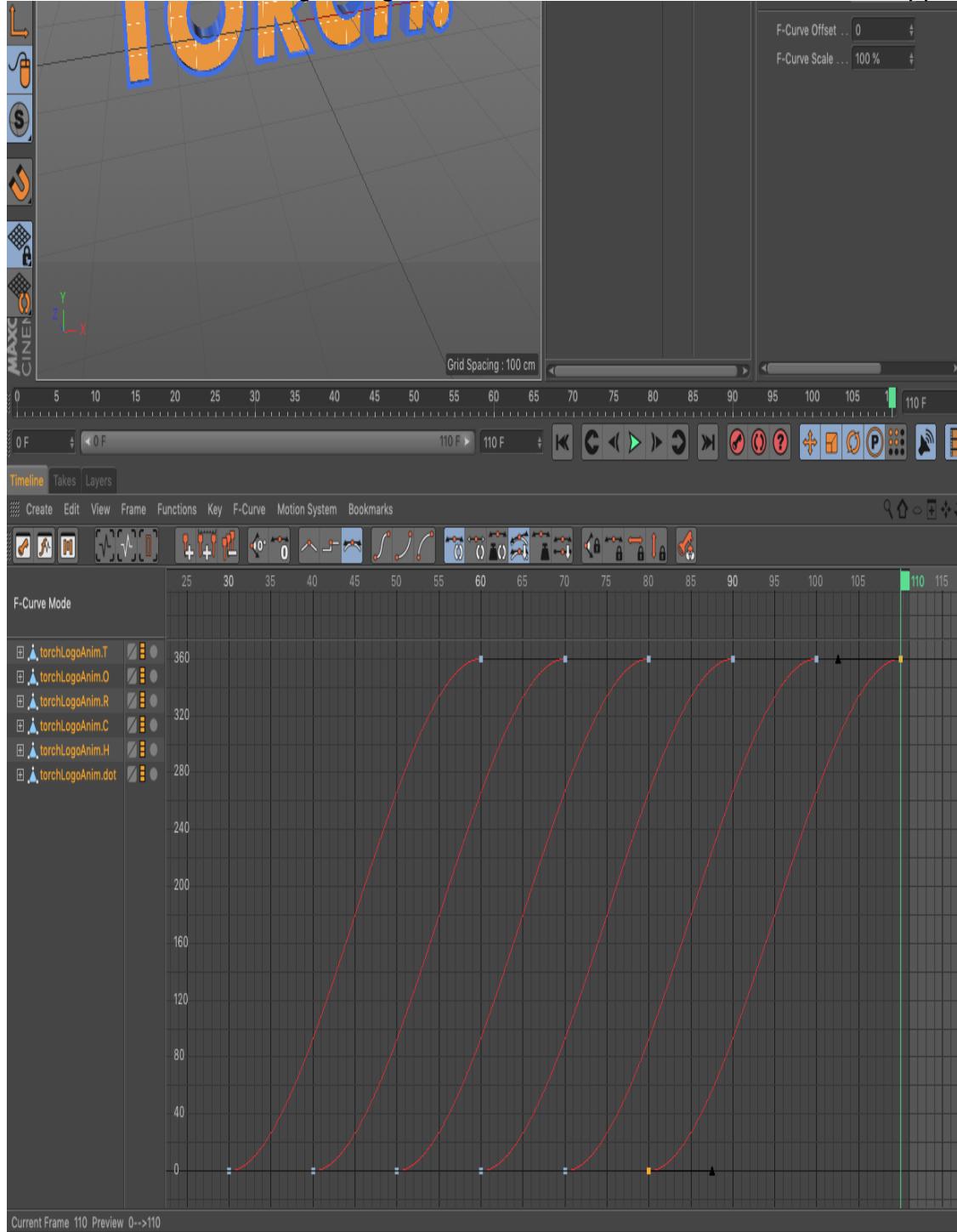


Figure 2-56. Cinema 4D

Cinema 4D ([Figure 2-56](#)) is a tool for creating, rendering and animating three dimensional objects. This desktop application uses a multi-panel approach to give the user simultaneous views of multiple information and tool spaces. There are panels that show

2. Organizing the Content: Information Architecture and Application Structure

source files and versions of the current work file, a central panel that shows the current 3D object, palettes, toolbars and panels that have all the controls for working on the 3D object, and finally a timeline tool that controls the animation of the 3D object.



Figure 2-57. The VSCode programming environment offers multiple panels, each with a different view into the same software: Code editing, source file directory, compiler, rendered code.

VSCode ([Figure 2-57](#)) is a software development application created by Microsoft. Software engineers use it to develop software products and Web applications. Again, there are multiple panels that give views and controls for multiple modes of working simultaneously. The user often switches between each panel to check and rework their code. There are panels that show the directory of source files. There is a robust text editor window for creating, editing and marking up the code in a specific source file. There are also windows for the compiler and monitoring compile errors and states, and finally there is a window for seeing the results of the code execution: A Web page object, a search query, an API call or any other computational output.



Figure 2-58. Multiple websites displayed on a carousel

On tiny mobile screens, you don't have room to show anything side by side. Android and iOS have solved this problem by letting the user open multiple websites at a time, then using a “Carousel” or scrollable list to move between windows ([Figure 2-58](#)). A user swipes up-down or left-right to reach the other windows.

Help Systems

What

Providing labels, explanations and descriptions of how to use the interface are a fundamental part of making usable software. The goal is to provide assistance, answers or training to users when they need it. Ideally, this can take multiple forms that are accessed in different situations. A lightweight or simple help approach can include displaying meaningful in-screen copy, such as descriptive headers and on-screen instructions. Rollover tooltips for all controls is another lightweight approach. When the user wants to learn in more detail, or as a task in itself, a user guide or online manual is the way to go. Sometimes this is in your application itself. Other times it is a separate web site or system. This is a more heavyweight technique, since it involves creating more content. The benefit is this help system information has a long life cycle—it provides value for a long time and doesn't need continuous updating. Users can refer to it again and again, and updates need happen only periodically.

Help systems can be thought of as existing on multiple levels of detail. Some are very close to the user and her task, and are meant to maintain focus on completing it. Others are separate experiences in themselves, and create a separate learning environment, for times when the user wants to focus on learning.

Inline/display

Helping your users in an immediate way starts with the copy that displays on the screen by default. The purpose of inline copy is to communicate what the user is looking at and what the purpose of a given section or component is. Additionally, examples of the inputs you may be soliciting from users also help prevent misformatting. Consider a mix of:

- Meaningful headlines and subheaders
- Instructions: A phrase or sentence directly on the screen to help with a particularly tricky interface.
- Labels for form elements
- Prompts or example inputs either in or next to form elements

Tool tips

Tool tips are brief descriptions or explanations of each component on the screen. On desktop web apps, these display when the user hovers over the interface component. Another method is to display a question mark or other icon (or a link) next to a specific component. Tapping or clicking on the icon displays a short explanation.

Full help system

This is a fully written out user guide that covers all the major features and functions of your app. This is most common for desktop applications. The help system can include descriptions, glossaries, FAQs, how-to's, videos and other information. Help systems often reinforce or replicate user training materials, especially for complicated apps. Help systems can be embedded in the app itself, or can be hosted on a separate web site.

Guided tours

It is now common to deploy step-by-step guided tours or walkthroughs within your application. Many companies now offer this capability. It usually takes the form of a lightbox or other layer on top of the application itself. These display as a series of pop ups or pointers that take the user through a tour, or help them complete a process in a stepwise fashion. These guided tours can be triggered by a variety of events: First time in the app, from the user selecting a “show me how” help option, or by more advanced user behavior analytics and behaviors.

Knowledgebase

Many contemporary customer success software platforms include a Quora-style knowledge base. This consists of a database of questions and answers that are built up over time by the users of the system. This can be created or used just by a customer support team. However, it is now common to open up the knowledge base to customers as well. Often this knowledgebase offers the ability to submit a question or topic, and see a list of most related questions and answers, created by previous users. Knowledgebases are often the core of modern app help systems, and are the first form of customer support-self-support.

Online community

2. Organizing the Content: Information Architecture and Application Structure

Software that is popular enough or specialized enough to have a significant user base can also be supported by an online community of users. This is an advanced technique where the goal is to build a long-term community of users who will help each other, spread usage, and create a culture that fosters platform growth. Sometimes these communities form on their own, as groups on social media platforms or forums like LinkedIn or Facebook, or Reddit. Many companies create, host and moderate their own online user communities to ensure quality discussions. They also create and moderate official groups and communities on social media sites.

Use a mixture of lightweight and heavyweight help techniques to support users with varying needs.

Use when

Every well-designed web site or application should have some form of help. Copy on the screen, prompts in form elements, and tool tips are a must. How will you help beginners become experts? Some form of new user experience mode helps with this. Some users may need a full-fledged help system, but you know most users won't take the time to use it. For complicated applications, full training and help documentation is a must.

Why

Users of almost any software artifact need varying levels of support for the tasks they're trying to accomplish. Someone approaching it for the first time ever (or the first time in a while) needs different support than someone who uses it frequently. Even among first-time users, enormous differences exist in commitment level and learning styles. Some people want to watch a tutorial video, some won't; most find tool tips helpful, but a few find them irritating.

Help texts that are provided on many levels at once—even when they don't look like traditional “help systems”—reach everyone who needs them. Many good help techniques put the help texts within easy reach, but not directly in the user's face all the time, so users don't get irritated. However, the techniques need to be familiar to your users. If they don't notice or open a “Collapsible Panels”, for instance, they'll never see what's inside it.

How

2. Organizing the Content: Information Architecture and Application Structure

Create help on several levels, including some (but not necessarily all) of the help types in the following list. Think of it as a continuum: each requires more effort from the user than the previous one, but can supply more detailed and nuanced information

Meaningful headings, instructions, examples and help text directly on the page.

This can include patterns such as “Input Hints” and “Input Prompt” (both found in Chapter 8). At the same time, try to keep the total amount of text on the page low.

Prompt text

Usually, this text appears in form fields and is replaced by user input.

Tool tips

Use them to show very brief, one-or two-line descriptions of interface features that aren’t self-evident. For icon-only features, tool tips are critical; users can take even nonsensical icons in stride if a rollover says what the icon does! (Not that I’d recommend poor icon design, of course.) Tool tips’ disadvantages are that they hide whatever’s under them and that some users don’t like them popping up all the time. A short time delay for the mouse hover—for example, one or two seconds—removes the irritation factor for most people.

“Hover Tools” (Chapter 6)

These can display slightly longer descriptions, shown dynamically as the user selects or rolls over certain interface elements. Set aside areas on the page itself for this, rather than using a tiny tool tip.

Longer help texts

These can be contained inside “Collapsible Panels” (see Chapter 4).

Introductory material

This can appear in static introductory screens, guided tours, videos, and so on. When a new user starts the application or service for the first time, these materials can immediately orient him toward his first steps (see the “Instant Gratification” pattern in Chapter 1). Users might also be interested in links to help resources. Offer a toggle switch to turn off the introduction—users will eventually stop finding it useful—and

2. Organizing the Content: Information Architecture and Application Structure offer a way back to it elsewhere in the interface, in case a user wants to go back and read it later.

Help shown in a separate window

This text is often in HTML via browsers, but sometimes in WinHelp or Mac Help (a desktop app for help). The help resource is often an online manual—an entire book—reached via menu items on a Help menu, or from Help buttons on dialog boxes and HTML pages.

Live technical support

Usually via chat email, the Web, Twitter, or telephone.

Online community support

This applies only to the most heavily used and invested software—the likes of Photoshop, Linux, Mac OS X, or MATLAB—but users may consider it a highly valuable resource. Host your own community, or use social networking resources for these, or more traditional online forums.

Examples

In-screen help: Labels and Tool Tips

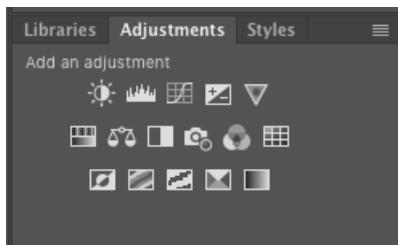


Figure 2-59. Adobe Photoshop: Animated titles as Hover Tools, and Tool Tips



Figure 2-60. Microsoft Excel: Animated titles as hover tools

2. Organizing the Content: Information Architecture and Application Structure

A screenshot of a Microsoft Excel spreadsheet titled "Site Map 20110619.xls". The spreadsheet contains a table of site map data across columns B through M. A red arrow points to the "Formulas" tab in the ribbon, which is currently selected. A red circle highlights a tooltip that reads "Display only the items that meet criteria that you specify". The ribbon also shows other tabs like Home, Layout, Tables, Charts, SmartArt, and Font.

Site Map for Release 1.0 LAUNCH / Releases 1.1-- 1.4													
	B	C	D	E	F	G	H	I	J	K	L	M	
1	1.0 LAUNCH / Releases 1.1-- 1.4												
2	Name	Level 1 ID	Name	Level 2 ID #	Name	Level 3 ID #	Name	Description	Notes	WFISpec?	Visual?	Front end?	Back End?
3	Public Splash Page			0.10.XXX	Blog Article								
4		1.10	Blog Hub										
5													
6													
7		1.20	Register Log In										
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													

Figure 2-61. Microsoft Excel: Tool Tips

Help Systems

2. Organizing the Content: Information Architecture and Application Structure

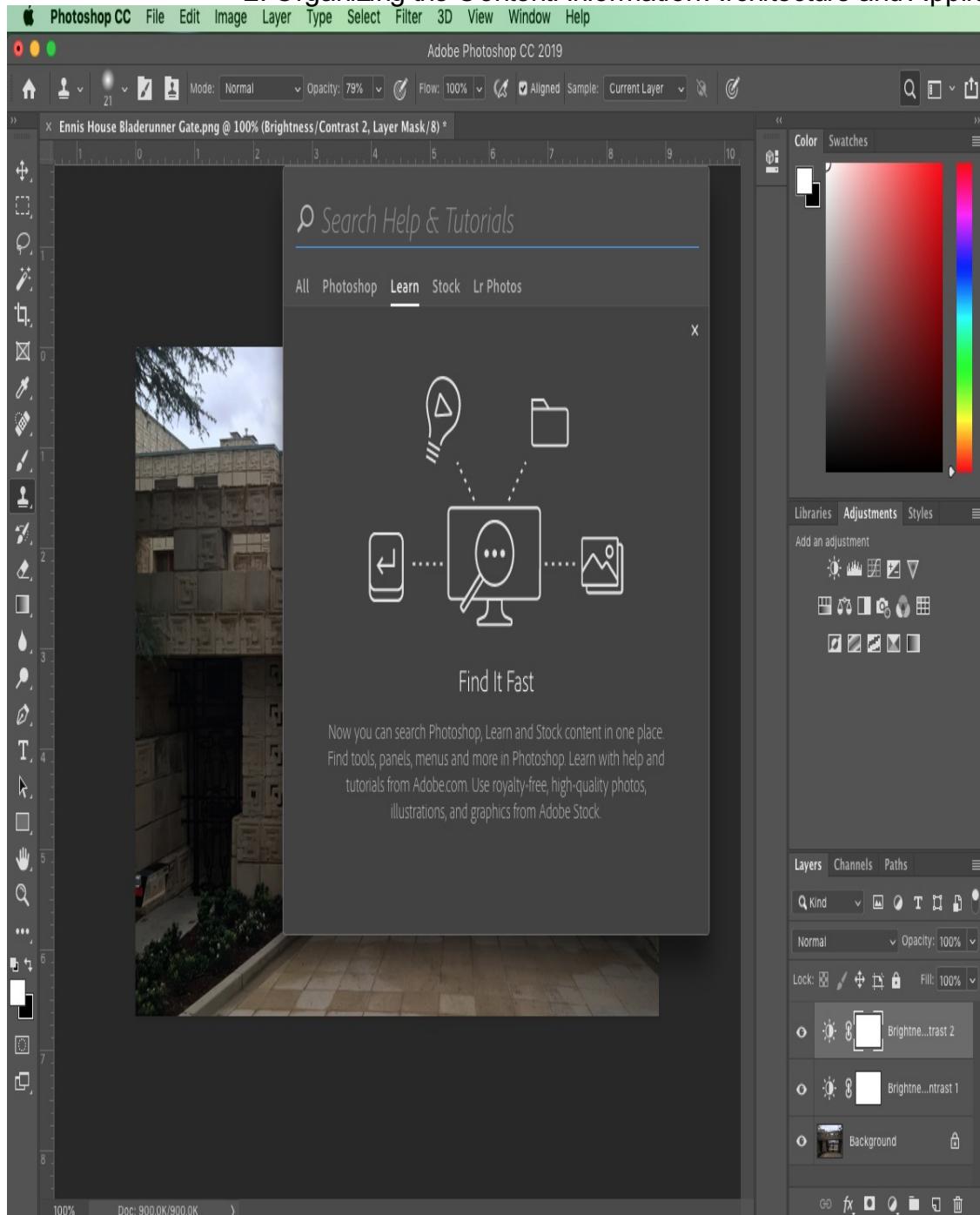


Figure 2-62. Adobe Photoshop Search and Help

2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows a web browser displaying the Adobe Photoshop User Guide. The top navigation bar includes links for Creativity & Design, Marketing & Commerce, PDF & E-signatures, Business Solutions, and Support. There are also icons for user profile, search, and grid. The main header 'ADOBE PHOTOSHOP' has a 'Ps' icon. Below it, a navigation menu includes 'Learn & Support', 'Get Started', 'User Guide' (which is underlined), 'Tutorials', and 'Free Trial'. A 'Buy Now' button is also present. The main content area features a title 'Layer basics' and a search bar 'Search Adobe Support'. A breadcrumb trail shows 'Adobe Photoshop User Guide > Layers'. A dropdown menu is open with the placeholder 'Select an article:' and a downward arrow. The main article content is titled 'About Photoshop layers'. It describes layers as sheets of acetate and provides a diagram showing multiple overlapping layers of a landscape scene. A note below the diagram states: 'Transparent areas on a layer let you see layers below.' The article continues with information about using layers for tasks like compositing and applying layer styles. A sidebar on the right lists 'ON THIS PAGE' links: 'About Photoshop layers', 'Photoshop Layers panel overview', 'Convert background and Photoshop layers', 'Duplicate Photoshop layers', 'Sample from all visible Photoshop layers', and 'Change transparency preferences'. It also indicates 'Applies to: Adobe Photoshop' and 'Last Published: February 15, 2017'. At the bottom right, there's a blue box asking 'Was this page helpful?' with 'Yes' and 'No' buttons.

Adobe

Creativity & Design Marketing & Commerce PDF & E-signatures Business Solutions Support

ADOBE PHOTOSHOP Learn & Support Get Started User Guide Tutorials Free Trial Buy Now

Search Adobe Support

Layer basics

Adobe Photoshop User Guide > Layers Select an article: ▾

About Photoshop layers

ON THIS PAGE

- [About Photoshop layers](#)
- [Photoshop Layers panel overview](#)
- [Convert background and Photoshop layers](#)
- [Duplicate Photoshop layers](#)
- [Sample from all visible Photoshop layers](#)
- [Change transparency preferences](#)

Applies to: [Adobe Photoshop](#)

Last Published: [February 15, 2017](#)

Transparent areas on a layer let you see layers below.

You use layers to perform tasks such as compositing multiple images, adding text to an image, or adding vector graphic shapes. You can apply a layer style to add a special effect such as a drop shadow or a glow.

Introduction to layers

Was this page helpful?

Yes | No

Figure 2-63. Adobe Photoshop Help

2. Organizing the Content: Information Architecture and Application Structure

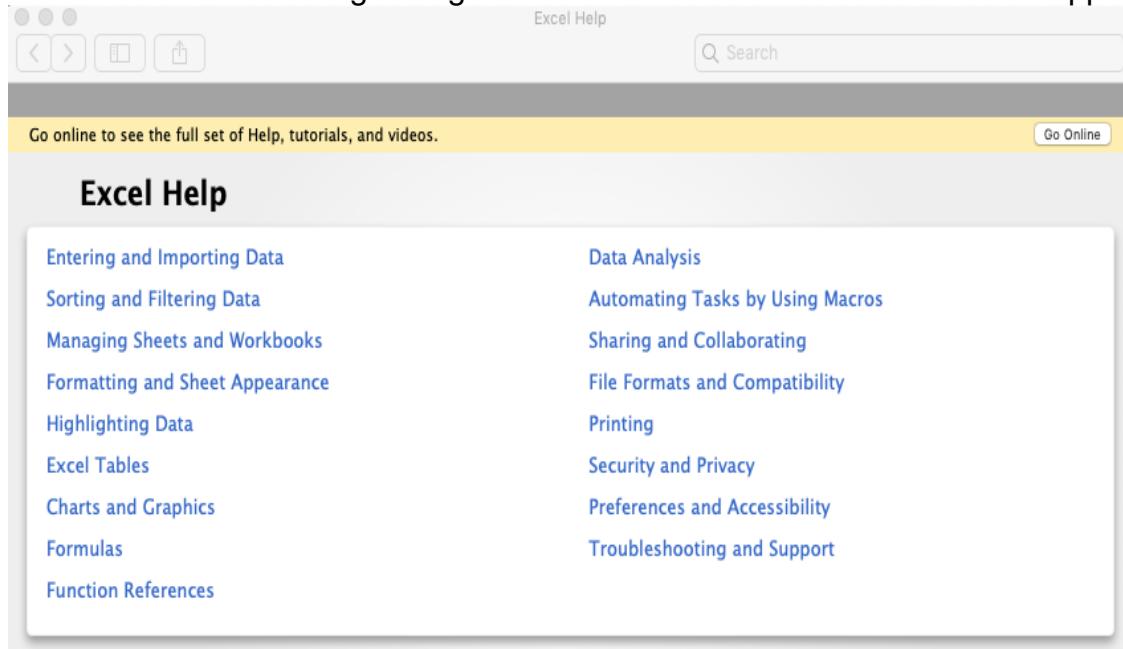


Figure 2-64. Microsoft Excel Help (in the Application itself)

New User Experiences: Guided Instruction

2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows the Wikipedia homepage with a lightbox modal overlay. The modal has a title "Search for 'userlane'" and a sub-section "INFO" containing text about Userlane's purpose. It also features a search bar with the placeholder "Type this or use the autofill button" and a blue "AUTOFILL" button. The background of the page shows the main content area with the "Welcome to Wikipedia" message, featured articles, and other sidebar sections.

Search for "userlane"

In Wikipedia, it is only possible to create a new article by searching for it. This way Wikipedia avoids duplicate articles. So for this demo, search for "userlane".

INFO

Userlane can wait for user input or suggest sample content for an easy user experience during demos and trials.

Type this or use the autofill button

AUTOFILL

(pictured) is

Mohamed Abdullahi Mohamed

er the Romanian
riminalize certain corruption cases.
England Patriots defeat the Atlanta Falcons to
go into overtime.

• The United Nations releases a report stating that the Myanmar Army has committed serious [human rights violations against the Rohingya people](#).

Ongoing: U.S. immigration suspension · Battle of Mosul

Recent deaths: André Salvat · Tom Raworth · Alan Simpson · Tara Palmer-Tomkinson

On this day...

February 13

- 1689 – Glorious Revolution: [Mary Stuart](#) and her husband [William III of Orange](#) were proclaimed co-rulers of England and Ireland.
- 1867 – Work began on the [covering of the Senne](#) (pictured), burying the polluted main waterway in Brussels to allow urban renewal in the centre of the city.
- 1913 – Thubten Gyatso, the 13th Dalai Lama, declared the independence of Tibet from Qing China.
- 1960 – African American college students staged the first of the [Nashville sit-ins](#) at three lunch counters in Nashville, Tennessee, part of a nonviolent direct action campaign to end racial segregation.

Figure 2-65. Lightbox Mode: Instructional panel with Highlighted Step (Userlane)



2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows a web application interface. At the top, there is a dark header bar with the "SPRING ENGAGE" logo on the left, a "Tip: Do You Use Bronto?" button, and user information ("mike@pendo.io", "Settings", "Help", "Log Out") on the right. Below the header is a green navigation bar with links for "What's new", "Dashboard" (highlighted in red), "Smart Offers", "Smart Leads", and "Reports". A large white box in the center contains the text "Click on the Settings Link" with a red arrow pointing to the "Settings" link in the top right corner of the box. The main content area displays "Today: Friday, April 1, 2016" and a section titled "All Channels" with a "Filter by conversion ..." dropdown. Below this is a table showing metrics: Conversions (0, same as last Fri), Visits (0, same as last Fri), Conversion Rate (0%, same as last Fri), Returning Users (0, same as last Fri), Avg. Time on Site (0s, same as last Fri), Revenue (\$0, same as last Fri), and Revenue / Conv. (\$0, same as last Fri). To the right of this table is a chart titled "All Channels Conversions: Cumulative" showing a line graph with zero data points from 8:00am to 6:00pm. A message at the bottom of this chart states "No additional data available for All Channels". On the far left, there is a vertical sidebar with icons for Home, Live Stream, Click, and Click.

2. Organizing the Content: Information Architecture and Application Structure



Figure 2-66. Overlay Mode: Step by step pop ups pointing to UI elements (Pendo)

Online Community

2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows a user interface for a community platform. On the left, there's a sidebar with various icons: a magnifying glass, a person icon, a speech bubble, a plus sign, a gear, and a refresh symbol. Below these are sections for 'Sketch' (with a Sketch app icon), 'Channels', and a purple 'Join community' button.

Sketch

The Sketch community on Spectrum. Sketch is the ultimate tool for designing user interfaces, websites, and icons.

<https://sketchapp.com>

10,905 members

412 members online

Join community

Channels

- Chat >
- # All channels >
- # General >
- # Design Critique >
- # Events >
- # Help >

Popular Search

Galya Iliev · 9h · Pinned
General
Sketch 54 is out now!
Sketch 54 is out now and includes improved snapping behavior, updates to the Fill Popover and the ability to switch to Dark Mode, independent of your system settings! 🎉 Check out the blog post and download Sketch 54 today. Happy...
12 31

Jordan Borth · 12h
General
Sketch 55 Beta with new Spacing feature
Pretty basic implementation for now, but it's promising!
20 17

Mowgli · 12h
Help
issues with undo
I just deleted a whole bunch of objects by accident and undoing it STRAIGHTafter didn't bring it back!
Is this a bug? if so it's a massive one. Not happy, I just lost 2 hours of work.
1 1

Mowgli · 17h
Help
anima stacks and fix size problem
I'm having some problems with anima and stacks. I keep deselecting "fix size" so that it is responsive but the second I click away it reverts to "fix size" which makes it impossible to do anything responsive. Any solutions?
0 1

Bostjan Vidovic · 76d
Help
Group pixel fitting
Hi, i have a problem. I have some elements on my artboard. When i select them i see total size of 40x40px. When i click group then i see a size of 40x41px. Is this a known bug?
1 11

Figure 2-67. User Community as a Learning Resource; Hosted by 3rd party (Sketch community hosted by Spectrum)

2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows the Adobe User Forums homepage. At the top, there's a navigation bar with links for Support Home, Forums Home, News, Product, Explore, and a user profile icon (@ 1). A search bar is also at the top right. Below the navigation, a section titled "Recent Discussions" lists 15 recent posts. Each post includes a thumbnail, the title, the poster's name, and the time ago. To the right of the posts, there are "RELATED COMMUNITIES" links such as Photoshop Scripting, Photoshop Plugin and Companion App SDK, Export Options, Camera Raw, Bridge, Photoshop Mix (Mobile), Photoshop Fix (Mobile), Dimension, Lightroom — The cloud-based photo service, and Lightroom Classic. Further down, there's a "TRANSLATE" section with a link to learn more about translating discussions into various languages like Deutsch, Español, Français, Italiano, Nederlands, Português, Polski, Ελληνικά, русский, Svenska, Čeština, and Türkçe. At the bottom, there are two call-to-action boxes: "You could be the answer!" encouraging users to share knowledge, and "Still have questions?" encouraging users to ask for help.

Recent Discussions

I need to learn how to make and print on CD's in Photoshop
44 minutes ago by Ken Nielsen

When will PS for iPad be released?
45 minutes ago by devins1081032

Problem with virtual memory
2 hours ago by eliteswag

Jagged edges on layer styles
2 hours ago by TorstenChrist

i have a problem with graphics
2 hours ago by franciscoh93570362

Photoshop CS6 - Pen pressure issue Wacom Intuos Pro M (Windows 10)
2 hours ago by anna0152767

Colors more saturated in photoshop after calibration
2 hours ago by patrik27882774

Invert layer changing colors for no reason
2 hours ago by hj87506845

Place an image so it is 100% canvas height/width
3 hours ago by petrik27901450

A0 print correct resolution & size for print
3 hours ago by turlon

Changes to the latest Photoshop brushes?
4 hours ago by AgentJ9

4 PAGES retouched PDF "shrinks" to 1 page
4 hours ago by margiannejulie

Image Blurry When Sent as JPG
5 hours ago by focusworksamanda

Follow AdobeCare

Follow Photoshop

RELATED COMMUNITIES

Photoshop Scripting

Photoshop Plugin and Companion App SDK

Export Options

Camera Raw

Bridge

Photoshop Mix (Mobile)

Photoshop Fix (Mobile)

Dimension

Lightroom — The cloud-based photo service

Lightroom Classic

TRANSLATE

It's easy to translate discussions from English into Deutsch, Español, Français, Italiano, Nederlands, Português, Polski, Ελληνικά, русский, Svenska, Čeština and Türkçe. Look for the translate link under every discussion.

Get a feed of this content

More

You could be the answer!

Share what you know and help fellow community members by answering questions!

Still have questions?

If you can't find your answer by searching our existing discussions, try asking the community

Figure 2-68. Adobe User Forums

Basic buttons—back, next, reload, home—will be familiar to almost all users, but the more obscure items may need to be explained.

2. Organizing the Content: Information Architecture and Application Structure

Text fields use “Input Prompt” to describe themselves. This is a more appropriate choice than “Input Hints” (which would be displayed beside the text fields) because it keeps the window clean and uncluttered. Furthermore, not much knowledge is lost when a user starts typing into the text field, erasing the prompt. See the pattern descriptions for “Input Hints” and “Input Prompt” in Chapter 8.

Other interface elements offer links to the formal help system.

Finally, if all other sources of help are exhausted, a user can turn to the wider user community for advice. We’ve now moved beyond the realm of software design per se, but this is still product design—the user experience extends beyond the bits installed on users’ computers. It includes the interactions they have with the organization, its employees or other representatives, and its website.

Community building like this happens only for products in which users become deeply invested, perhaps because they use the product every day at work or at home. But having an online community of users of some sort is common. It is also a huge advantage for the product. So it is worth considering how to foster such a community.

Tags

What

Tags are a method of classifying and categorizing information by adding labels to it. To put it another way, it is a method of creating or attaching descriptive metadata to an article, a social media post, or any other piece of information. These labels or keywords are called tags. They form an additional kind of classification and navigation system for your app or website: A topic based one. A key aspect of tags is that they are often (but not always) provided by the reader or consumer as a way to help with later grouping of content, searching, browsing, sharing or recall, rather than being supplied at time of creation by the publisher. Multiple tags can be added to the same post or article. With enough tagged content, online consumers can view or browse the content or online activity using tags created by co-users. This is a common feature of discussion boards, blogs and social-enabled apps and sites of all kinds.

Social media has created and spread the near-universal standard of adding the hash or pound symbol “#” to the beginning of a word to mark it specifically as a tag, or “hashtag.” Hashtags create a way for users to link from a single social post or article to a

2. Organizing the Content: Information Architecture and Application Structure
search results feed that contains posts from other people also marked with the same hashtag. This is a powerful way for information to spread rapidly.

Use when

Enable tagging when you want to take advantage of your users' desire to classify, browse, share and promote content associated with topics of interest to them. Tags are a user-generated classification system. When this is generated by your readers or customers and is accessible to others, it becomes a way for them to discover and navigate to content that's of interest. Apps or platforms that have a large amount of information, such as news publishers or social media sites, will add tags to their content, or allow users to tag their posts. In this way, a home-grown topic navigation and discovery system grows along with the content, a complement to whatever formal navigation and IA scheme you have designed into your product.

Why

The purpose of tagging from a design perspective is twofold. First, it promotes increased usage of your app or website because your users are able to find content, media and other information that is highly relevant to the tag or keyword of interest at the moment. Even more, if you have users who actively tag content, then they are becoming more invested in your product or platform because they are creating their own content (usually for social media sharing) using your content. Both of these help increase usage and retention. The second reason to tag your content, or enable your users to tag, is to crowd-source an organic organization scheme for your app or community that would otherwise be too expensive and time consuming to do on your own. These can form quickly around topics of the moment, and they can also be long-lived topics. Over time, a naturally user-centered structure and order can emerge. This can be especially useful if you want to activate or amplify your customers' own interest in finding, reading, sharing, commenting on and more related to topics that are naturally interesting to them.

How

Create or incorporate software that allows words to be added to content or posts as tags. Furthermore, your search function must be able to search the tagged content and create index or results screens that show all content that is tagged with a particular keyword. Tags or hashtags also should be formatted as links automatically. Selecting the tag generates a search based on that tag, and creates a search results page with most relevant

2. Organizing the Content: Information Architecture and Application Structure
or most recent content also tagged with that term.

Examples

The screenshot shows the Stack Overflow homepage with the 'Top Questions' section. The sidebar on the left includes links for Home, PUBLIC, Stack Overflow, Tags, Users, and Jobs, along with a Teams section for Q&A for work. The main content area displays ten questions with their titles, vote counts, answer counts, view counts, tags, and last modification times. To the right of the questions, there is a sidebar for 'Invesco Ltd.' featuring its logo, company name, location (San Francisco, CA), industry (Financial Services), size (Public, 5-10k people), tech stack (Java, node.js, angularjs, reactjs, mysql, spring, spring-boot, oracle, python), and benefits (medical, dental, vision coverage). Below the sidebar is a button to 'View all job openings!'. At the bottom, there is a section for 'Hot Network Questions' with several questions listed.

Rank	Title	Votes	Answers	Views	Tags	Last Modified
1	Modifying Java if statements	0	0	4	java, if-statement	modified 1 min ago dimo414 32.3k
2	How to extract source code from react js build version?	-1	0	6	javascript, html, reactjs, react-native, sass	asked 1 min ago PvDev 101
3	Github commit/push rule, check two files have same list of variables	0	0	10	git, github	modified 1 min ago jhpratt 3,305
4	Can i draw custom donut chart that has each width in teechart?	0	0	2	delphi, gdi, teechart	asked 1 min ago 이정수 1
5	Variable name in string doesn't evaluate	0	1	5	makefile, gnu-make	answered 1 min ago Matt 3,411
6	XRStats.TryGetGPUTimeLastFrame is saying it used 2+ seconds last frame, how can that be?	0	0	3	unity3d	asked 2 mins ago Svp 33
7	how to set showModalBottomSheet to full height	2	2	777	dart, flutter	modified 2 mins ago ibhavikmkwana 681
8	Redefinition of module 'YAML'	0	0	2	objective-c, swift, xcode, frameworks, libyaml	asked 2 mins ago Coel Wu 35
9	Is there a way to detect when the ActionMode overflow menu is opened and closed?	0	0	31	android, android-actionbar, android-actionmode	modified 2 mins ago Freddap 6
10	Cross Site Scripting In JSP	0	0	7		

2. Organizing the Content: Information Architecture and Application Structure

votes answers views [jsp](#) [xss](#) [veracode](#) modified 2 mins ago [Jaydeep Bobade](#) 315 [in javascript](#)

0	1 answer	5 views	Setting up Validation via VBA script	modified 2 mins ago iM Hitesh Surani 3,280	 Do I have to worry about players making "bad" choices on level up?
1	1 answer	7 views	Dynamic convert qspinbox to qdoubleSpinBox	modified 3 mins ago Nikos C. 34.8k	 Weird result in complex limit
0	0 answers	11 views	Fabric upload-symbols fails in Xcode 10.2	modified 3 mins ago Paul Kite 1	 Was it really necessary for the Lunar Module to have 2 stages?
			ios xcode fabric google-fabric xcode10.2		 What does YCWYODTRFDY mean?
					 Is thermodynamics only applicable to systems in equilibrium?
					 What word means to make something obsolete?
					 Need help understanding harmonic series and intervals

Figure 2-69. Stackoverflow, showing relevant tags (each with a blue rounded rectangle) for each question

stack overflow [android] Log In Sign Up

Home Questions tagged [android] Ask Question

PUBLIC Stack Overflow Sponsored links for this tag

- Get started with Android
- Android Developers Blog
- Download Android Studio
- Android Developers on YouTube



Android is Google's mobile operating system, used for programming or developing digital devices (Smartphones, Tablets, Automobiles, TVs, Wear, Glass, IoT). For topics related to Android, use Android-specific tags such as android-intent, android-activity, android-adapter and etc. For questions other than development or programming, but related to the Android framework, use this link: <https://android.stackexchange.com>.

Learn more... Top users Synonyms (7) android jobs

1,191,548 questions Info Newest 27 Featured Frequent Votes Active Unanswered

0 votes can I get all data like steps, heart rate, activities from google fit by using single DataReadRequest?

0 answers I have recently started working with google fit to get different types of data like: Active time Steps count Distance measurement Calories burnt Moving minutes Pace But for retrieving all these data,...

5 views   

asked 2 mins ago  amit pandya 809 8 15

0 votes Manage android fragment creation in an efficient way?

0 answers I have a ViewPager inside which I have 4 tabs with fragments. Previously at app launch, it just loaded the two first fragments by default. Makes sense but problem was each time I navigated to 4th tab ...

0 answers   

asked 5 mins ago  Srinivas Iyer

Featured on Meta

- Unicorn Meta Zoo #1: Why another podcast?
- Announcing the arrival of Valued Associate #679: Cesar Manara
- Data science time! April 2019 and salary with experience
- Should we burninate the [code-smell] tag?

Microsoft Azure Build AI apps with just a few lines of code Try Azure free >

Jobs near you

Backend Software Engineer Startgrid Burlingame, CA go elasticsearch

2. Organizing the Content: Information Architecture and Application Structure

The screenshot shows a list of three questions on Stack Overflow, all tagged with 'android':

- How can I add a custom marker to a map from data I input into FireStore**
0 votes
I have an application that allows me to submit a note containing restaurant information to FireStore, this data is saved in a collection called "notes" and contains fields such as the "Restaurant Name" ...
0 answers
java android firebase google-cloud-firebase
asked 7 mins ago by Andrew Taylor
- AsyncTask in TabLayout**
0 votes
I've an app with TabLayout. Each tab is a Fragment and three of them contain an AsyncTask for doing heavy background operations. See below code example. The backgroundLoadList AsyncTask is calling ...
0 answers
java android android-framents android-asynctask android-tablayout
asked 10 mins ago by Patrick
- Trying to duplicate my Android Studio project and make it run**
-1 votes
Working on an Android app for a university course. I want to make a duplicate so I can add new code/modules without breaking anything. I followed this answer but it did not work for me, I get this ...
1 answer
android android-studio copy project
asked 12 mins ago

On the right side, there is a sidebar titled "Related Tags" with a list of tags and their counts:

- java × 229911
- android-layout × 50794
- android-studio × 44318
- android-framents × 38064
- listview × 34116
- android-intent × 27240
- xml × 26583
- sqlite × 26476
- android-activity × 25535
- eclipse × 25279

[more related tags](#)

Figure 2-70. Stackoverflow filtered by tag

Stack Overflow is a hugely popular question-and-answer discussion board and online community that serves the software developer community. This website consists almost totally of user-generated content in the form of threaded, tagged discussions. In other words, it offers a deep and robust crowd-sourced tag/topic system for finding information. Readers can browse lists of most recent and most popular questions. They also use the tags attached to each post. Participants tag their posts liberally. This allows readers to find discussion threads tagged with the same topic, and closely related topics. Tracking and displaying related tags creates a rich information and navigation browsing ecosystem with high likelihood of readers finding relevant content related to their topic of interest.



2. Organizing the Content: Information Architecture and Application Structure



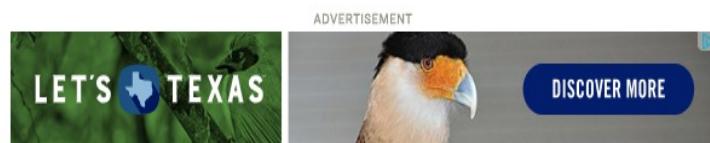
W Dallas-Victory.

STAY

[W Dallas-Victory](#) // Splurge on an ultra-embellished cowboy-chic suite —or just an equally festooned cocktail in the Living Room Bar....

BEFORE YOU GO

Read...[this 1984 profile](#) of Dallas developer Trammell Crow by Joe Nocera



DISCOVER MORE

Bookmark...[this directory](#) of the dozens of Design District businesses

Follow...@DDesignDistrict on [Twitter](#) and [Instagram](#)



0

COMMENTS

SHARE

TAGS: TEXAS TRIP GUIDES,
TRAVEL, DALLAS DESIGN
DISTRICT, THE WANDERER

Never Miss a Story

Sign up for *Texas Monthly's State of Texas* newsletter to get stories like this delivered to your inbox daily.

Enter email address

SIGN UP

RECOMMENDED

01



02



03



04



2. Organizing the Content: Information Architecture and Application Structure

Figure 2-71. Texas Monthly article with tags (powered by Wordpress)

The screenshot shows the homepage of the Teachers & Writers Magazine website. At the top, there is a dark blue header bar with social media icons (RSS, Facebook, Twitter, Instagram, YouTube) on the right. Below the header is the magazine's logo: "TEACHERS & WRITERS" in white and orange text on a blue background, with "MAGAZINE" in white on an orange background. A subtitle "Teaching Creative Writing • Educating the Imagination" is centered below the logo. A horizontal navigation bar follows, featuring links for "LESSON PLANS", "ART OF TEACHING WRITING", "INTERVIEWS", "STUDENT WRITING", "ABOUT US ↓", and "SUBSCRIBE & DONATE". Below the navigation is a large, vibrant photograph of a city street at night, likely San Francisco, with a cable car and festive lights. The main title of the article, "IN A FLASH: KEEPING STUDENTS FOCUSED AND WRITING WITH FLASH FICTION", is displayed in large, bold, dark blue capital letters. Below the title is a thin horizontal line, followed by the article's category "ART OF TEACHING WRITING / FEBRUARY 11, 2019".

IN A FLASH: KEEPING STUDENTS FOCUSED AND WRITING WITH FLASH FICTION

ART OF TEACHING WRITING / FEBRUARY 11, 2019

2. Organizing the Content: Information Architecture and Application Structure

By J.D. Mader

For six years, I was lucky enough to teach writing to at-risk, low-income youth in San Francisco. I worked at two Catholic schools, but for a non-profit named ACHIEVE. I was a shadow. I designed my own curriculum. I was able to work with students in small groups, not a typical classroom. We spent a lot of time discussing life and literature, and I didn't assign much homework. I chose the materials we read, and I was not tied to any kind of timeline. I was the ghost in the machine. It was amazing.

The students I worked with were all between the ages of thirteen and eighteen, but many of them had rough home lives and their educational foundation was shaky at best. Some of the seniors I worked with read at a second-grade level. They did not like to write. Trying to convince them that they could enjoy reading was almost impossible, and it usually took a few months. Trying to convince them that they could be great writers was hard, too. For about four days.

Students came into my writing workshop having read few, if any, books that were not comprised mainly of pictures. At the time, I was working on a novel, though I thought of myself primarily as a short story guy. I love short-form fiction, and part

Search here... 

CATEGORIES

- About T&W
- Art of Teaching Writing
- Featured Posts
- Interviews
- Lesson Plans
- Student Writing
- Uncategorized

Figure 2-72. Teachers & Writers Magazine article with tags as category navigation (powered by Wordpress)

Both Texas Monthly (Figure 2-71) and Teachers & Writers Magazine (Figure 2-72) use the massively popular Wordpress content management system and Web publishing platform to create their online magazines. A prominent feature of Wordpress is the ability to add tags to articles and posts at time of publication. These tags are displayed along with the main article, and link to other posts within the Wordpress system, which generates additional pages for the reader based on the tag they clicked on. In the Texas Monthly example, several tags display at the bottom of an article. The authors have chosen a number of keywords to describe this and other articles. When the user clicks on the tag “Travel,” for example, they can see a list of other travel articles that Texas Monthly has published. This promotes increased time on the site, site circulation and reader engagement.

Teachers & Writers Magazine uses the Wordpress tagging system as well. However, they use selected tags as a more formal topic-based navigation system. The “Categories” module displays a list of topics as a site navigation system. Selecting a given category will display all site articles tagged with that category label. This allows the readers to find articles of interest even as the number of published articles grows.



The screenshot shows a magazine article titled "Designing Tables for Reusability" by Ada Rafalowicz and Huyana Nguyen. The article is part of the "UX Collective" section. A sidebar on the right contains a search bar, a "Save clip" button, and a "Clip format" dropdown menu set to "Article".

2. Organizing the Content: Information Architecture and Application Structure



Figure 2-73. Evernote Screen Clipper with ability to add tags

2. Organizing the Content: Information Architecture and Application Structure

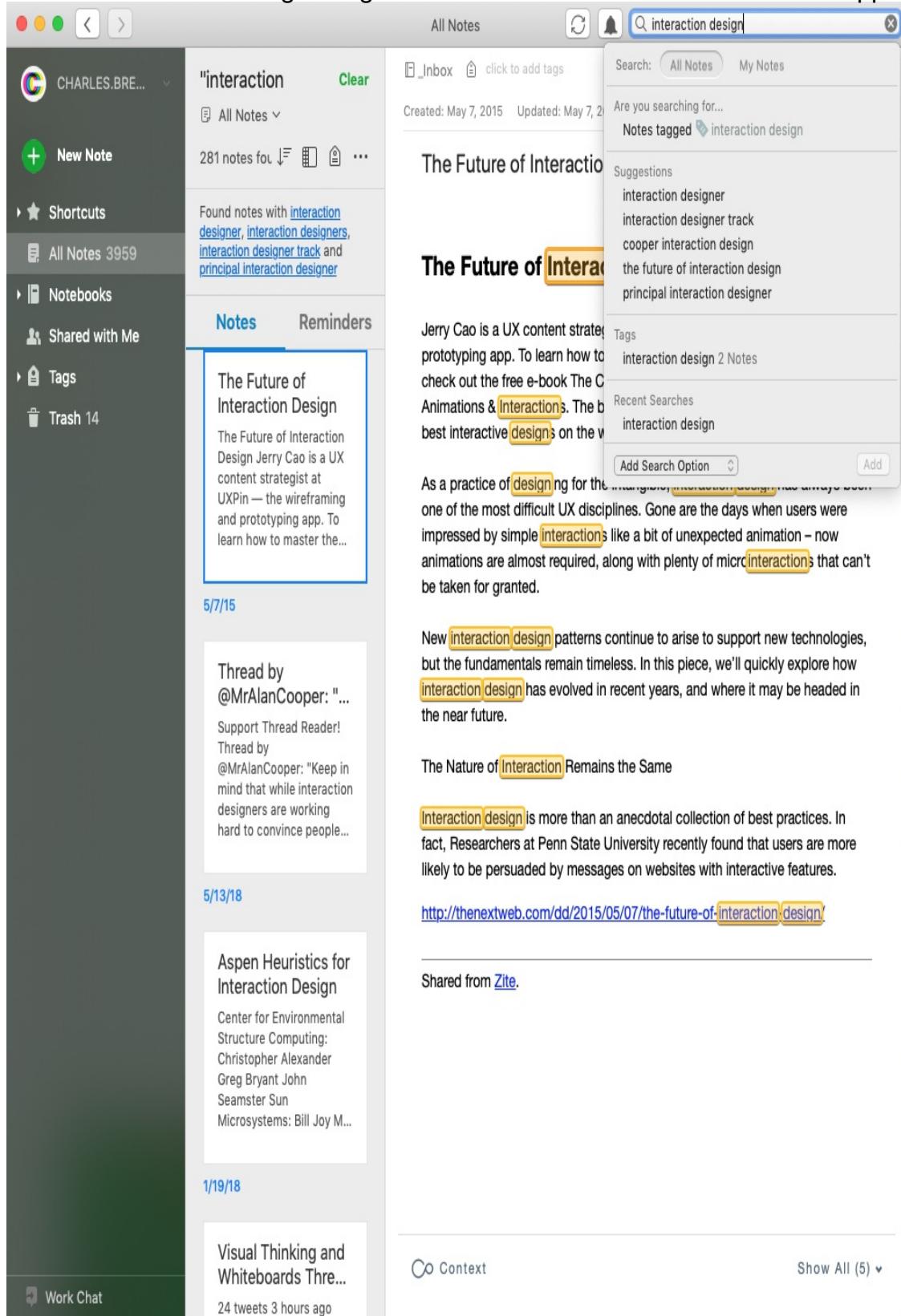


Figure 2-74. Evernote App with ability to search by tags

2. Organizing the Content: Information Architecture and Application Structure

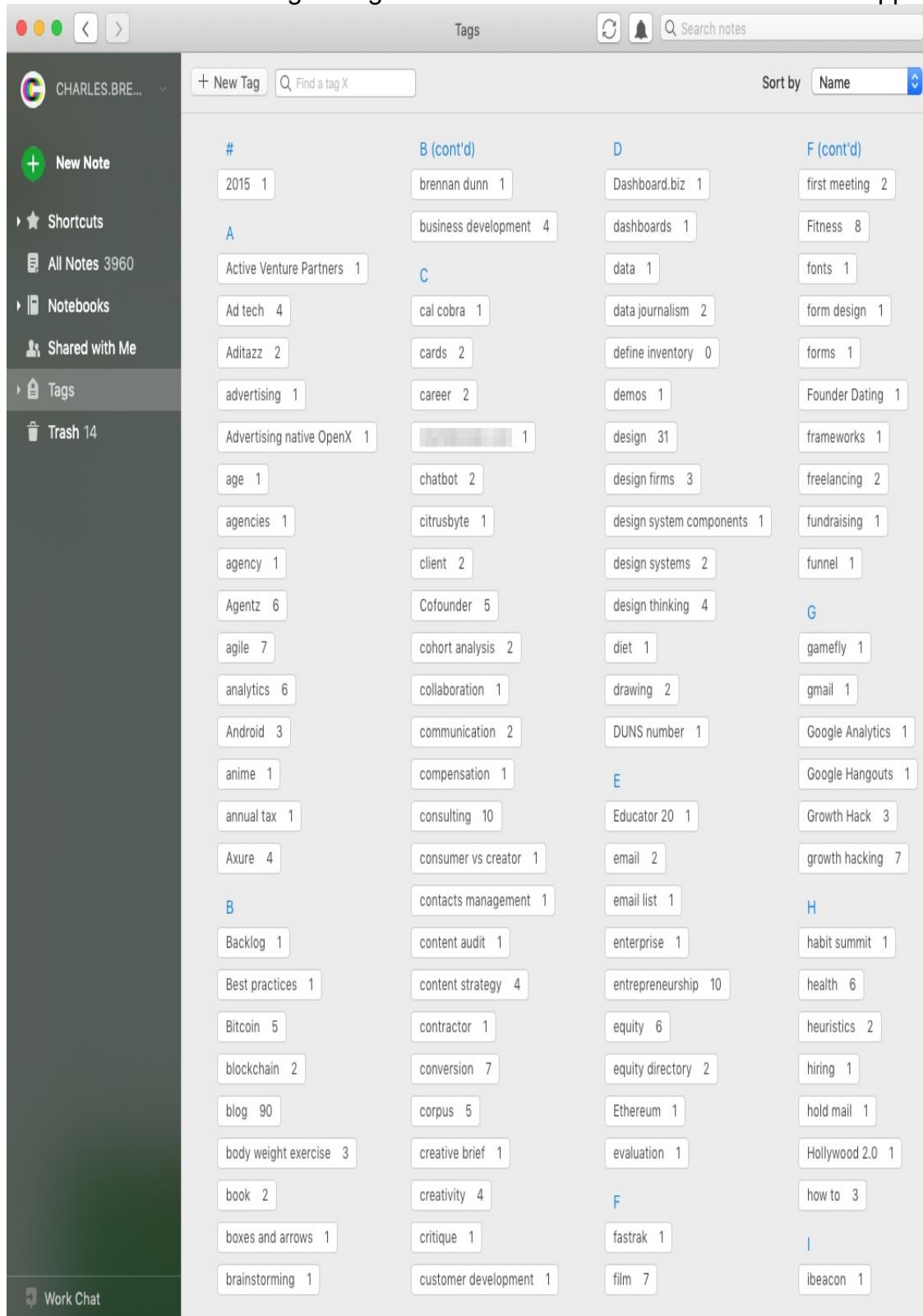


Figure 2-75. Evernote App with tags browser

2. Organizing the Content: Information Architecture and Application Structure
Evernote ([Figure 2-74](#) and [Figure 2-75](#)) is a personal note-taking and Web archiving application for mobile, Web and desktop. Its users save articles, files, presentations and web pages into a central location: The Evernote platform. It allows users to add their own tags when “clipping” a web page to Evernote. This helps with categorizing and finding similar saved articles later. Inside the Evernote app itself, the user can search by tag. There is also a tag index screen, where the user can see what tags exist, and how many articles or clippings are tagged with each term. Selecting a given tag creates a search results list of clippings with that tag. In addition to free-text searching, Evernote’s user-generated tag system creates a powerful method for categorizing and managing large amounts of disparate media and information.

[1](#) “Rich Internet Screen Design,” in UX Magazine: <http://www.uxmag.com/design/rich-internet-application-screen-design>.