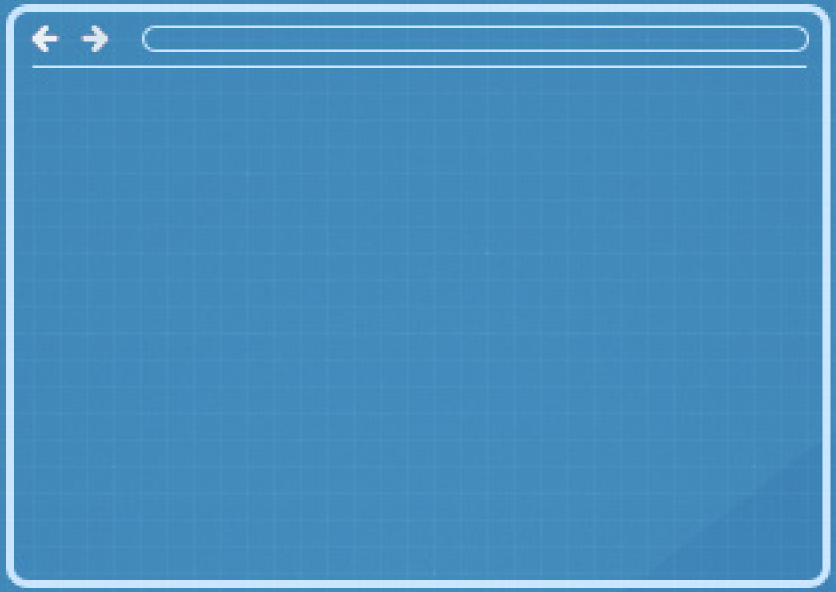


DESIGNING WEB APPLICATIONS

BY NATHAN BARRY

SAMPLE SECTIONS

Included are several sample sections from Designing Web Applications. They are pulled from different parts of the book. I hope you find them useful!

The next page is the table of contents for the full book. Followed by a few excerpts.

TABLE OF CONTENTS

SAVING

Whew, this is a hot topic. Get a few engineers and designers in a room and you can have a fiery debate about when to save data in your web application. I know I have. At a previous software startup we argued about this every time it came up and I still don't think we got it right everywhere. On the surface it is a problem of not wanting the user to lose data. Ever. That should be our top priority.

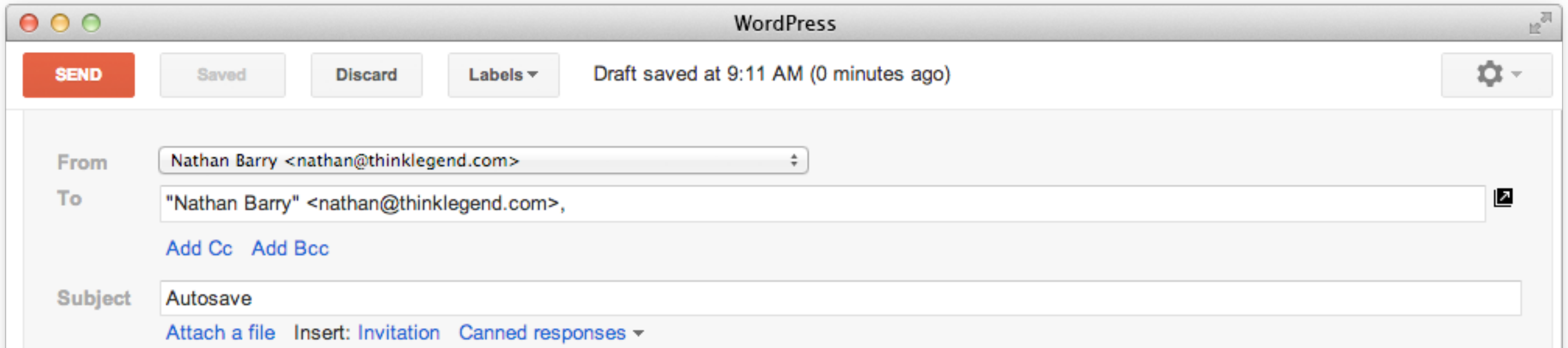
But remember that we are designing not just based on features, but also for emotion. We want the user to feel that their data is safe and that our software will take care of them. So the last thing we want is any confusion as to whether or not their data is saved. Imagine you've just filled out a detailed form on a page and are ready to close the browser. You think the page has autosaved, since there isn't a save button, but you aren't sure. Not knowing for sure creates anxiety. Anxiety that we want to avoid. Maybe it was nice that the software has been saving all along, so their data is perfectly protected. But they don't know that. Keeping the user informed should be a top priority.

So in the times that I use autosave I like to have an explicit save button as well. That way the user can always click that to be certain that they won't lose their hard work.

The Solution

Assuming we've decided a page should have autosave, what is the best way to implement it? A complete solution requires three elements. Gmail does this perfectly, so I'll use them as an example.

First you must have an explicit save button. This should let the user save anytime there are unsaved changes. Second, the save button should be disabled if there aren't any changes to save and become clickable as soon as an unsaved change has been made. This gives the user confidence that their changes are saved. Finally, you need a timestamp. Gmail uses the simple text, "Draft saved at 8:15 AM (3 minutes ago)." Perfect.



Draft saved at 9:11 AM (0 minutes ago)

From: Nathan Barry <nathan@thinklegend.com>
To: "Nathan Barry" <nathan@thinklegend.com>,

Subject: Autosave

In case my computer or browser crashes I have the benefits of autosave, but I also have complete confidence in my own ability to save data. This is a classic case of over-communicating with the user to give them confidence. Remember, it's about how we make them feel.

When is autosave not needed?

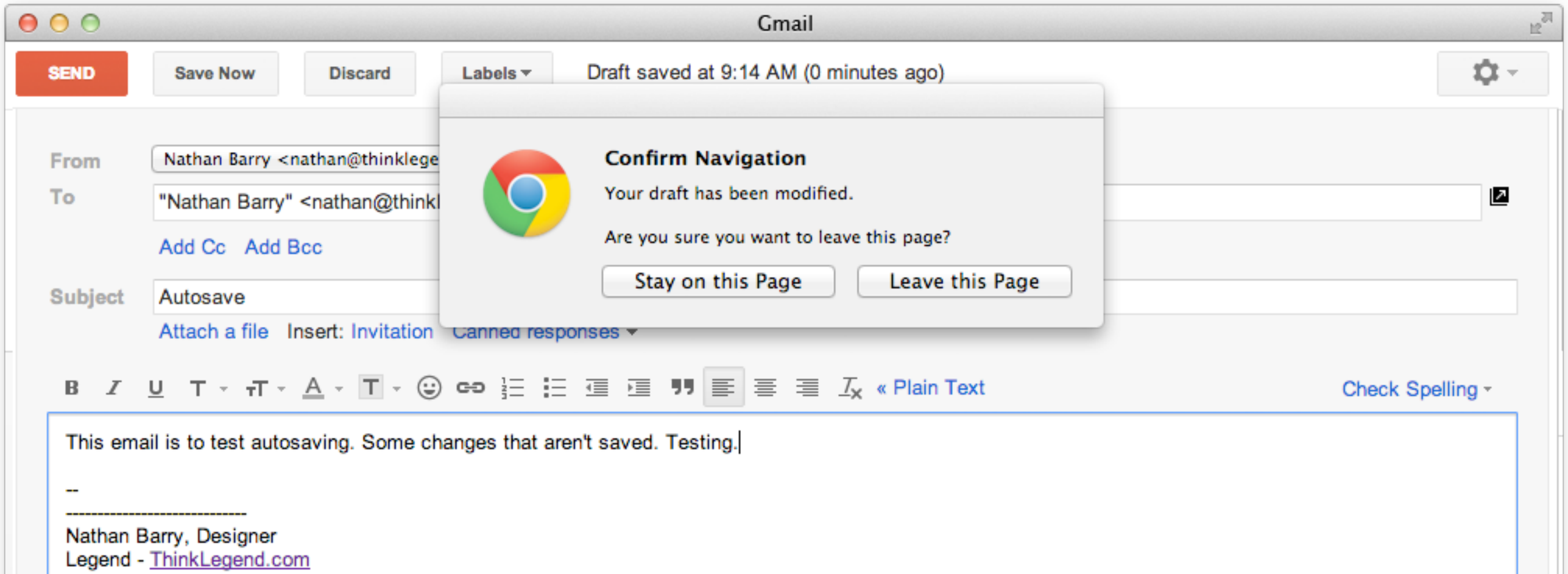
This doesn't mean that autosave is needed on every form. It is certainly not needed for short forms that won't have much data added. As a general rule, if it takes less than 2 minutes to recreate the data, then autosave is not needed.

Everyone is accustomed to using a save button to save the page. So if the risk of data loss is low, just stick with a simple save button.

Leaving the Page

Whether you use autosave or a manual save you should check for unsaved changes when the user is about to leave. If they are about to lose changes it is a good idea to let them know. After all, our top priority when designing our save processes is to keep them from losing data.

Gmail just uses a simple alert that says, "Your message has not been sent. Are you sure you want to leave this page?" This runs whether I am closing the tab or navigating away with the URL or back button.



Draft saved at 9:14 AM (0 minutes ago)

Nathan Barry <nathan@thinklege

"Nathan Barry" <nathan@thinkl

Autosave

Confirm Navigation

Your draft has been modified.

Are you sure you want to leave this page?

This email is to test autosaving. Some changes that aren't saved. Testing.

--

Nathan Barry, Designer
Legend - ThinkLegend.com

DESTRUCTIVE ACTIONS

When deleting an object, the easiest thing to do is ask, “Are you sure you want to delete the last year of accounting data?” with simple a simple Yes / No response. The problem with this approach is that we are used to dismissing dialogs like this every time we use the computer. So many actions require dismissing a dialog that we don’t think twice before clicking Yes …and losing months of vital information.

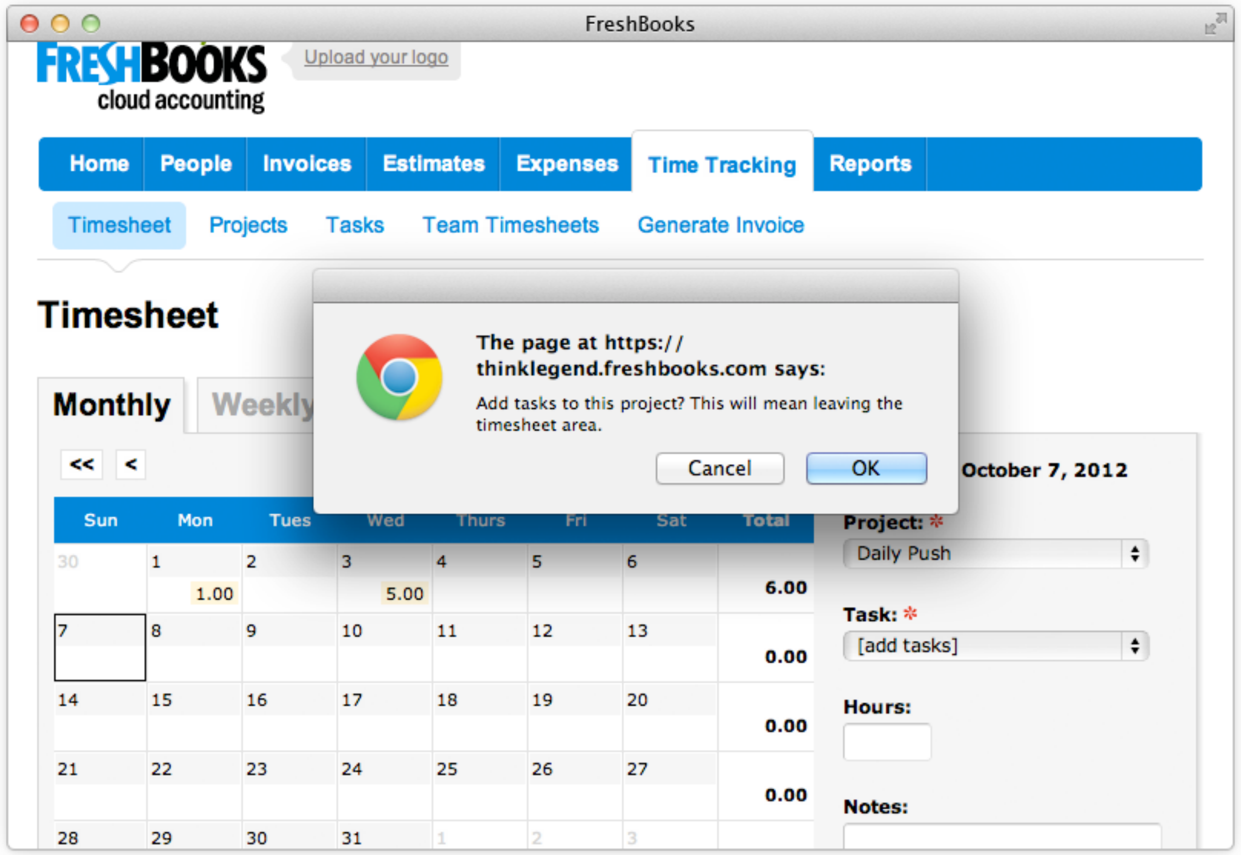
The flip-side is that this may be a common task that the user does quite often. In that case we want to just let them perform the task without constantly interrupting them with dialogs that block their path.

The Solution

The solution is to let them do the action they requested, in this case delete a year’s worth of accounting data, without any warning or dialog. The reason is that the “Oh, ****” moment, where they realize what they’ve done, doesn’t set in until after confirming the action. So that extra dialog doesn’t even register for the user. They often won’t realize the mistake until after it is done.

So let them do it, but on the next screen where you show a notification of what just happened, display an option to undo. “Accounting data deleted successfully. Would you like to undo?”

This would just be a simple inline message with the notification, not requiring an interaction. As software designers we want to trust the user that they actually wanted to do the action they told the software to perform. The accounting software Freshbooks warns the user far too often. If you try to add a task to a project while on the timesheets page, it asks you if you are sure you want to do that.

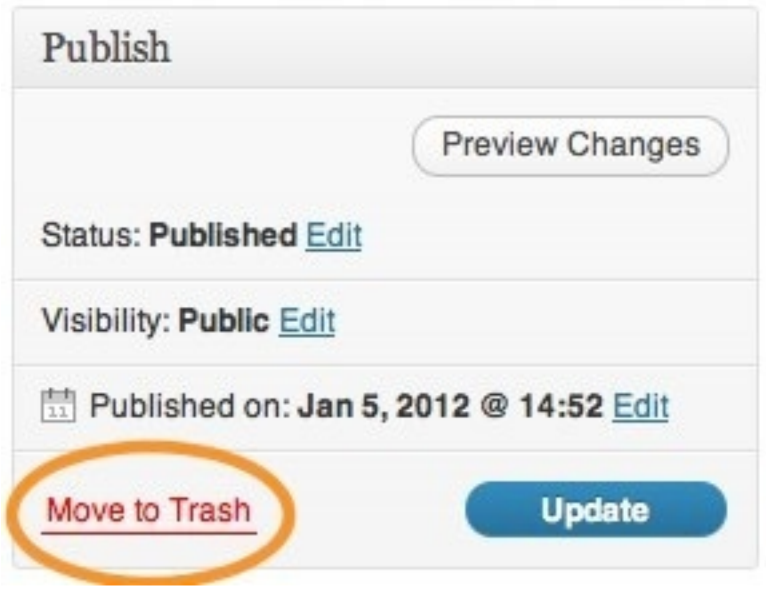

The page at https://thinklegend.freshbooks.com says:

Add tasks to this project? This will mean leaving the timesheet area.

Project: Daily Push

Task: [add tasks]

Since all the information on the timesheets interface is already saved, there is no reason to question the user for such a basic action. That would be like your waiter asking you, “Are you sure?” after every item you say you want to order for dinner. Trust the user and give them an option to undo if they make a serious mistake.


Status: Published

Visibility: Public

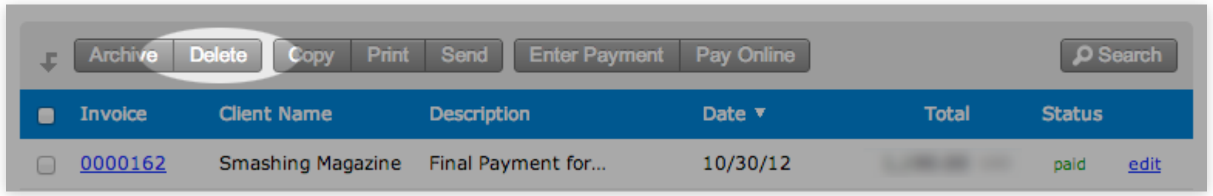
Published on: Jan 5, 2012 @ 14:52

Trash

You can also use the concept of a Trash, like on the desktop, to allow users to undo actions at any time. WordPress added a trash feature so that you can delete blog posts and comments without worry, but always recover them if needed. Many applications never permanently delete information, but instead just hide it from the interface. This is just a way to expose that data back to the user in case they want to recover it (without contacting Support).

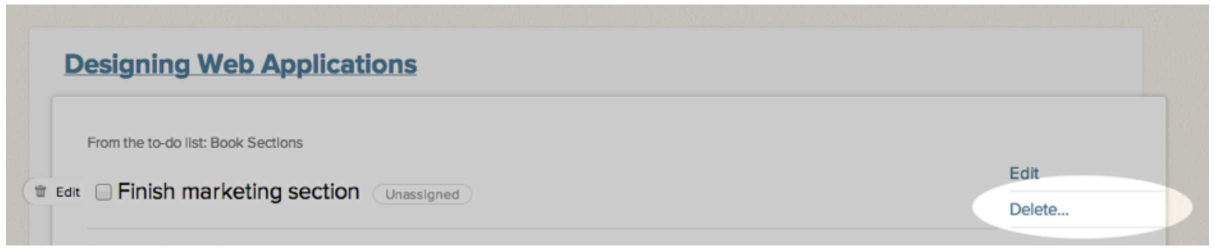
Where to place delete buttons?

I like to place delete buttons in two places: on the list of objects (let's say invoices in Freshbooks), and then again on the details page. Freshbooks has an option to delete invoices from the invoice list, which is good, but they don't have an option when working with a specific invoice.



Invoice	Client Name	Description	Date ▼	Total	Status	
0000162	Smashing Magazine	Final Payment for...	10/30/12	[redacted]	paid	edit

Basecamp, on the other hand, has both. When viewing a to-do list you can mouse over a particular item to see a trash icon. Or when viewing the details of that to-do item a delete button is provided on the right side.


Designing Web Applications
From the to-do list: Book Sections
Edit
Finish marketing section
Unassigned
Edit
Delete...

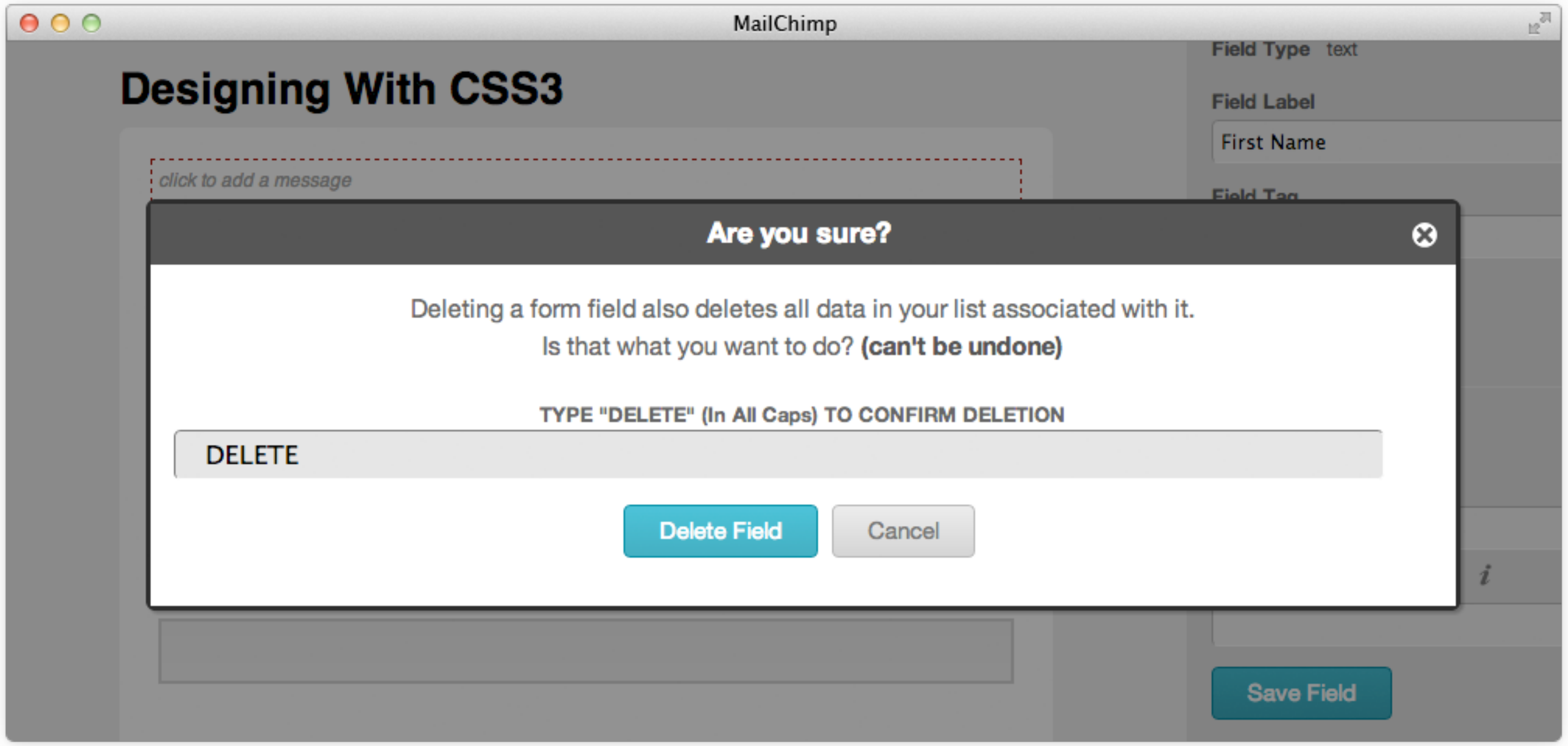
Potentially Destructive

Now there are other actions that may not actually delete information, but could be destructive for the user. Sending email is a great example. You could include the wrong people in an email, send it too early, or forget to include attachments. Many embarrassing emails have been sent through simple mistakes that the sender wishes they could take back. Just like our delete examples above, the mistake won't be realized until after the email is sent. So not only would a confirmation box be annoying to the user every time they want to send an email, it also wouldn't do any good.

Gmail's solution was to include an optional 30-second delay when sending an email (you can enable it in Gmail Labs). If you wanted to send the email, this is such a small delay that it doesn't matter. After all, it doesn't require any extra action from you. But if you made a mistake it gives you enough time to realize the mistake and stop the email before it sends.

In software that sends email notifications it may be good to consider including features such as the 30-second delay. Freshbooks emails invoices and estimates to clients; the user may want to be sure invoices are mistake free before sending.

Think through your software for examples of destructive actions and how you can minimize the damage.


MailChimp

Designing With CSS3

click to add a message

Field Type text

Field Label

First Name

Are you sure?

Deleting a form field also deletes all data in your list associated with it.
Is that what you want to do? **(can't be undone)**

TYPE "DELETE" (In All Caps) TO CONFIRM DELETION

DELETE

Really Destructive

In MailChimp you can delete a field from a signup form for any of your email lists. Not a big deal, right? Except that deleting that field will also remove that column from all of your existing data that you have collected from the form. Ouch! That could be really costly. So they display a dialog that explains this and forces you to type in DELETE (in all caps) in order for it to actually perform the delete.

This is a case where they should warn you and let you undo. But here the extra warning, and forcing the user to read it, can help prevent disaster.

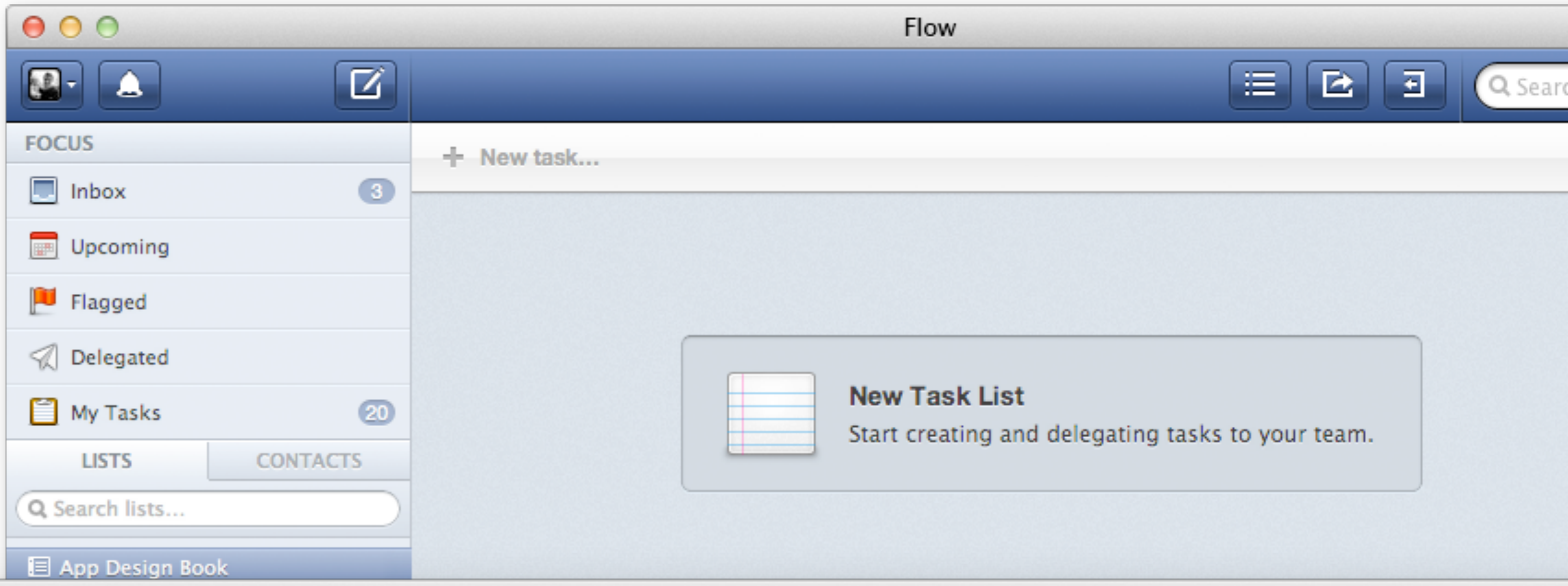
THE BLANK SLATE

The blank slate is what I call a screen with dynamic content, before that content has been created. So the question is, what do you show on the blank slate? Most software just shows a message saying “No Invoices Found” (or something similar) and call it good. But that’s not good enough. Remember that you want to teach the user how to use your software every chance you get, and here you have a perfect opportunity.

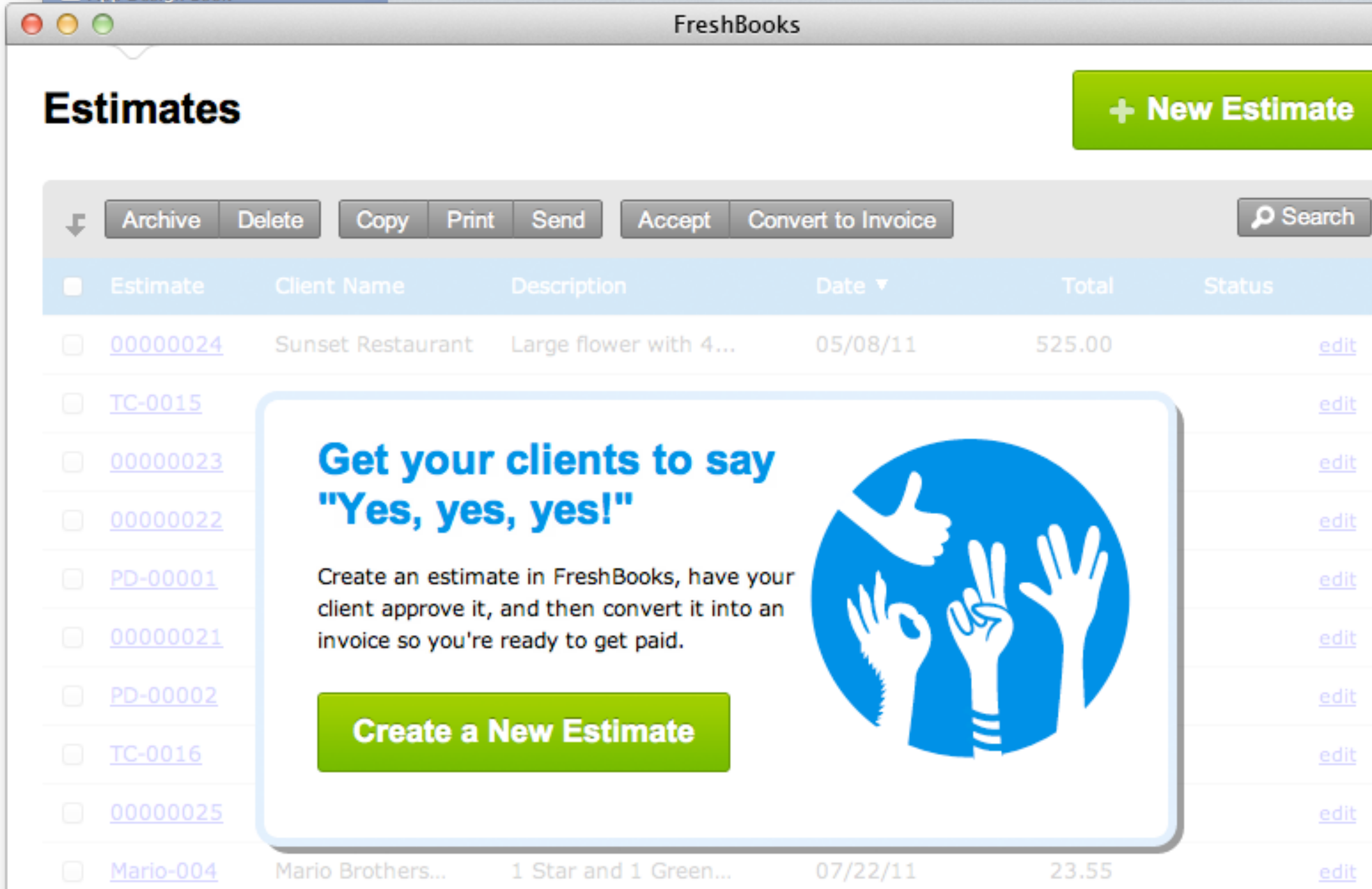
Chances are, since they don’t have any invoices, they haven’t created one before. So we should use that empty space where the invoices will go in the future to teach them about invoices and how to create them.

Flow uses a simple message to introduce the idea of creating a task and to tell you what to do next.

Freshbooks explains what estimates are and how to use them, giving you a button to create an estimate right away.



New Task List
Start creating and delegating tasks to your team.



Estimates

Estimate	Client Name	Description	Date ▼	Total	Status
00000024	Sunset Restaurant	Large flower with 4...	05/08/11	525.00	edit
TC-0015					edit
00000023					edit
00000022					edit
PD-00001					edit
00000021					edit
PD-00002					edit
TC-0016					edit
00000025					edit
Mario-004	Mario Brothers...	1 Star and 1 Green...	07/22/11	23.55	edit

Get your clients to say "Yes, yes, yes!"

Create an estimate in FreshBooks, have your client approve it, and then convert it into an invoice so you're ready to get paid.

Create a New Estimate

What to Include

A blank slate screen should include a few basic elements:

- Say what will go on the page (tasks, invoices, etc).
- Explain any information someone may want to know about that thing if they are encountering it for the first time.
- Show how to create that thing.

Just take that opportunity to teach the user how to use your software better.

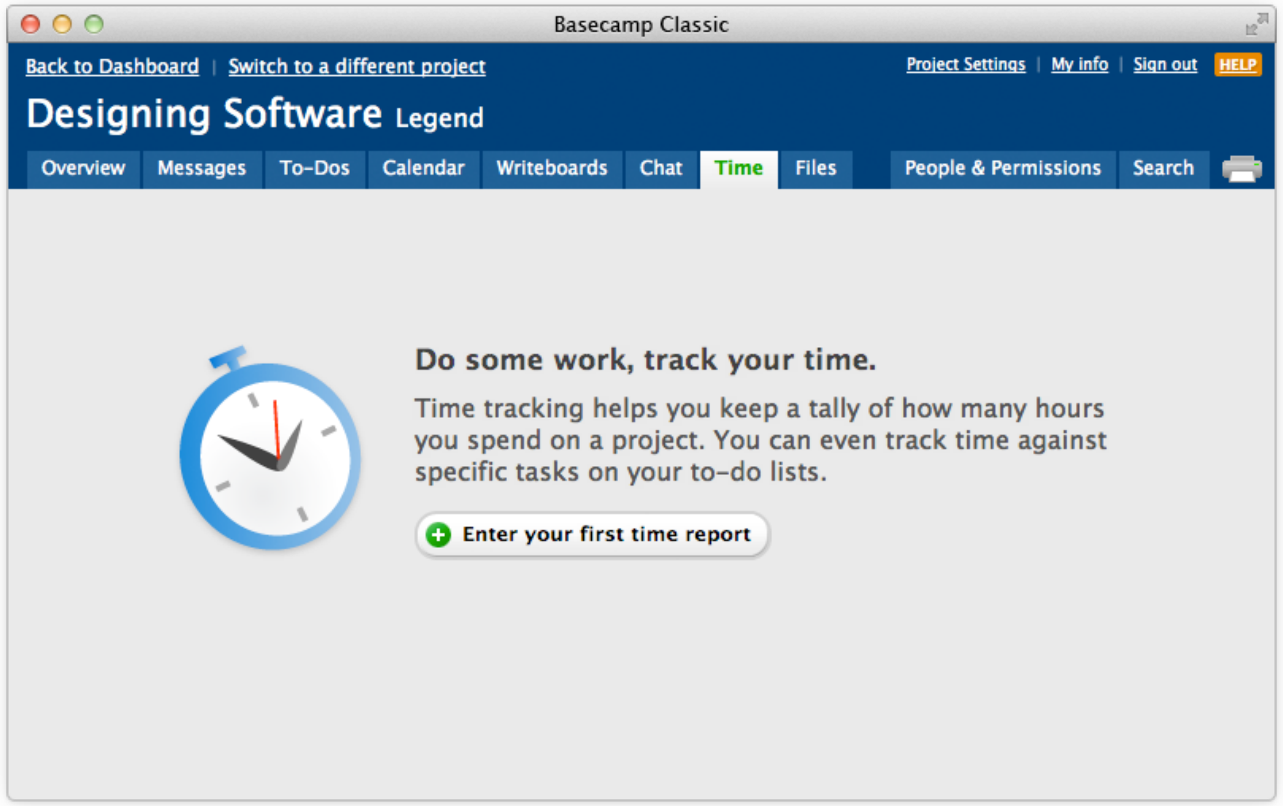
A Common Mistake

A common mistake in your first-run experience is to have a help tutorial that takes the user through the setup process. Like a setup wizard in old desktop software this process creates an entirely new, though step-by-step, UI for getting started. The problem with this is, though you may explain some concepts of your software, this is an interface they will never see again.

The most common example is having a button on your blank slate screen to create a new object. This would be fine, except that is where you are training them to click. As soon as the first object is created, the blank slate disappears, and the method they learned to create with is gone as well.

Instead, it is better to use the blank slate to point to a permanent UI button that will be available every time. This way if the message or tooltip is gone, they are still using the same button as last time.

It is very important that you teach users to use the actual software the first time around. Don't use a pseudo-interface that will be gone after the first run experience is complete. This also means that you should provide pointers and tooltips to the actual functions, not a screenshot of what you want the user to do next. Just remember to teach through practice. Guide them through the step-by-step process of using the actual application.

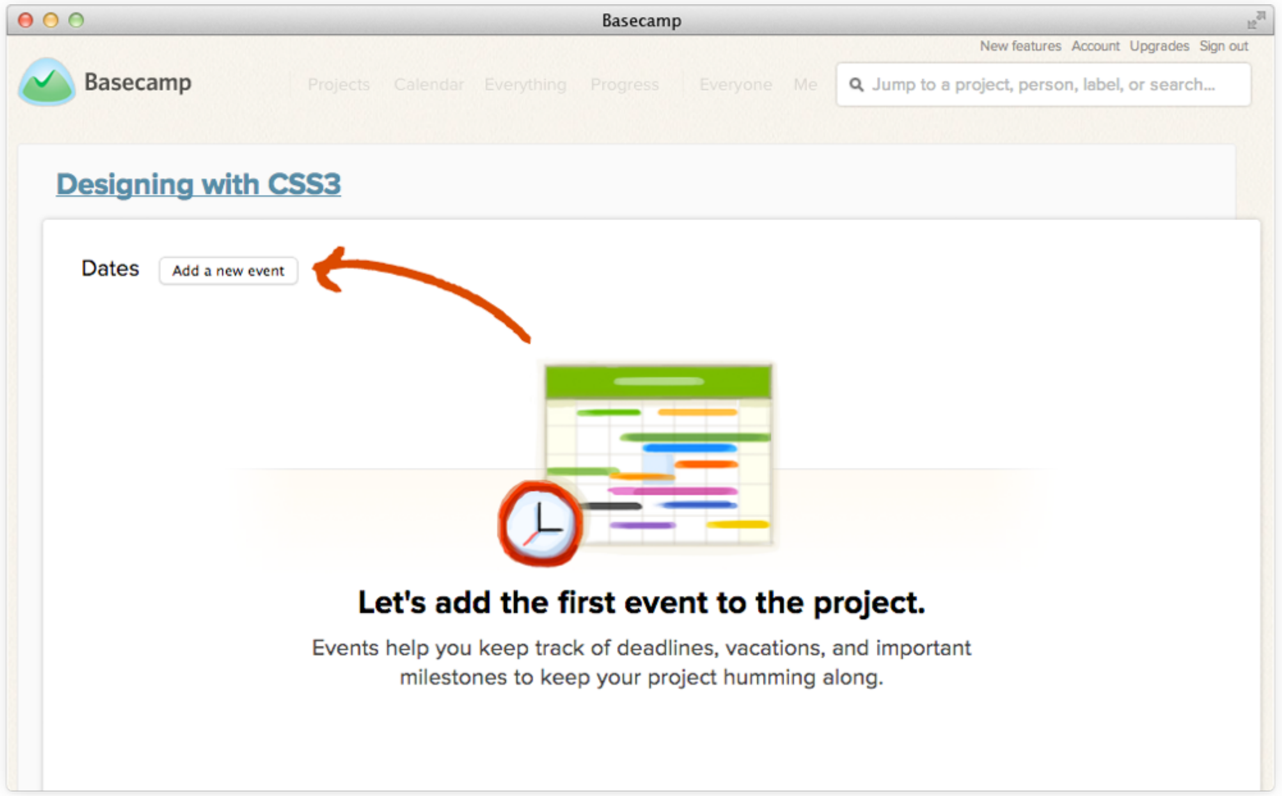

Basecamp Classic
Back to Dashboard | Switch to a different project
Project Settings | My info | Sign out HELP
Designing Software Legend
Overview
Messages
To-Dos
Calendar
Writeboards
Chat
Time
Files
People & Permissions
Search

Do some work, track your time.

Time tracking helps you keep a tally of how many hours you spend on a project. You can even track time against specific tasks on your to-do lists.

Enter your first time report

In Basecamp Classic, the blank slate screens have a button to Add Time (or whatever the action for that page is), but it disappears once the first time is added. This was fixed in the latest version of Basecamp, which actually uses an arrow to point to a constant button from the blank slate message.


Basecamp
New features Account Upgrades Sign out
Basecamp
Projects Calendar Everything Progress Everyone Me
Jump to a project, person, label, or search...
Designing with CSS3
Dates
Add a new event

Let's add the first event to the project.

Events help you keep track of deadlines, vacations, and important milestones to keep your project humming along.

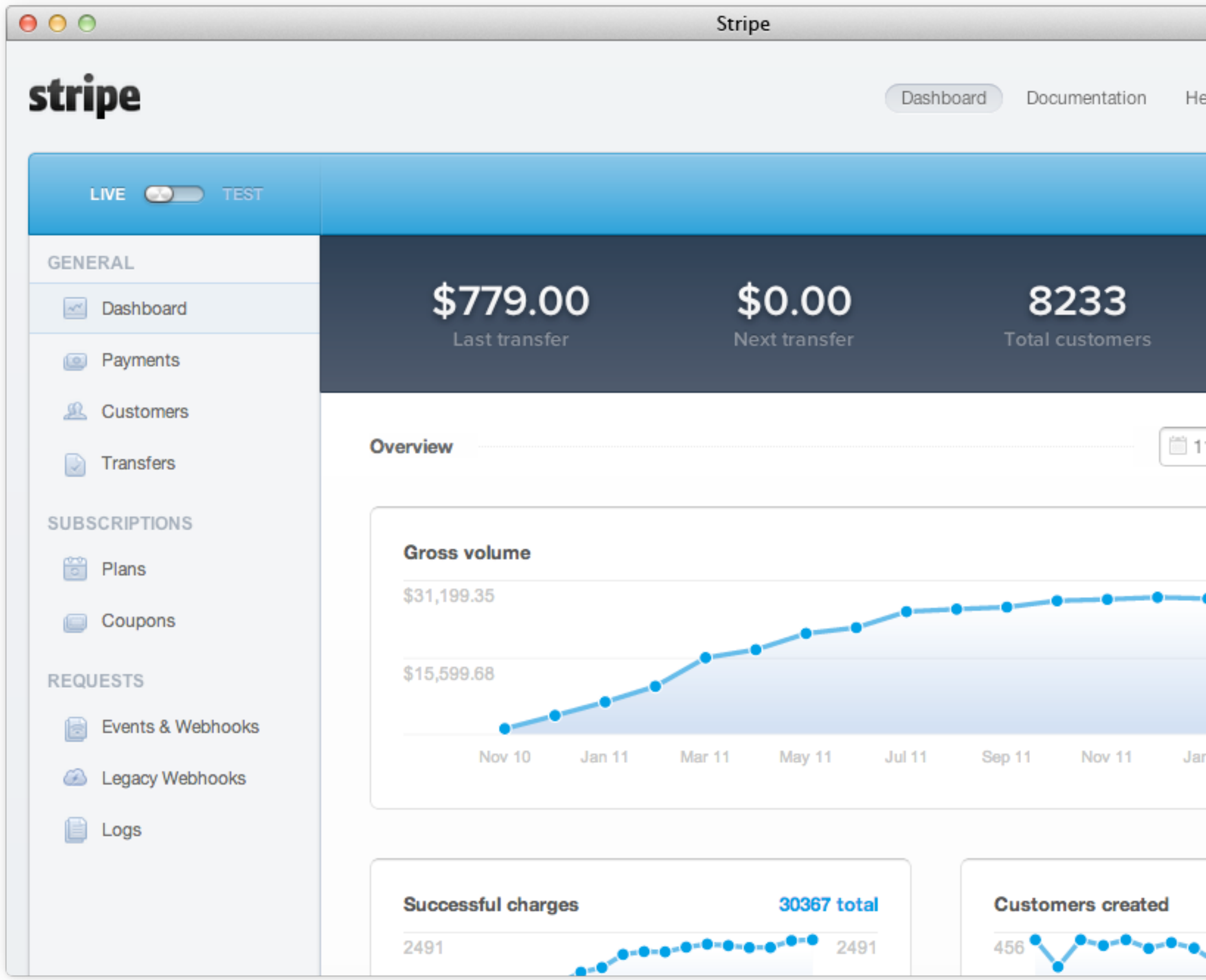
Teach your users one way to use the application, and be consistent.

Isn’t the blank slate just a help screen?

Well, yes and no. It does help the user understand the software, which is really important, but unlike most help, it is low friction and only there on the first run, when the user needs it most. After they know how to interact with that element, it won’t appear again. Basically, it is an easy way to show help only when it is needed.

DESIGNING THE DASHBOARD

The first thing to know is that dashboards are often not necessary. If your application is fairly small or focused around a single task then you may be better off taking the user to a task page. In our sample time-tracking application I decided a dashboard is not needed and made the projects page the default page after the user signs in. So if you are short on time and need to cut features, a dashboard is a great place to start.



The place for a dashboard

Just because a dashboard shouldn't be at the top of your development priorities, doesn't mean there isn't a place for a well designed dashboard. For some apps that include analytics or sales information a dashboard is critical. Just think about an application like Stripe, which processes credit cards. When I sign in to Stripe the first thing I want to know is how many sales I've made. Next I want to know how that trends over time with a graph.

This is where a dashboard is perfect. The key metrics can be displayed large and bold, along with graphs to give historical information. Compare this to the experience of being dropped on the Payments page. Sure, maybe there would be a total, but it would be a long list of recent payments and would not be great for scanning quickly.

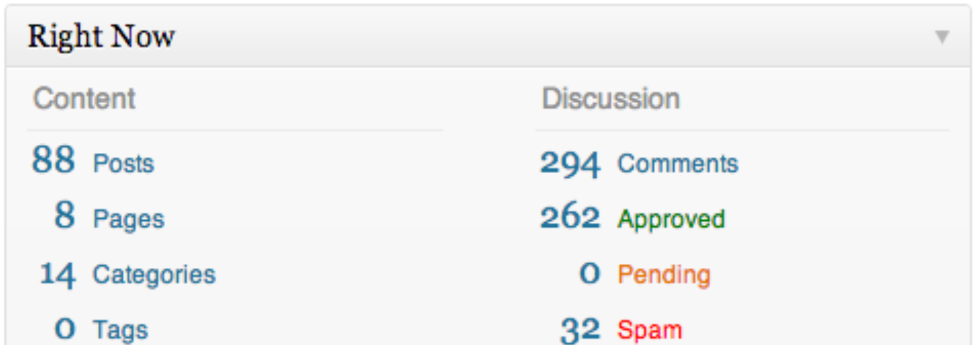
If there are any metrics that prove your software is being successful for their business, consider including them.

“Be clear first and clever second. If you have to throw one of those out, throw out clever.”

-Jason Fried, 37signals

A springboard

Even a blogging software that may seem relatively simple can have a lot going on. For the WordPress dashboard they pull in your draft posts, recent comments, stats, and even let you write a post. This is great for pulling all the most important information about your entire web app into a single place. Not only does it give you information, but each of these widgets has a link to more details. For example, the Recent Drafts widget has a “View All” link so that you can jump straight to all your draft posts. At the same time the Right Now widget lets you click on the post or comment counts to be taken straight to those pages.



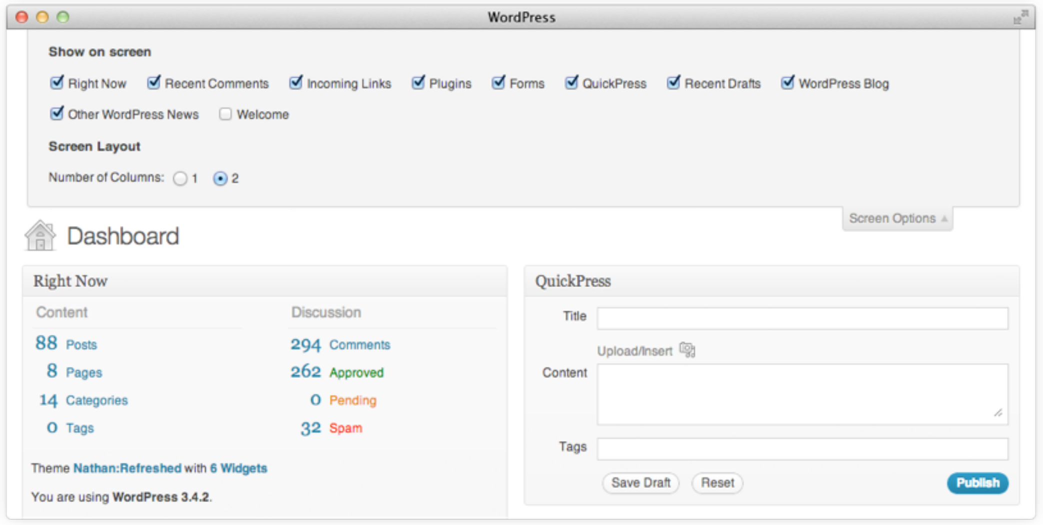
Right Now

Content		Discussion	
88	Posts	294	Comments
8	Pages	262	Approved
14	Categories	0	Pending
0	Tags	32	Spam

Don’t just display static data. Let the user interact with it and use the dashboard to jump off to different parts of your application.

Customizable Widgets

If you have as much information to include as WordPress does, you should probably let the user customize it. On the WordPress dashboard you can show or hide any widget, customize certain settings, and change the order to prioritize what you care about. This is not a requirement for your dashboard, but as your software is used by a broader range of people it may be more helpful to let it be personalized.



WordPress

Show on screen

✓ Right Now ✓ Recent Comments ✓ Incoming Links ✓ Plugins ✓ Forms ✓ QuickPress ✓ Recent Drafts ✓ WordPress Blog
✓ Other WordPress News ☐ Welcome

Screen Layout

Number of Columns: 1 [✓ 2]

Screen Options

Dashboard

Right Now

Content		Discussion	
88	Posts	294	Comments
8	Pages	262	Approved
14	Categories	0	Pending
0	Tags	32	Spam

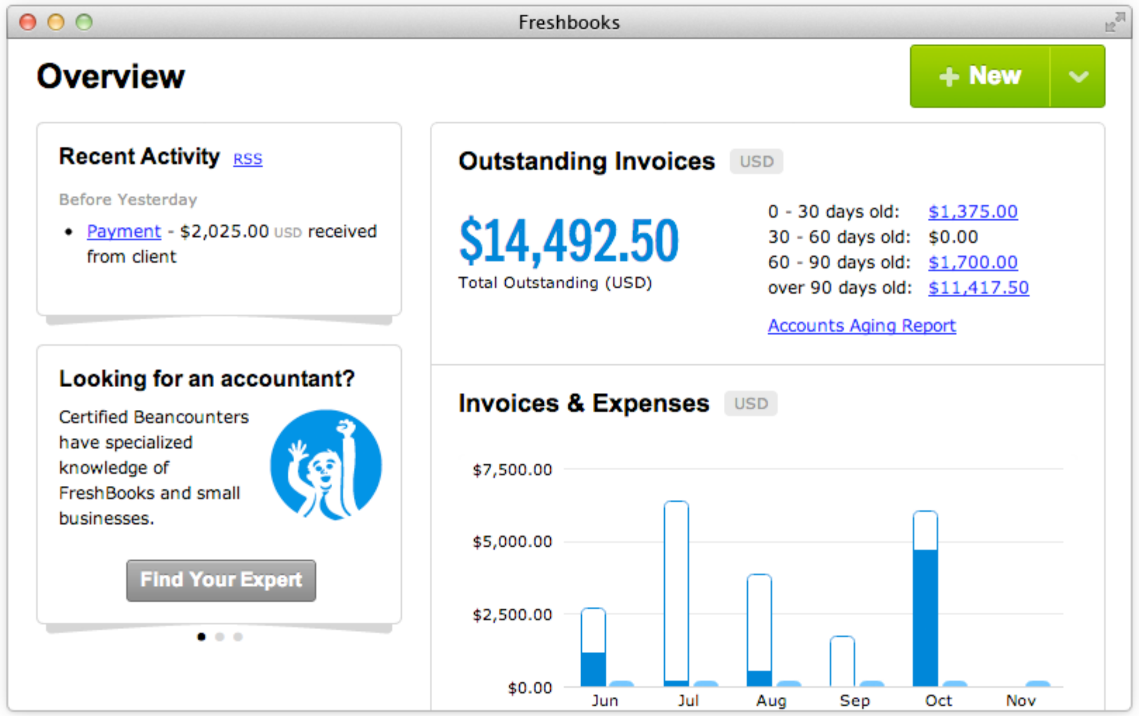
Theme Nathan:Refreshed with 6 Widgets

You are using WordPress 3.4.2.

QuickPress

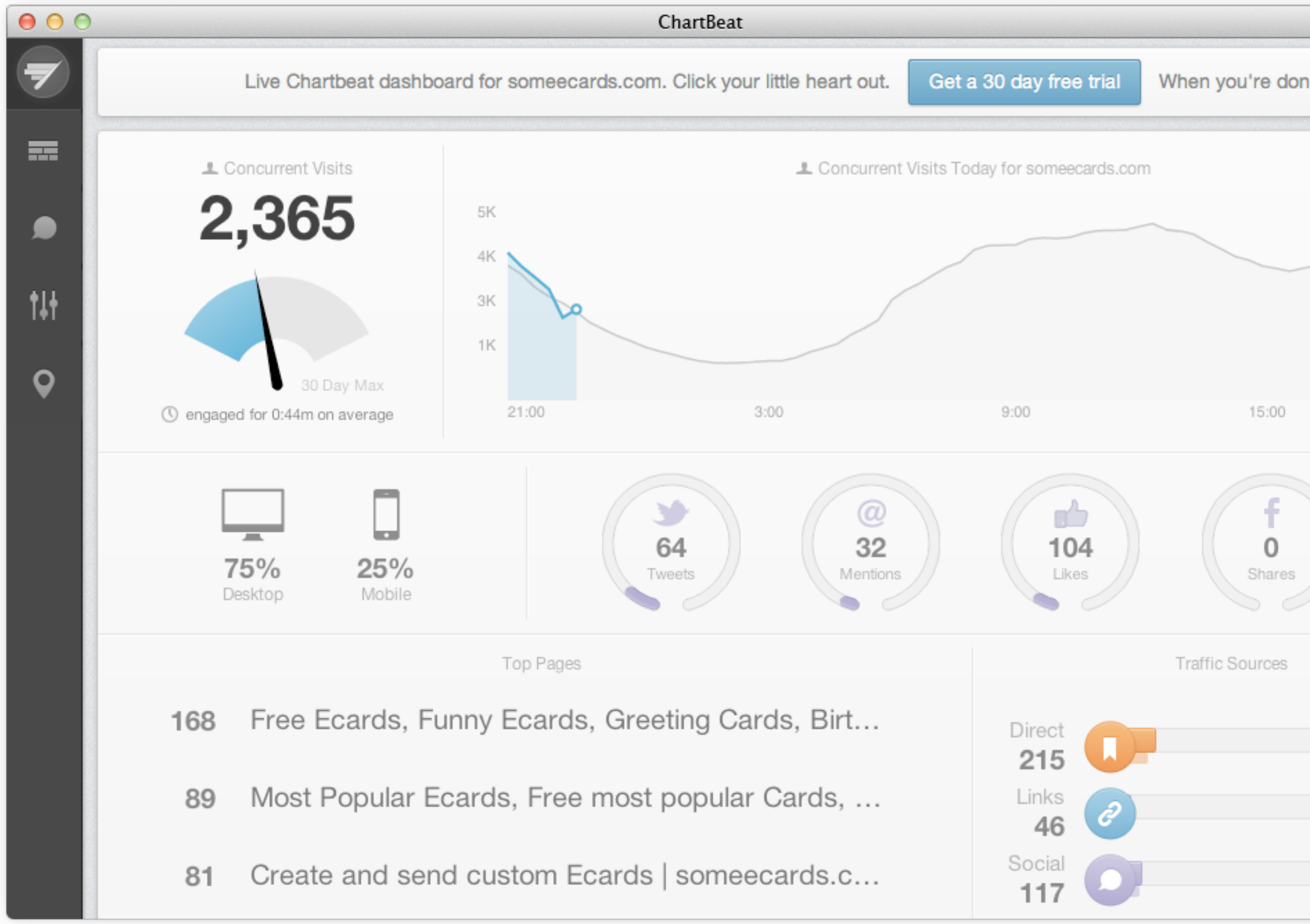
WordPress even lets you change the number of columns and drag widgets between columns to get the right data density for your site.

Freshbooks lets you see your outstanding invoices in a nice big number right away, always a reminder that I need to get better at collecting on invoices. Though they have two columns, the Recent Activity portion is fixed, but the sections in the right column can be reordered.



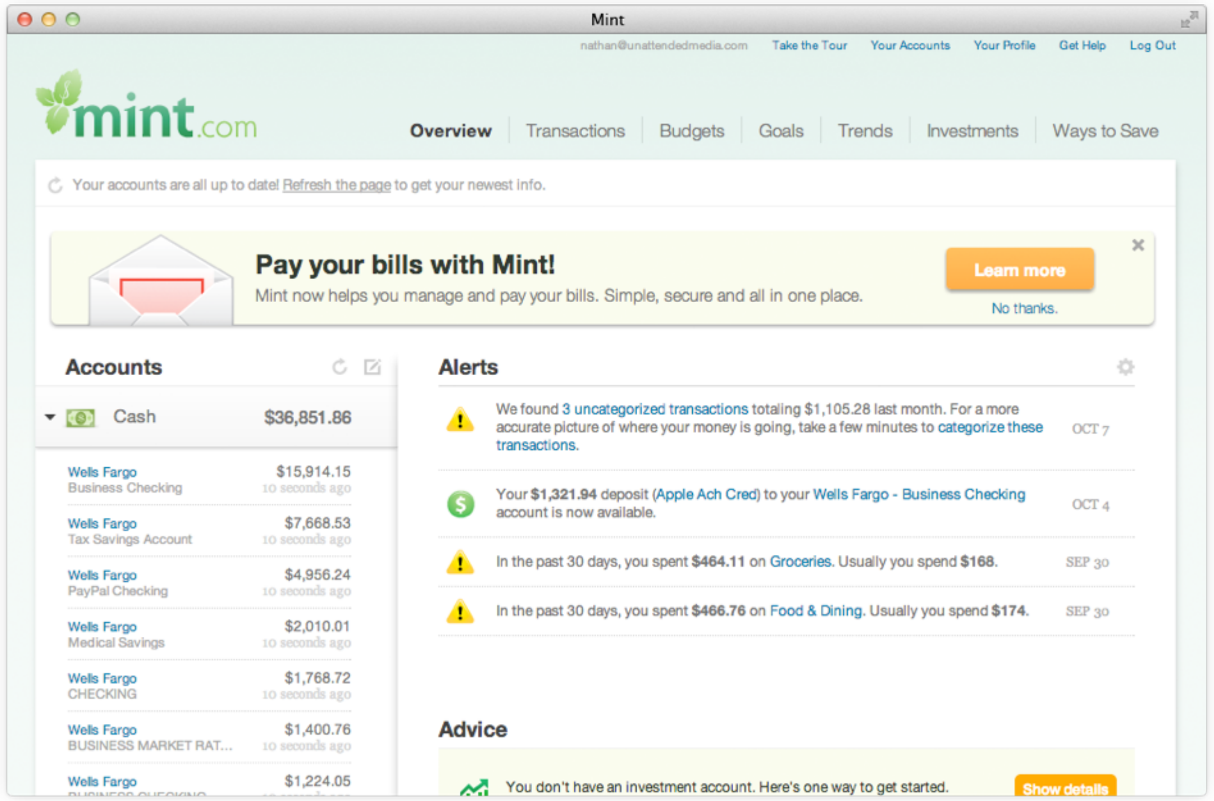
Impressive Visuals

Usually a dashboard will be the user's first introduction to your product, so it helps if you make it look great. You can use graphs and other data visualization to add some great visuals to the dashboard. Icons, large numbers, and colors (within reason) help as well. Take a look at ChartBeat for an example of a beautiful dashboard.


ChartBeat
Live Chartbeat dashboard for someecards.com. Click your little heart out.
Get a 30 day free trial
When you're don
Concurrent Visits
2,365
30 Day Max
engaged for 0:44m on average
Concurrent Visits Today for someecards.com
5K
4K
3K
1K
21:00
3:00
9:00
15:00
75% Desktop
25% Mobile
64 Tweets
32 Mentions
104 Likes
0 Shares
Top Pages
168 Free Ecards, Funny Ecards, Greeting Cards, Birt…
89 Most Popular Ecards, Free most popular Cards, …
81 Create and send custom Ecards | someecards.c…
Traffic Sources
Direct 215
Links 46
Social 117

Naming

“Dashboard” may feel too formal and stuffy for your application. Instead, consider using a different name like “Overview,” if it is a better match. The financial management app Mint calls their summary page Overview since that feels more accurate for their service. It still works great; right after sign in I can see my cash balance, account alerts, and upcoming bills at a glance. If I scroll further down I can see goals, trends, and other useful information. It really is an overview of my finances.



Your accounts are all up to date! Refresh the page to get your newest info.

Pay your bills with Mint!

Mint now helps you manage and pay your bills. Simple, secure and all in one place.

Accounts

Cash	$36,851.86
Wells Fargo Business Checking	$15,914.15 10 seconds ago
Wells Fargo Tax Savings Account	$7,668.53 10 seconds ago
Wells Fargo PayPal Checking	$4,956.24 10 seconds ago
Wells Fargo Medical Savings	$2,010.01 10 seconds ago
Wells Fargo CHECKING	$1,768.72 10 seconds ago
Wells Fargo BUSINESS MARKET RAT...	$1,400.76 10 seconds ago
Wells Fargo	$1,224.05

Alerts

We found 3 uncategorized transactions totaling $1,105.28 last month. For a more accurate picture of where your money is going, take a few minutes to categorize these transactions.	OCT 7
Your **$1,321.94** deposit (Apple Ach Cred) to your Wells Fargo - Business Checking account is now available.	OCT 4
In the past 30 days, you spent **$464.11** on Groceries. Usually you spend **$168**.	SEP 30
In the past 30 days, you spent **$466.76** on Food & Dining. Usually you spend **$174**.	SEP 30

Advice

You don't have an investment account. Here's one way to get started.

Final Thoughts

Always keep in mind that this dashboard is for your users. Not you. So focus on what they need, not what you think would be cool. It may be a good idea to build your own internal dashboard for traffic, support, and sales information.

The best thing to do with a dashboard is make your best guess, then iterate quickly based on testing and feedback. This is true for everything in software design, but especially true for dashboard design.

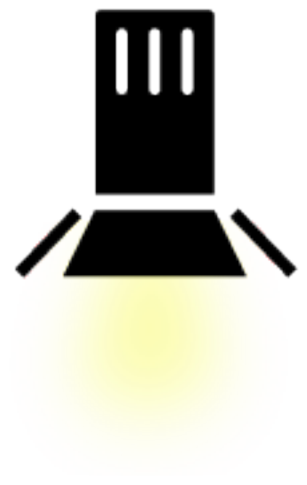
For more information on designing dashboards read Data Visualization in Web Apps by Des Traynor.

LIGHTING & GLOSS

Without a light source our designs would look flat and boring. Shadows, highlights, and depth wouldn’t exist. These are all things that bring our buttons and other elements to life and make them look like you can interact with them. Proper lighting is crucial to a great interface.

A Light Source

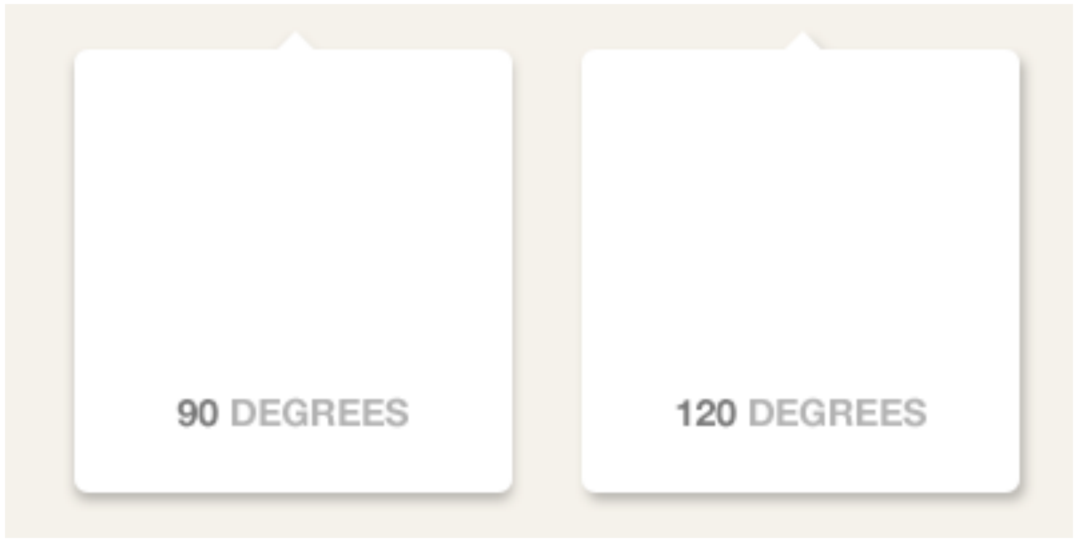
Remember learning to draw in grade school? Someone may have told you to draw a small sun in one corner of your image as a reminder where the light source was. Think about it in the same way for your application. It doesn’t matter as much where you choose to place it (either top left, top, or top right), but that you are consistent.



Personally, I prefer the light source in my designs to be 90 degrees. That is straight down from the top. It gives things a nice uniform look that I like, though probably it is a habit I brought over from designing for the iPhone (where the light source is always 90 degrees).

The Implications

Once you have a light source, it is time to think about what that means for each of your elements. Say our light source is 90 degrees. That means all the highlights need to be on the top and the shadows need to be along the bottom. The left and right edges of your elements should be mostly free of shadows and highlights. One reason I prefer this method is that drop-down menus and other floating elements will have a shadow on both the left and right side which looks better to me.

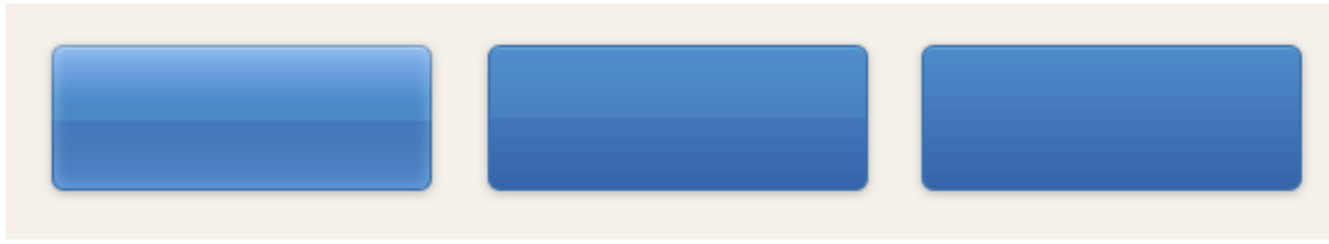


If instead you have an angle of 120 degrees (coming down from the top left), which is the Photoshop default, then your highlights will be on the top and left sides and the shadows will be on the bottom and right sides of each element. Again, it doesn't matter so much what you choose, so long as you are consistent. This is where the global light feature in Photoshop can be helpful, as it will force your light direction to be the same on all elements.

Continue to apply this to text shadows and any other elements where you are using lighting to give the appearance of depth.

Gloss

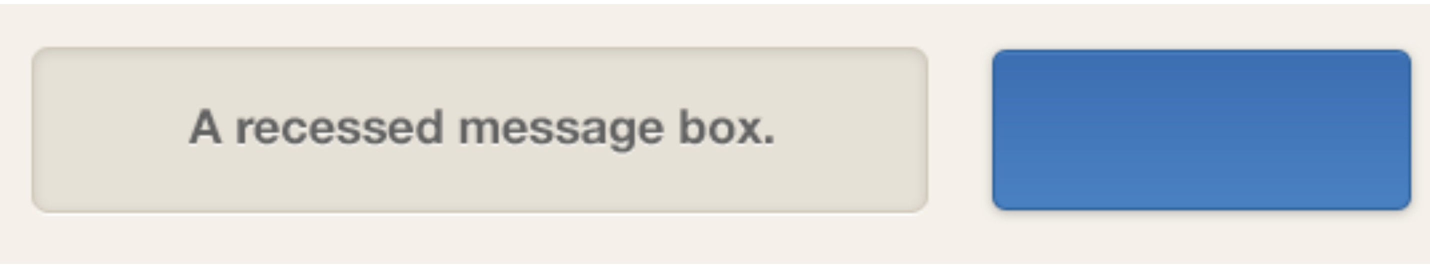
Really, gloss is a texture. Unlike a repeating wood pattern or linen texture, it is not obvious, but it still describes the texture of the element. We know that a button with gloss on it is smooth and shiny. The shape of gloss depends on each element. A heavily rounded button will have an entirely different reflection than something that is flat.



Generally, you want to keep gloss subtle. If used, it should add a little bit of detail to your element, but not scream for attention. Back in 2004 to 2006 heavy gloss buttons were in style; now they are just a great way to make your site look outdated.

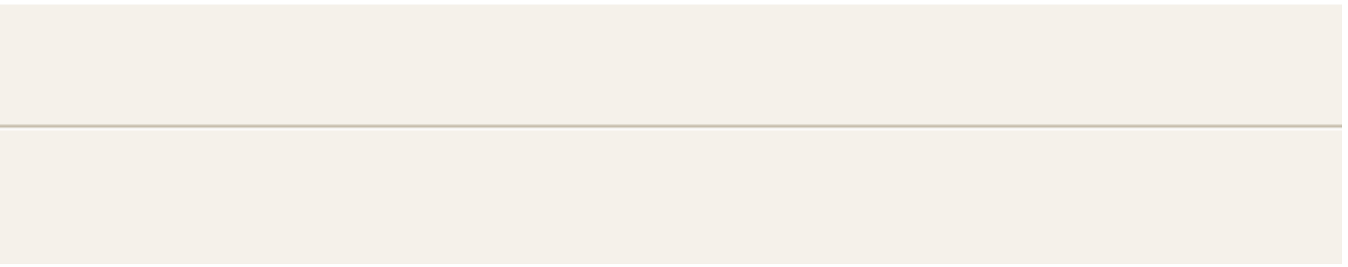
A Recessed Effect

To make a button look recessed, you just need to flip the highlights and shadows. Now the highlight would be on the bottom, the shadow on the top. This is helpful for the pressed or active states on buttons. Also, an inner shadow will help enhance the effect.



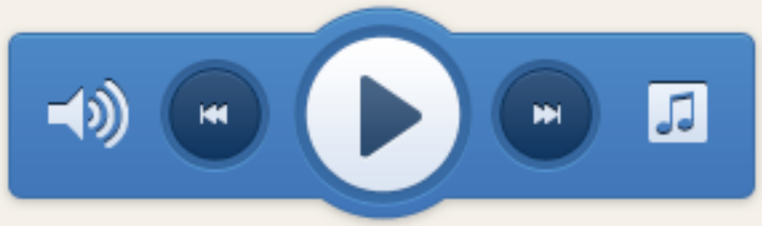
One Pixel Highlights & Shadows

I love really crisp looking interfaces. Where the details of each line and highlight really stand out. This effect is created with lots of 1 pixel highlights and shadows. A basic example is a recessed dividing line. It is just made up of two lines. Dark on top, light on the bottom. The highlight on the bottom edge changes it from a flat, dark line, to something that actually looks indented in the page.



These simple one pixel lines can be used in so many places. But there are a few basic rules to follow.

- To make a light colored element look recessed on a darker background it needs a 1 pixel *dark shadow along the top edge*. So that would be a Y position of -1.



- To make a dark element look recessed on a light background it needs a 1 pixel *highlight on the bottom edge* (a Y position of 1).

Those two rules apply to text, buttons, message boxes, icons, and just about anything else. You can see them both in use if you look closely at the audio player design above. Look for which lines are dark and which are light.

These highlights and shadows will make your design look more realistic, but if you take it too far, then the effect is just tacky.

ENJOYED THE SAMPLE?

Purchase Designing Web Applications when it is available on December 12th, 2012.

Learn more about the book »