

Step by Step UI Design

Designing a User Interface From Scratch



ABOUT THE AUTHOR

Sacha Greif

I'm a freelance designer and entrepreneur from Paris, France. I've worked with startups such as [Hipmunk](#), [MileWise](#), [Codecademy](#), and [Sharypic](#).

You can read my blog at [SachaGreif.com](#) or [follow me on Twitter](#), and check out my startup, [Folyo](#).

Table of Contents

Introduction	4
What We're Building	6
What About Wireframes & Sketching?.....	8
Step 1: Rulers.....	9
Step 2: Main Area & User List.....	10
Step 3: The Text Field.....	12
Step 4: Conversation	14
Step 5: Conversation, Again.....	16
Step 6: More Conversation	18
Step 7: Sidebar.....	20
Step 8: Tabs	21
Step 9: Color Tweaks.....	22
Step 10: More Buttons & Search.....	24
Step 11: Sidebar Improvements	25
Step 12: Sidebar Separator	27
Step 13: Collapse Control.....	28
Step 14: Collapsed Sidebar	30
Step 15: Hover Widgets	31
Step 16: Music Player	33
Step 17: Cleaning Up.....	36

Step 18: Add Media Overlay.....	37
Step 19: Search	38
Step 20: Conclusion.....	40
Step 21: Oops!.....	41
Appendix: Design Resources.....	42

Introduction

When does design become art? Do you need to have artistic talent to become a good designer? And in fact, how do you even *define* good design?

Those are all questions that I have no intention of answering in this short tutorial.

Instead, what I intend to do is very simple: **design a user interface from scratch.**

User interface design is a very practical and utilitarian subset of design. If the word “design” evokes images of stylish yet comfortable office chairs that retail for thousands of dollars, UI design would be more like finding a way to build subway benches that homeless people can’t sleep on.

So if like me you can’t draw a straight line and hate art museums, don’t despair: you can still be a great UI designer. You just need to know a couple tricks and remember some basic principles.

This will also be a little different from most UI tutorials out there, as I won’t try to teach you things directly so much as give you some insight into my thought process.

I will also focus more on design and less on actual implementation, so you won't find too many Photoshop or CSS tips.

And to help you go further, I will also include my favorite design resources and articles in the Appendix.



Design Principles

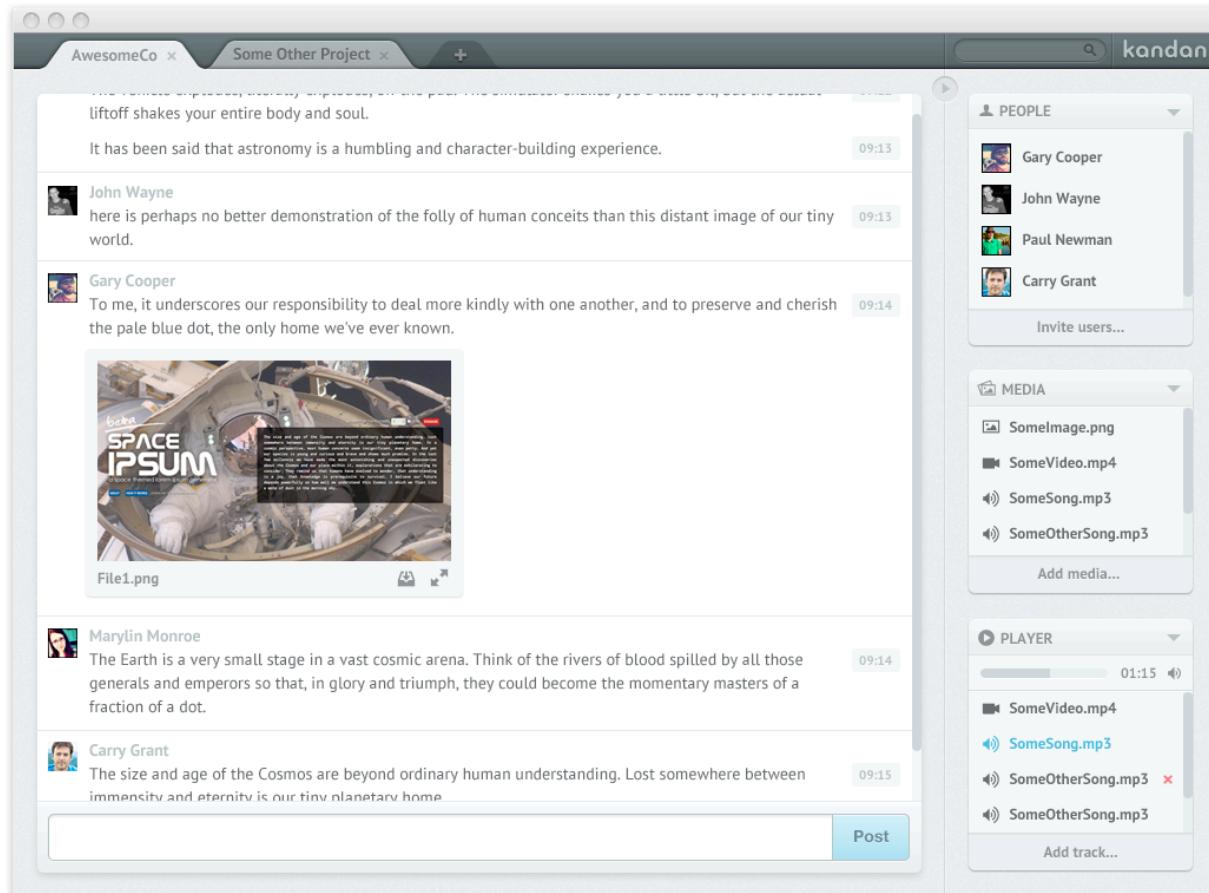
I'll also use these fancy yellow boxes to highlight general principles that can be applied to your own designs as I go along.

These "principles" can be high-level concepts like hierarchy or unity, or very concrete elements like type or icons.

But enough talk, let's get started!

What We're Building

We're going to work on **Kandan**, a modern open-source chat app that lets teams create their own private chatrooms. First, let's take a look at the final result:



Looks complex, but don't worry, we'll get there step by step.
Let's start by outlining the app's main features:

- View the conversation
- Participate in the conversation

- Support widgets in the sidebar, including a list of online users

On top of these basic features, our chat client has a couple aces up its sleeve:

- Switch between multiple chat rooms using tabs
- Search across all chat rooms history
- Upload media into a conversation and display it



Hierarchy

It's a good habit to prioritize features into "must-haves" and "nice-to-haves".

This way, you always know which features to put forward in the UI, and can institute a clear hierarchy throughout your design. A strong hierarchy is one of the most important characteristics of good user interfaces.

What About Wireframes & Sketching?

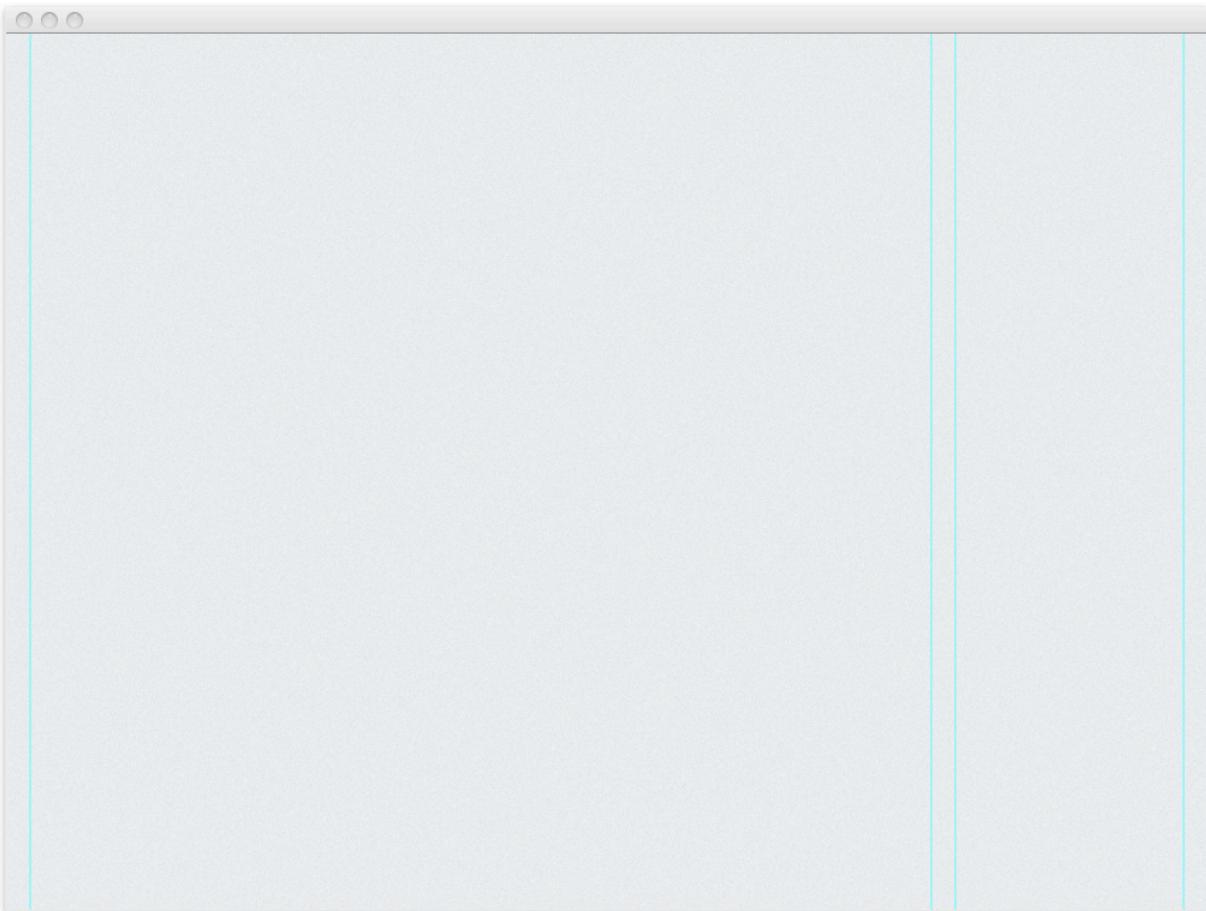
A common workflow for UI design is to start off with sketching, then move on to low-fidelity wireframes (with an app like **Balsamiq** for example), then open Photoshop to produce high-quality mockups, and eventually code a prototype in HTML/CSS.

While there's nothing wrong with this in itself, I don't like getting fixated on a specific workflow.

Some projects will lend themselves to designing in the browser straight away, while others will require extensive wireframing first.

Today we're using a pretty standard layout, so I'll skip wireframing and sketching and go straight to the high-fidelity mockup phase.

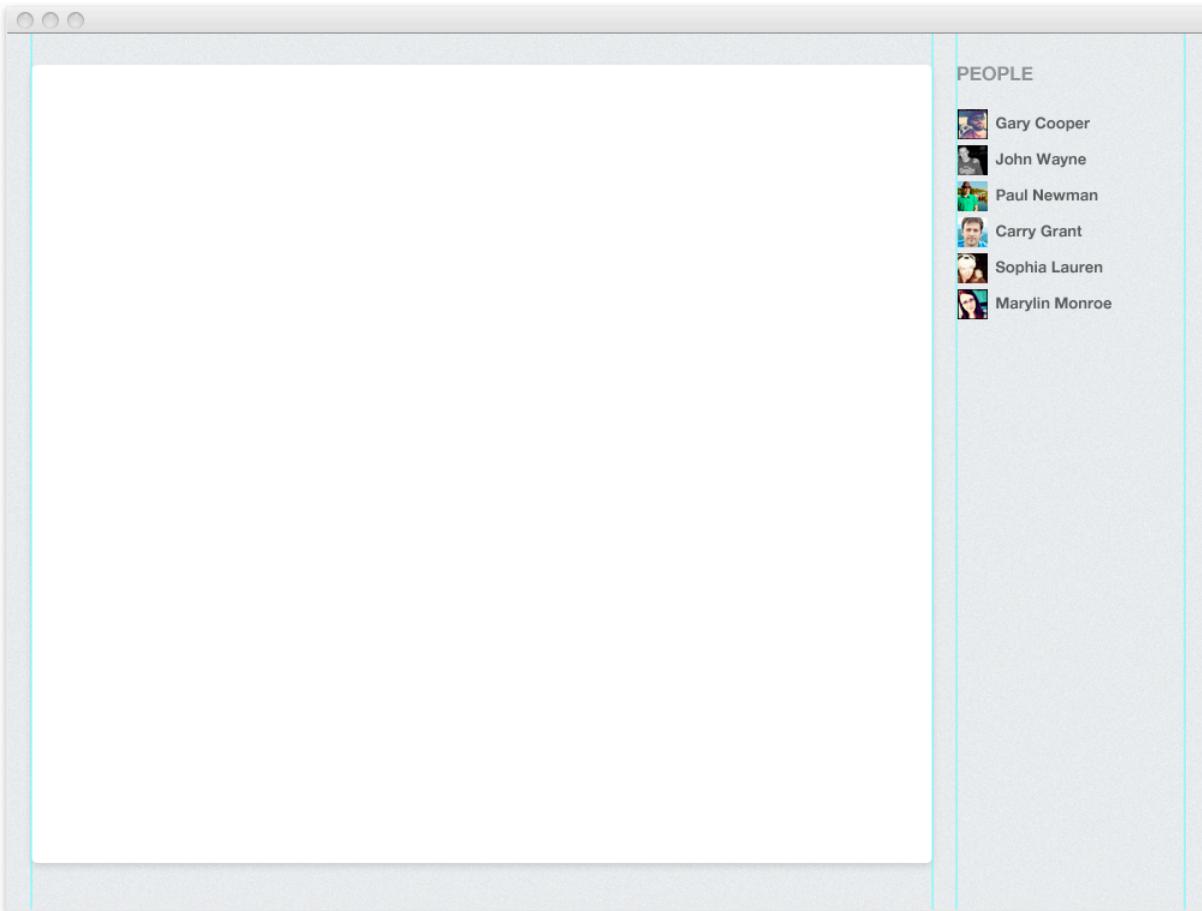
Step 1: Rulers



Designers love going on about their oh-so-precious grid systems. But the truth is that you don't need grids ALL the time. A relatively simple fluid app like the one we're building is a good example of a case where a grid might not be that useful. Instead, we simply set up a couple rulers to make sure our spacing remains consistent.

This might not seem like such a big deal, but **consistent spacing** is one of those little details that can add up to a big difference overall.

Step 2: Main Area & User List



Now let's put in a list of people and a big white area for our main conversation view. You'll notice that the conversation zone is white while the rest of the screen will have a grey background.

This goes back to well-defined priorities and good use of **contrast**: we want to put the most important thing forward, and a white background with strong contrast achieves that goal.

We'll make that contrast even stronger by giving that whole area a faint shadow. This way it'll seem like it's closer to the user compared to the rest of the interface.



Depth

Although UIs are two-dimensional (at least, they should be if you want your users to keep their sanity) it helps to think of them as being composed of **multiple layers**.

Think of **depth** as one extra way to differentiate elements, along with, say, **color** or **size**.

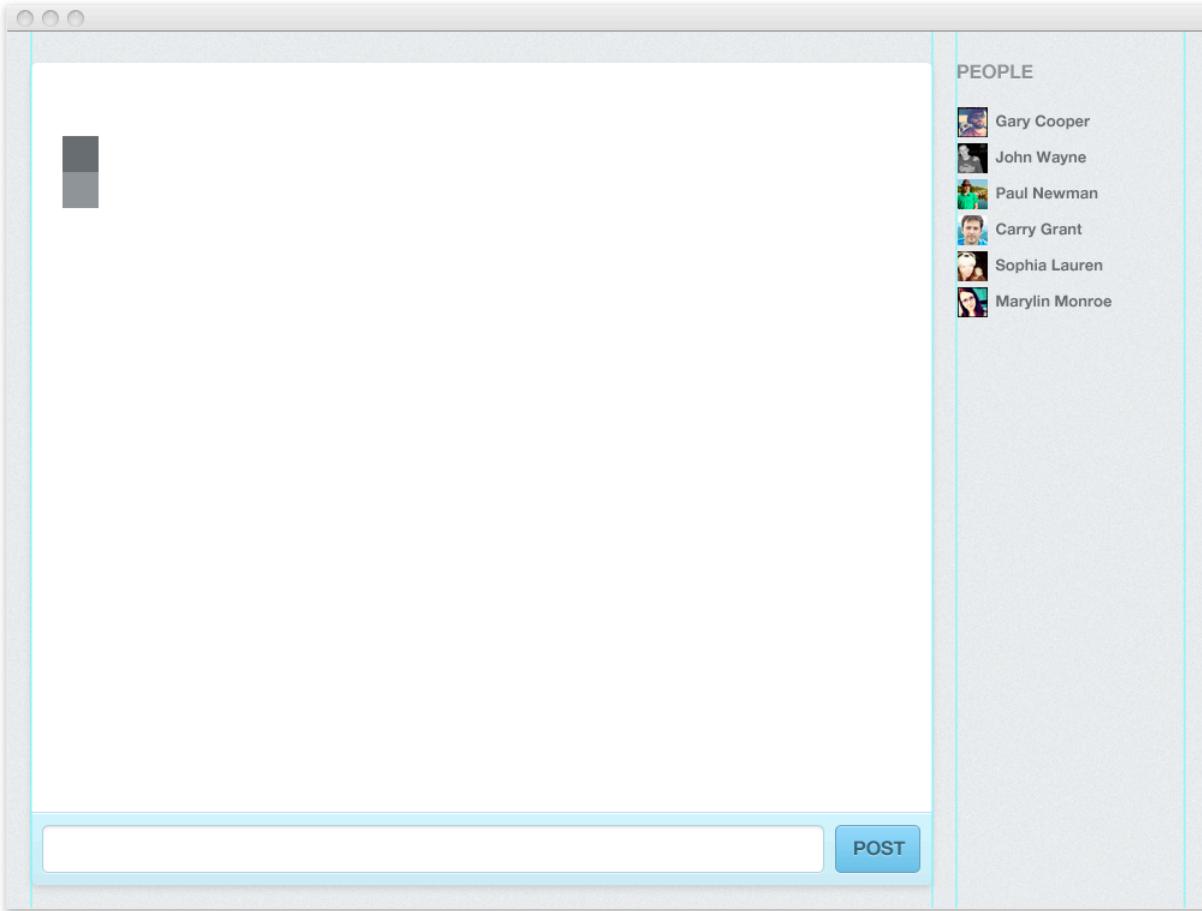
The screenshot shows a website for 'sharypic.pro' with several distinct depth levels:

- Top Level:** A large, semi-transparent callout box with a white background and a thin black border, containing the text "Discover & import photos from anywhere".
- Middle Left:** A screenshot of a desktop application window titled "My Event | Sharypic" showing a photo library with various thumbnail images.
- Middle Right:** A screenshot of a monitor displaying a "real-time Photowall" with a group of people.
- Bottom Left:** A section titled "Do you organize events?" with the subtext "sharypic.pro will keep the Internet buzzing about your event, even when it's over. [Learn more](#)".
- Bottom Right:** A sidebar with links: "About us", "Contact us", "Jobs", and "Press". It also includes a "Follow" button with "273 followers".

DESIGNS CAN OFTEN COMBINE 3 OR 4 DIFFERENT DEPTH LEVELS

But be careful! Once you enter the third dimension you can't go back! Make sure everything stays realistic otherwise you'll introduce ambiguities and incoherences in the UI (think M.C. Escher as UI designer...).

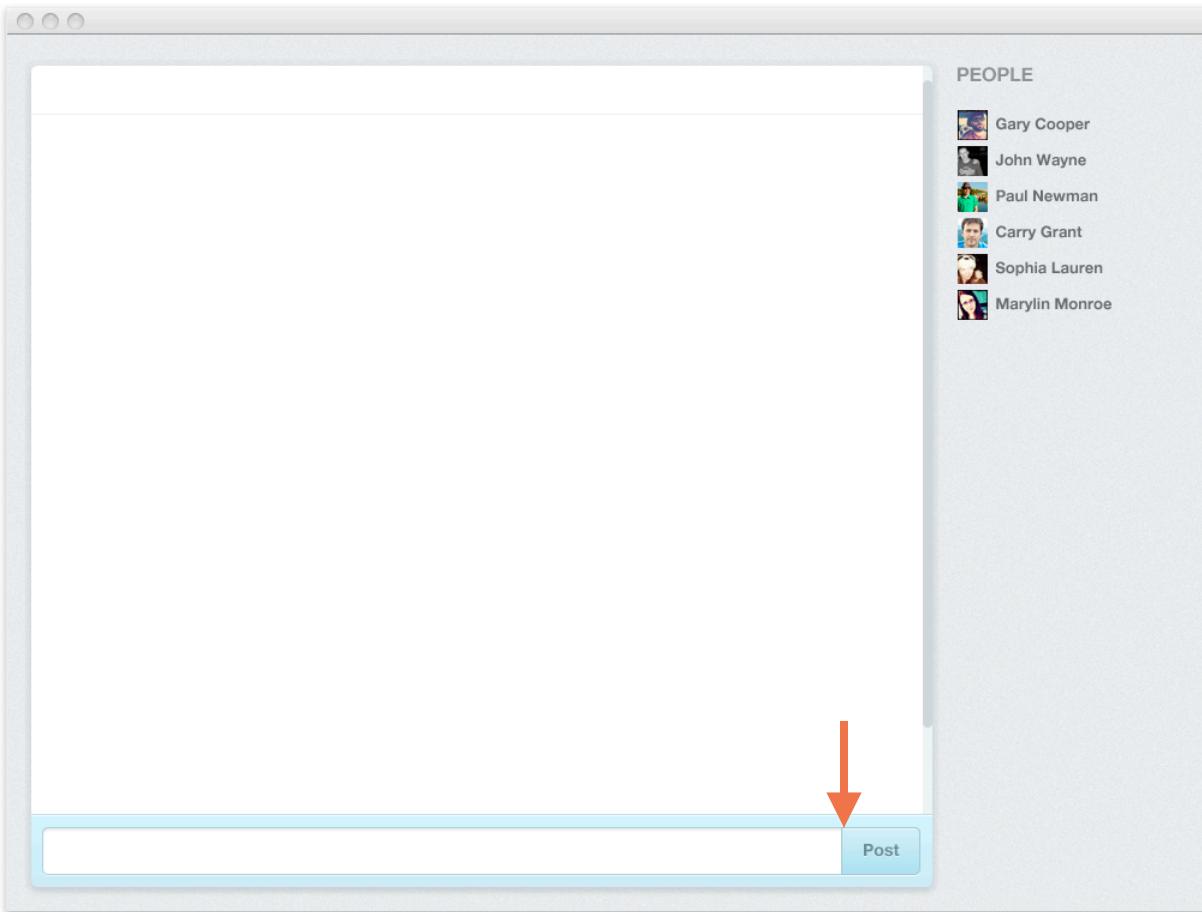
Step 3: The Text Field



The third item on our high-priority list is including a way for people to participate in the conversation, i.e. a text field.

This looks ok, but how can we improve it?

Although *Post* is the whole interface's main action, most people usually simply press enter to send their messages. Having the button there is still useful for people using a touchscreen, but it doesn't need to be that visible.



We can also get rid of a couple lines and some visual complexity by gluing the button right next to the textfield.



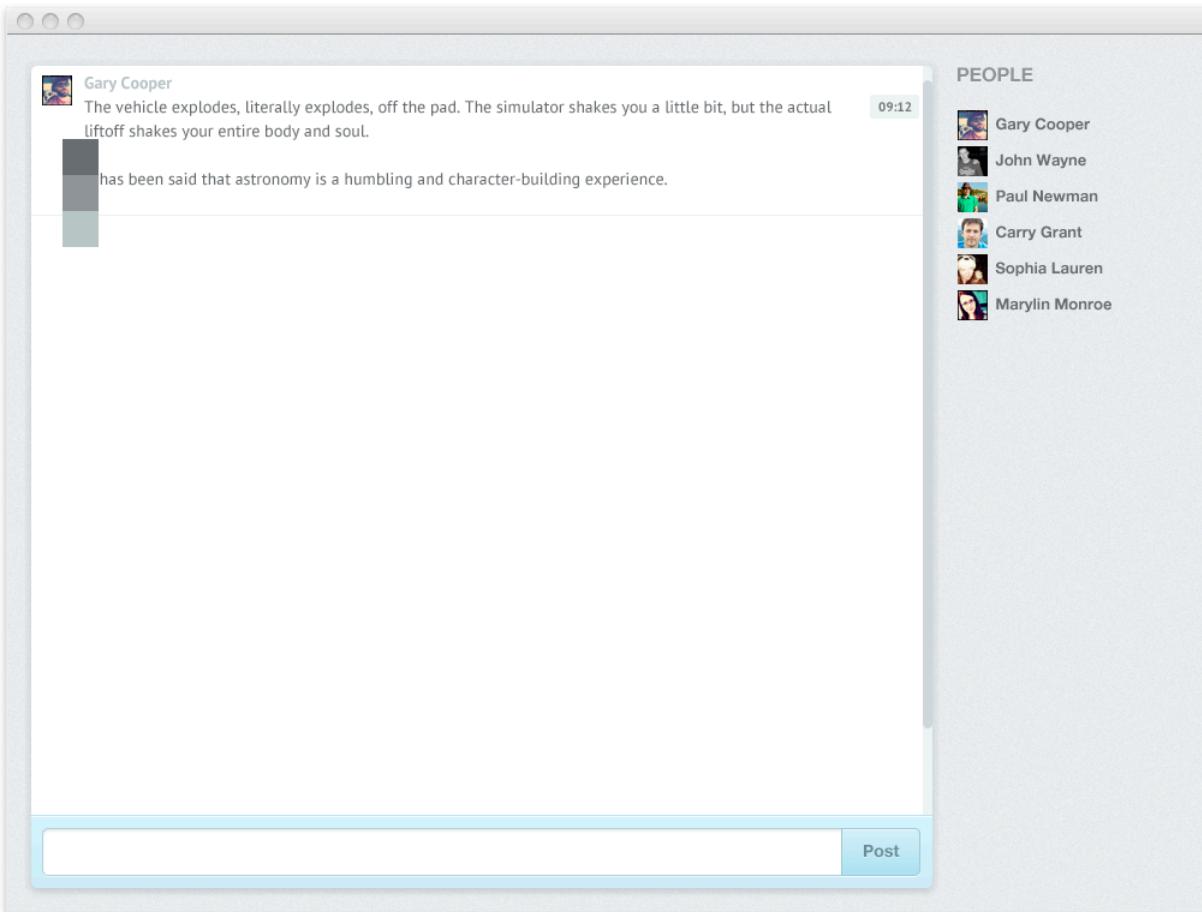
Proximity

Related elements should be positioned next to each other. Even small spaces or lines can confuse the user if misused.

There should be only **one single unambiguous way** to parse your UI, and paying attention to proximity is a good way to achieve this.

Step 4: Conversation

Now let's start fleshing out the main conversation view.



At this point, you might notice that I've introduced a new typeface for the text, the excellent free font **PT Sans**¹ (I had been using **Helvetica** up to now).

Helvetica is a good, safe default choice (in part because Helvetica makes everything look iPhone-y and modern), but I

¹<http://www.fontsquirrel.com/fonts/PT-Sans>

wanted to give the design a little more personality, so I'm trying out PT Sans for a while to see if I like it.



Typography

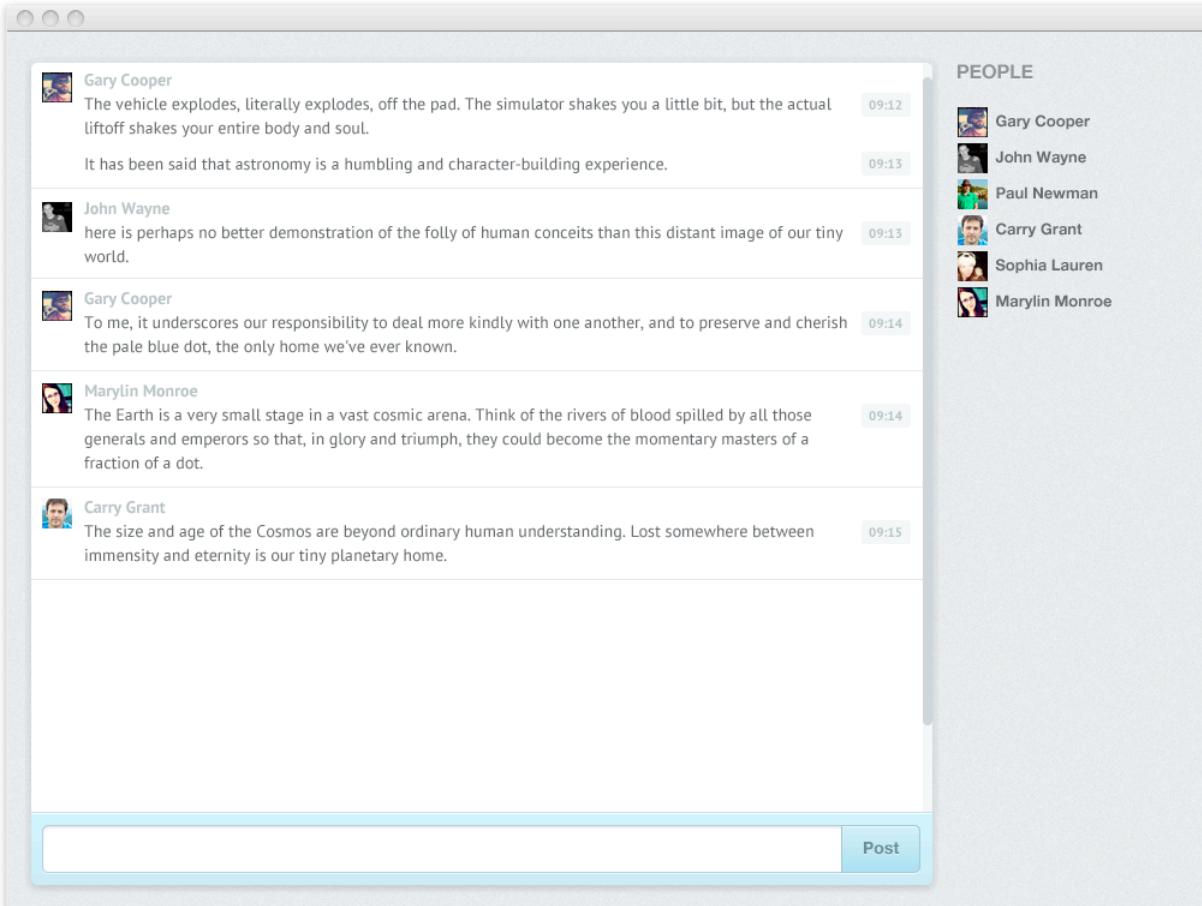
Type is the web designer's **secret weapon**. A great font can single-handedly set the tone for the whole design, make boring text look appealing, or become your company's new brand identity.

There's no easy recipe to master typography, you just need a lot of work. Here are a couple sites to get you started:

- **MyFonts** (premium fonts)
<http://www.myfonts.com/>
- **FontSquirrel** (free fonts)
<http://www.fontsquirrel.com/>
- **Typekit** (fonts as a service)
<https://typekit.com/>
- **I Love Typography** (typography blog)
<http://ilovetypography.com/>
- **Fonts In Use** (real-world examples)
<http://fontsinuse.com/>

By the way, in case you're wondering, those squares floating over in the top left corner are my color palette. It helps me set colors quicker with the eyedropper, and keeps the UI's colors consistent.

Step 5: Conversation, Again



The next step is tedious, but necessary. I duplicated the first message, and changed the avatar and text for each block. The human brain is very good at picking up patterns. If you simply duplicate the same content, your eyes will glaze over and you'll never get a good hard look at the UI.

This is also why I'm using english text (from [Spacelpsum](#)) and not latin. Text is made to be *read*, not *looked at*, and since I can't read latin, I use english filler text to simulate a more realistic user experience.



Flow

You might be wondering if there's any reason why I put the main conversation on the left, and the sidebar on the right. As a matter of fact, there is.

When you design a UI, it's important to think about the flow of the user's attention, which can be visualized with eye-tracking heatmaps.

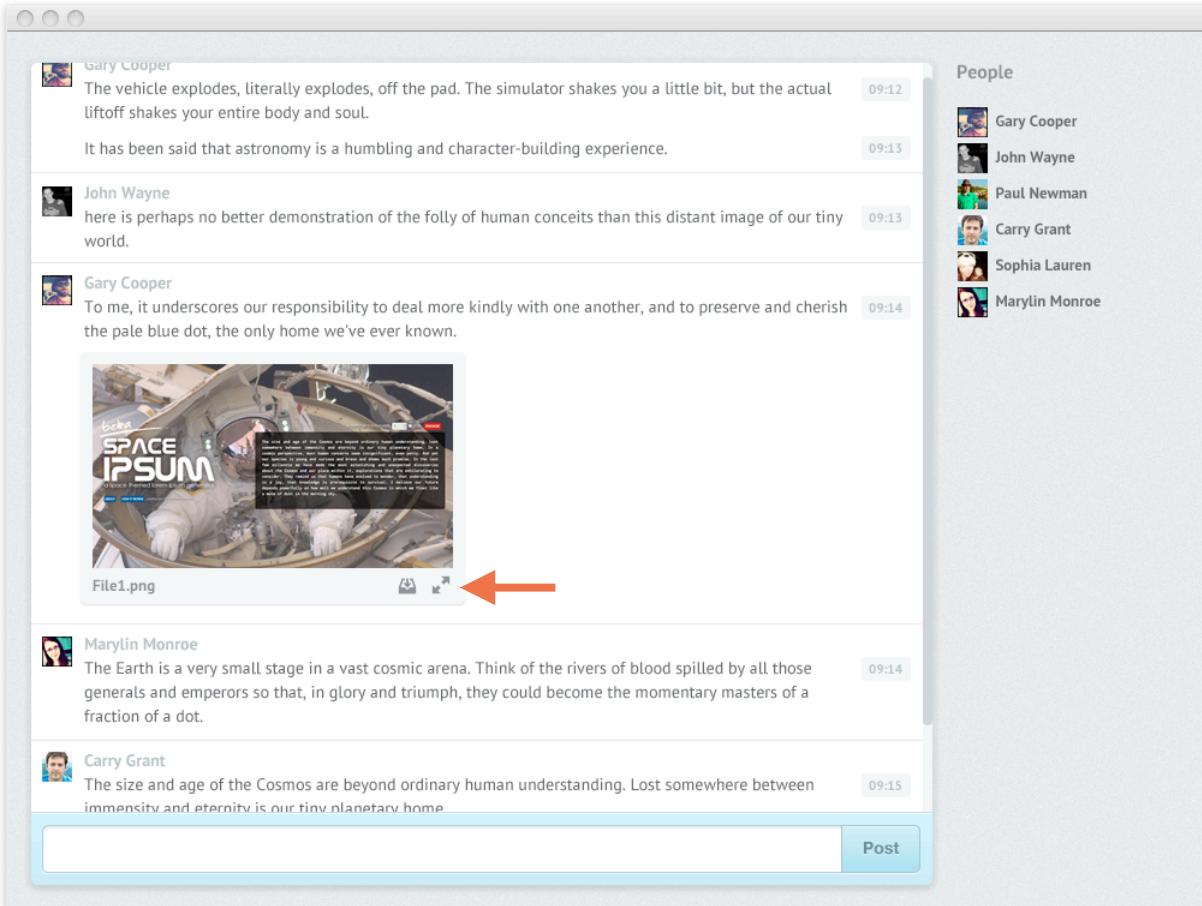


FROM [Jakob Nielsen's Uselt.com](http://JakobNielsen.com)

As you can see, it forms an **F-shape pattern**, with most of the user's attention focused on the left side of the page (since we read from left to right).

As a result, it generally makes sense to put the most important elements (like the conversation) on the left, and secondary elements (like the sidebar) on **the** right.

Step 6: More Conversation



Our spiffy chat app also lets us embed media in the main view. Users will be able to drag and drop files straight into the main chat area, but this kind of modal functionality can often be prototyped straight in the browser so we don't need to mock it up here.

We also have two actions related to media: download and view full-size. Notice I'm using icons instead of text labels.



Iconography

Icons are another powerful weapon for UI designers.

Although they're mainly used for their functional qualities, don't under-estimate their visual appeal:

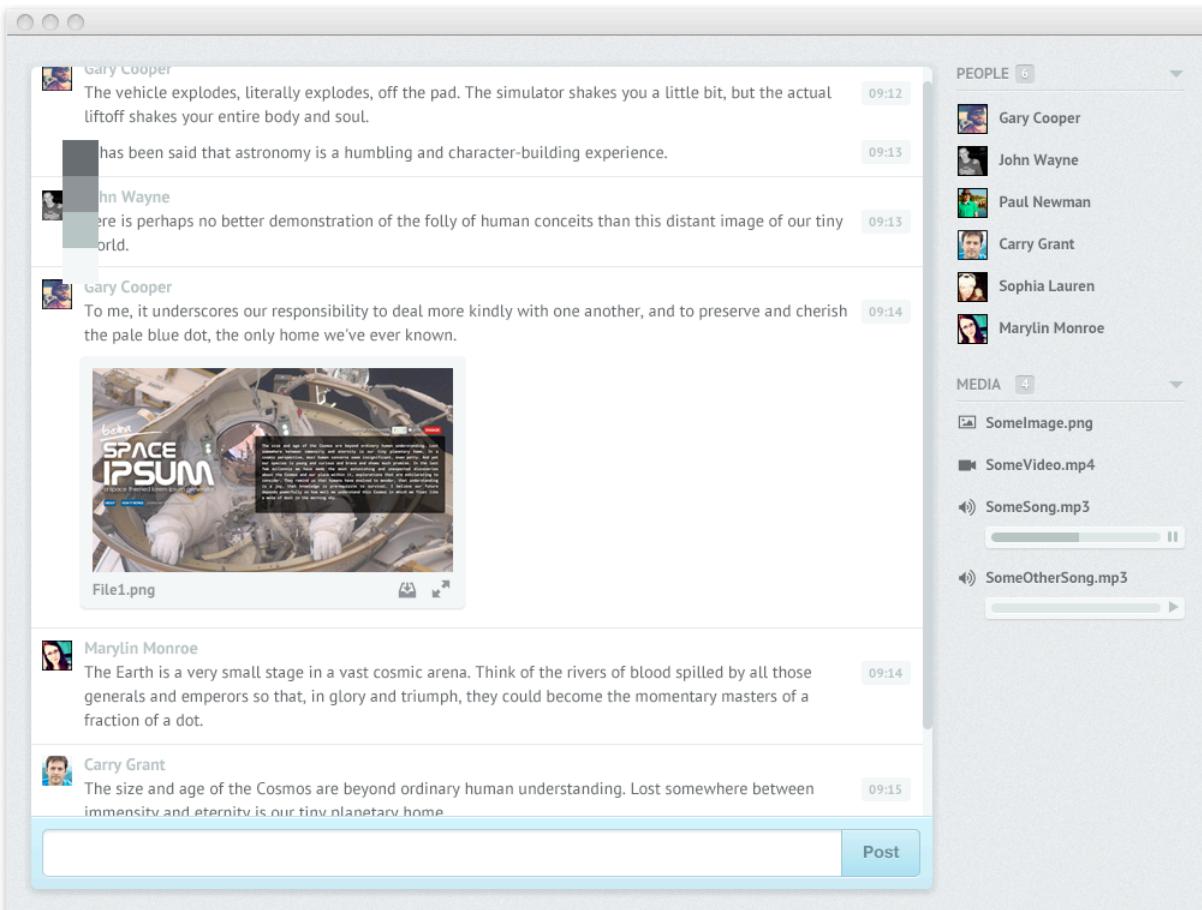


These four icons (from sets [Pictos](#), [Symbolicons](#),

[Steedicons](#), and [Geomicons](#)) all depict an envelope, but they'll each give a different visual style to your interface.

Icons are best paired with a text label, but in some cases (for obvious actions, or less important elements) they can also be used alone.

Step 7: Sidebar

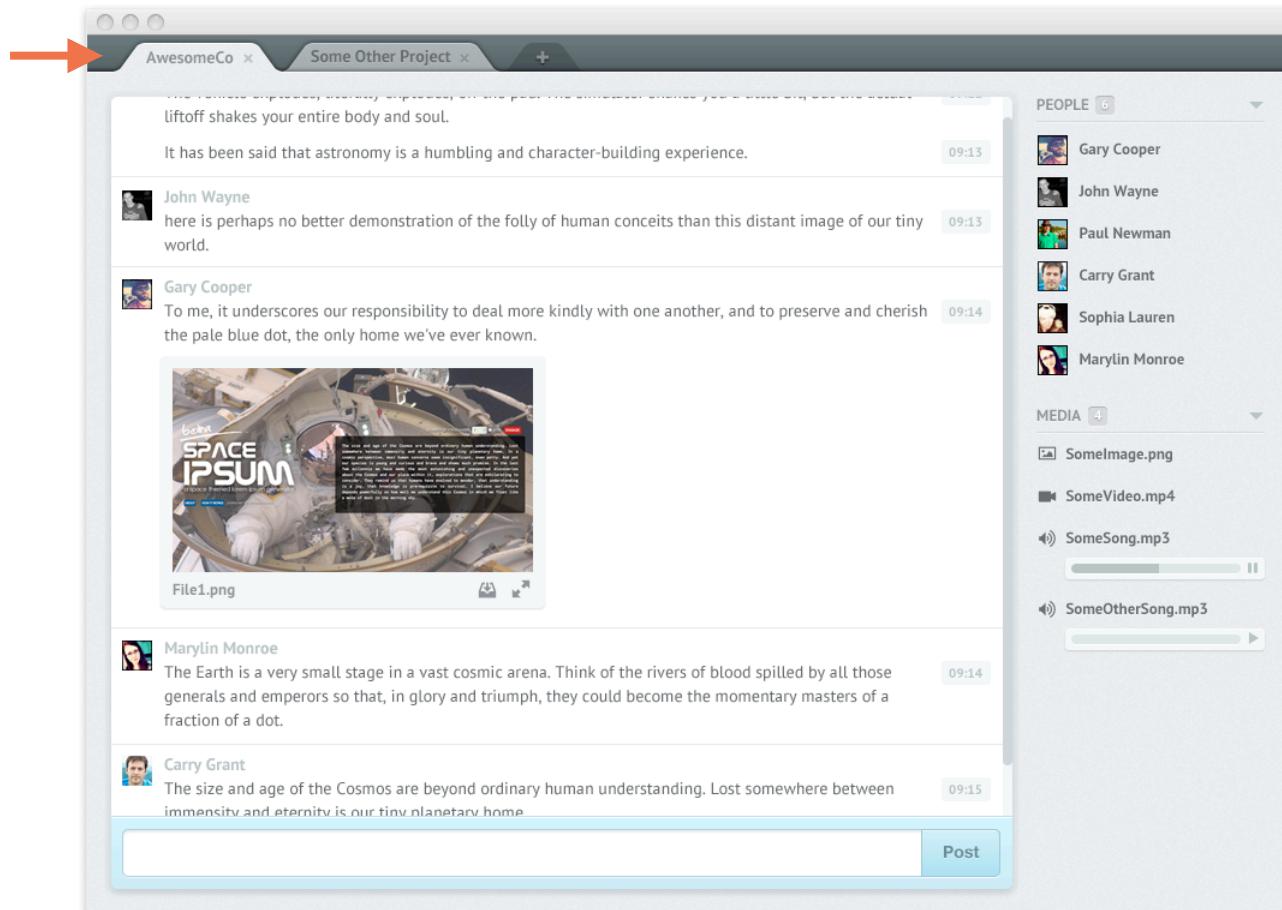


Next step, the sidebar. This chat room app has a special feature that lets you upload audio files and play them to everybody (don't tell the RIAA!).

So we'll use the sidebar both to list uploaded media and control playback. Clicking videos and image links will trigger an overlay that lets you view or download the file.

This leaves a lot of room for improvement, but let's take things one step at a time and come back to this later.

Step 8: Tabs

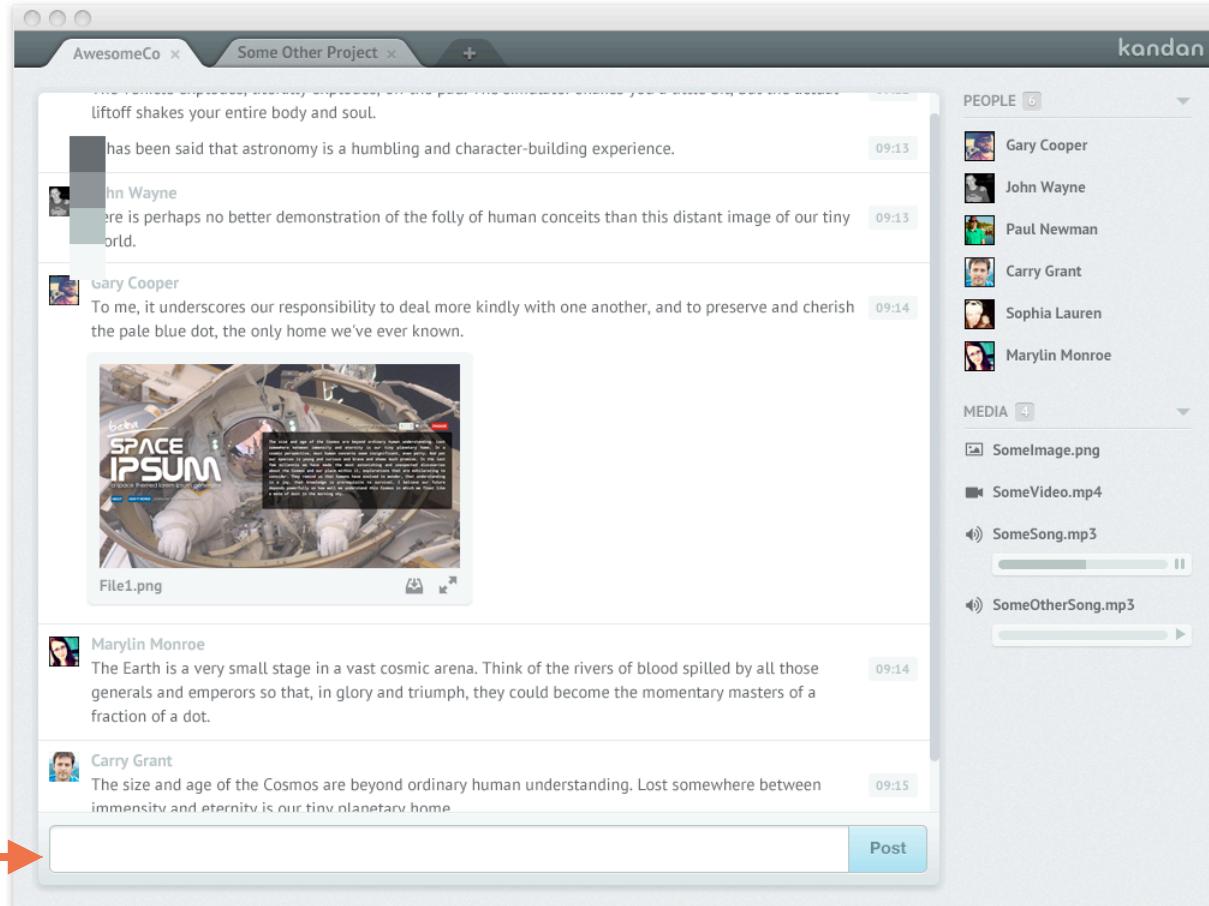


We still have one more feature to design, and that's the tabs to switch from one chatroom to another.

We'll keep things simple and copy Google Chrome, with an additional "+" tab for opening a new tab.

Notice that we changed the depth organization of the design, as the the dark header is now the bottom-most layer, with the grey background resting on top of it as the body of the active tab.

Step 9: Color Tweaks



Now let's add some branding in the top right, and also get rid of that annoying baby blue for the text field's background.

Bright colors might look good the first couple times you see them, but please keep in mind that users will see your work day after day for months, or maybe years. So you should probably avoid harsh colors unless you really know what you're doing.

It's not a coincidence that so much of the UIs we use are pretty much all grey.

And in fact if you look at our current design, we've used only shades of blueish greys, except for the "post" button.



Color

One of the most common questions beginner designers ask is "**how do you pick colors?**".

While you'll find no end of articles on color theory, and even whole sites dedicated to picking colors (such as **Kuler** or **Colourlovers**), choosing colors for UI design is a different matter. For user interfaces, you need to think about the color's **function** first, and only then about its visual appeal.

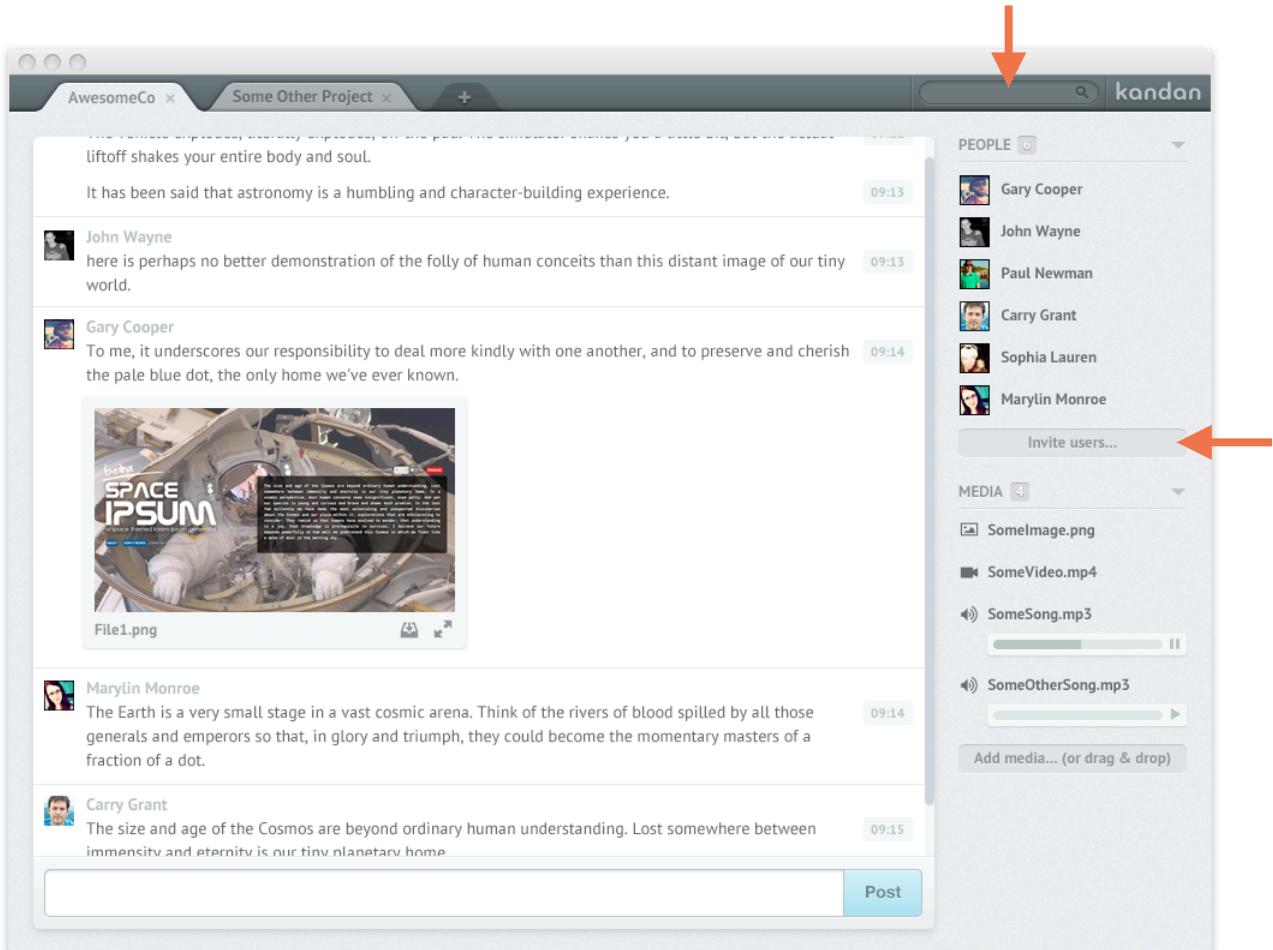
For example, red is almost always associated with deleting, so you probably shouldn't pick it for your main action button. And a green "delete" button would be downright disastrous...



IT'S A BEAUTIFUL PALETTE, BUT WOULD IT WORK FOR A USER INTERFACE?

So rather than picking a color palette for its aesthetic qualities, ask yourself what you want to communicate to the user, and which color is best suited for the job.

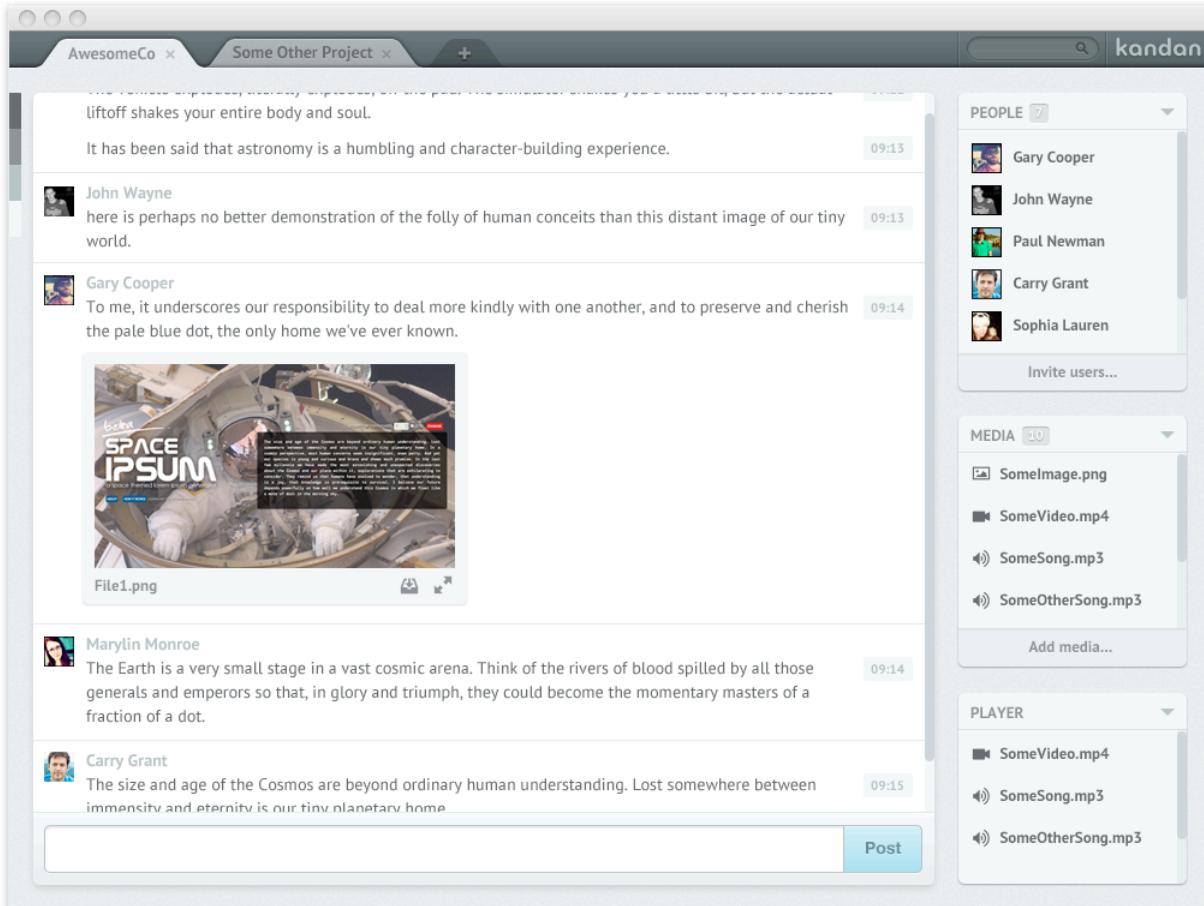
Step 10: More Buttons & Search



We need to provide a way for people to invite other users to the chat room and upload new files. But keep in mind that those are not the most frequent actions, so we don't want to call too much attention to them.

We're also adding a search field in the top bar next to the logo. The search field is greyed out by default, but will get a white background when focused on.

Step 11: Sidebar Improvements



I wasn't very happy with the previous sidebar. It was getting crowded, and the lack of a clear separation between widgets meant that you weren't sure where one ended and the next one started.

The new design makes this much clearer by giving each widget its own little box.

One downside is that these new widgets take up more space, since they require more inner padding. But we can make up for this by giving each of them its own scrollbar.

User interface design often requires to make do with tight constraints and find compromises. In this case, our new widgets might seem a little clunkier, but they're also more flexible and will be able to accommodate more varied content types.



Unity

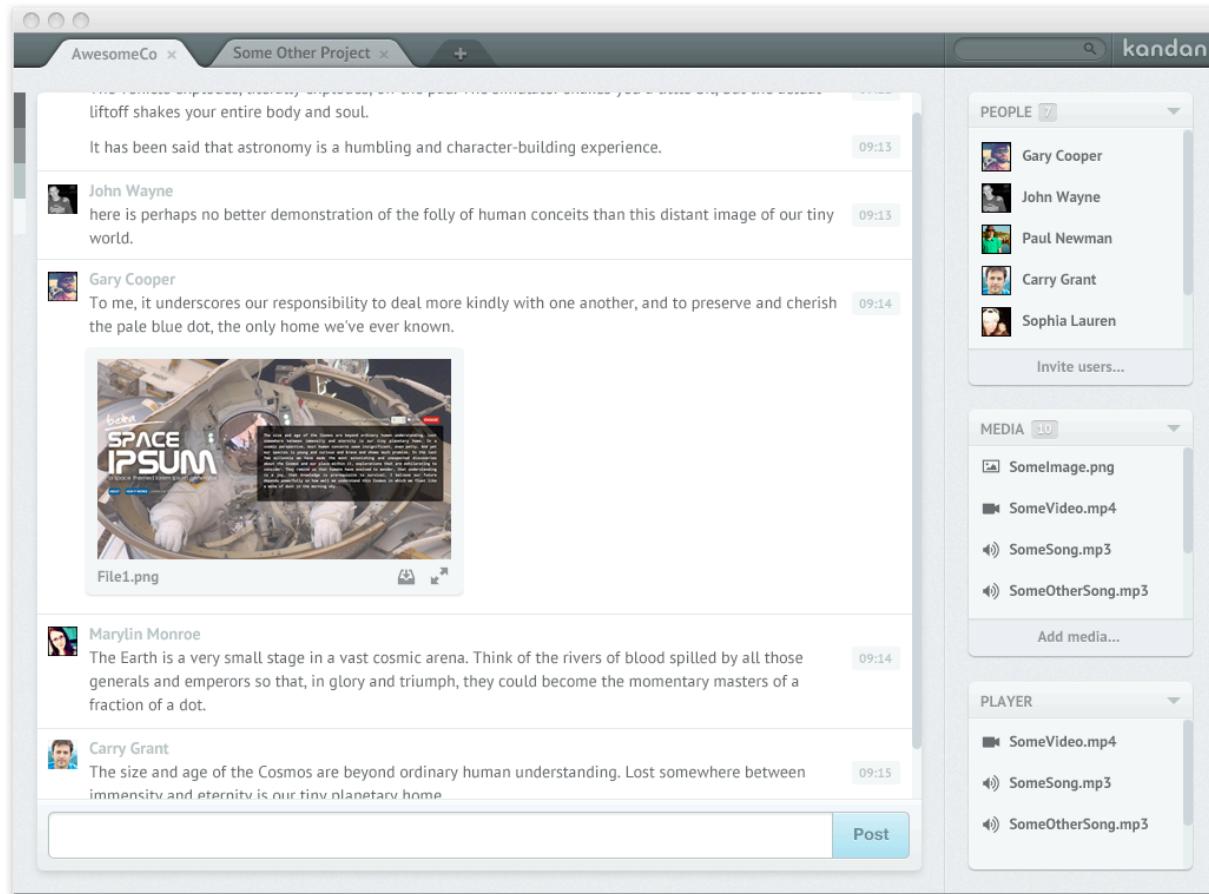
It's important to make sure things are consistent throughout your UI. Our widgets might have different purposes, but by using the same styling we can make sure they all fit together.

Remember that every time you introduce a new visual element, the user has to process it and decide in which mental category (button, widget, text field, etc.) to file it.

So just like you can save computer memory by storing the result of a calculation in a variable (instead of making the same calculation multiple times), you can save the user's brainpower by re-using elements.

As a result they'll be more focused on the rest of the UI, and this in turn will make them more productive.

Step 12: Sidebar Separator

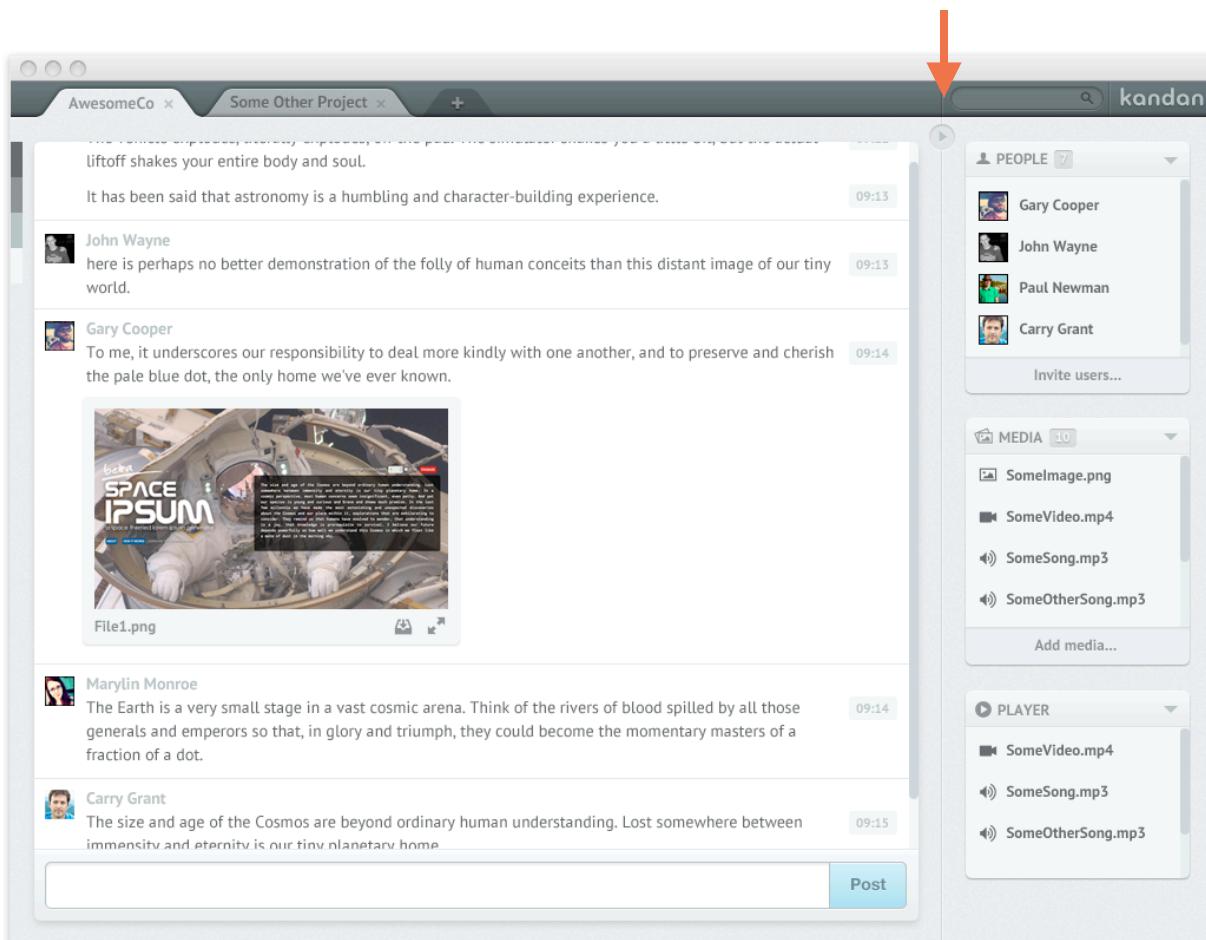


Let's clean up the general sidebar area a bit. We'll add a gradient to each widget header to indicate that it can be clicked to collapse and expand the whole widget.

We'll also add a separation between the sidebar and main content area. We're using the classic UI design trick of juxtaposing one dark pixel and one light pixel to give the illusion of an edge. Once you know that trick, you'll start seeing it absolutely everywhere.

Step 13: Collapse Control

That separation will give us a way to add an element to collapse and expand the sidebar. After all, not everybody will need those widgets. A good UI should be **flexible** and adapt to different types of use.



When you think about it, desktop UIs are amazingly flexible: you can hide and show elements, move windows around, resize them...

Web apps can't (and probably shouldn't) emulate this level of personalization, but they can still give the user a little more control.



Flexibility

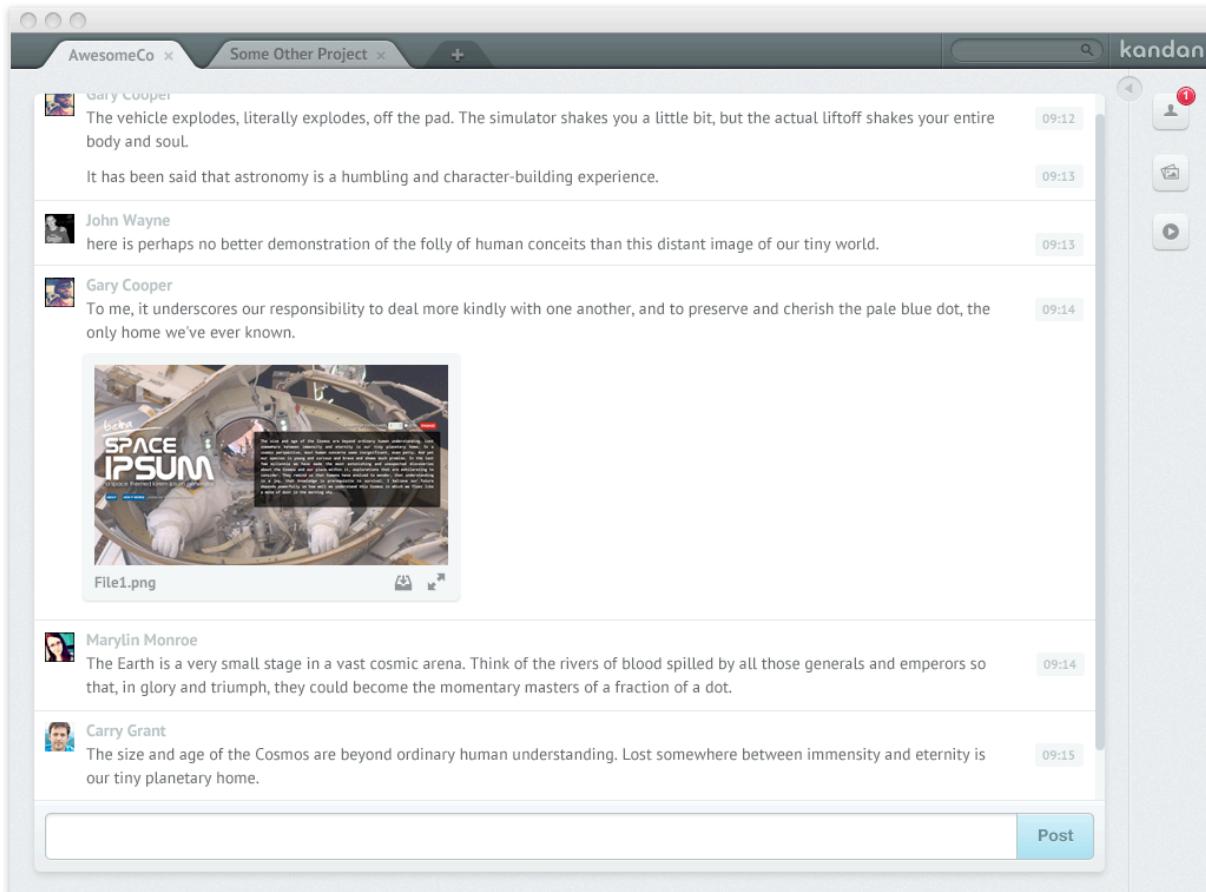
It's not easy to decide how much control should the user have over the UI.

Flexibility is often associated with relatively useless features like personalized color schemes, but its real advantage is letting the UI adapt to **different usage patterns**.

The screenshot shows the Expression Engine Control Panel. At the top, there's a navigation bar with links for Content, Design, Add-Ons, Members, Admin, Tools, Snippets, and Custom Channel Fields. Below that is a breadcrumb trail: Le Maths Club > CP Home. The main area has two main sections: 'Create' on the left and 'Modify or delete' on the right. Both sections have dropdown menus for 'Entry', 'Template', 'Template Group', and 'Channel'. To the right of these sections is a vertical sidebar with icons for various functions like User Management, Groups, and Snippets.

For example, **Expression Engine** (a PHP CMS) lets users customize the top-level navigation bar and add shortcuts to any screen of the app.

Step 14: Collapsed Sidebar



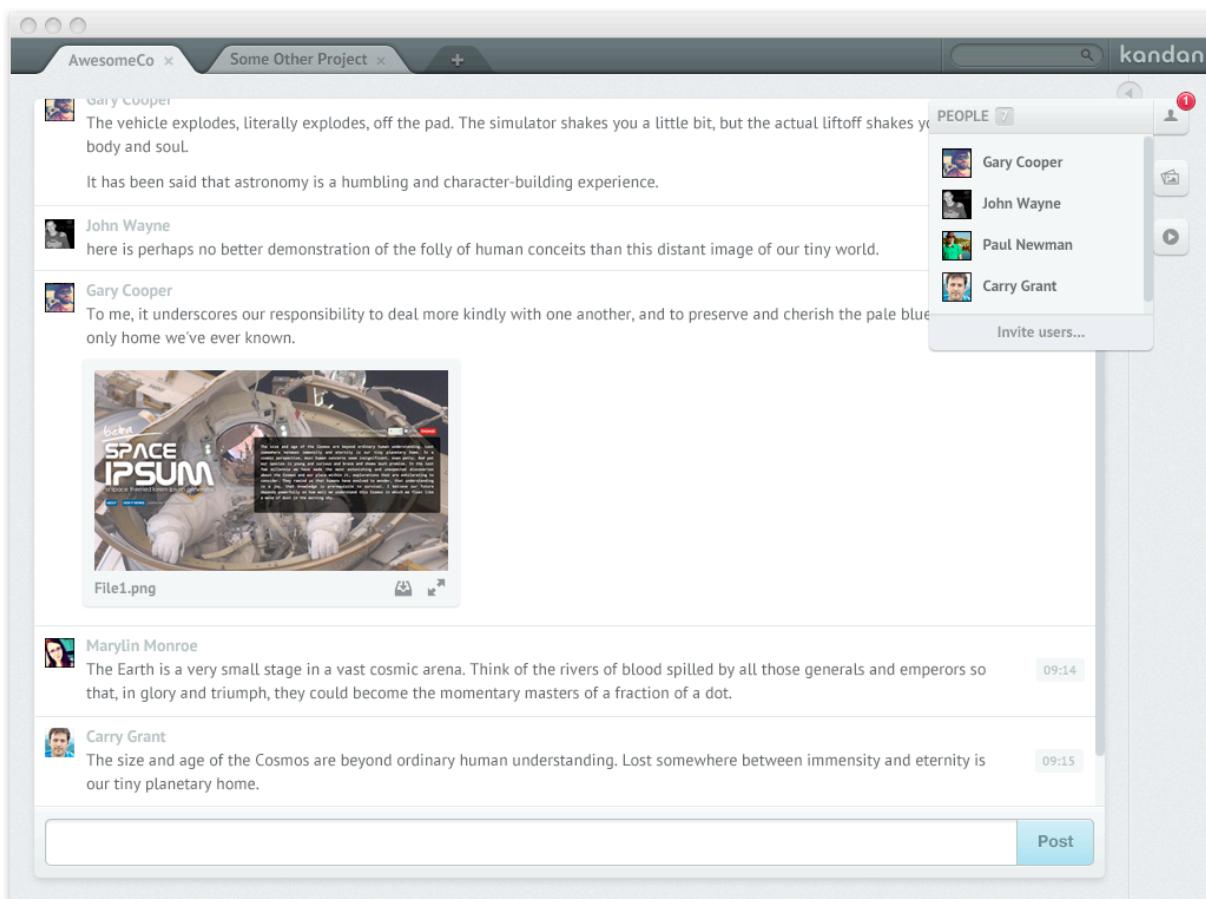
And here's what things look like when we collapse the sidebar.

Even when the sidebar is collapsed, we still want a way to notify users if something happens, like a new user connecting or new media being uploaded. That's where the red notification bubble comes in.

And by the way, the separator line itself can also become a control to resize the sidebar using the standard column resize cursor ()

Step 15: Hover Widgets

Now if you put yourself in the mind of someone's who has just collapsed the sidebar, you'll realize they probably don't want to go through the trouble of expanding it just to check up on new people. So let's introduce a third intermediary states for widgets.



This stage is almost the same as the first one, except it will appear when you mouseover a collapsed widget. Also, we've removed the icon before the widget title to avoid repeating it.

If you're familiar with the WordPress admin UI, you'll have noticed that we're using basically the same pattern. Once again, you don't need to reinvent the wheel every time, and it's in fact better for users if you stick to familiar concepts.

We've kept everything else the same to make it easier to re-use the same code when it comes time to implement the UI. Even if you're not the one who's going to code the design, it's important to **think ahead** and make some practical choices to lighten the load for everybody.



Patterns

One big difference between art and design is that if an artist copies another artist, he's a fraud. But if a designer copies another designer, he's, well, a good designer.

So you shouldn't shy away from using existing design patterns. [Patterntap](#) and [Dribbble](#) are both great resources to see how others have already dealt with a problem.

Step 16: Music Player

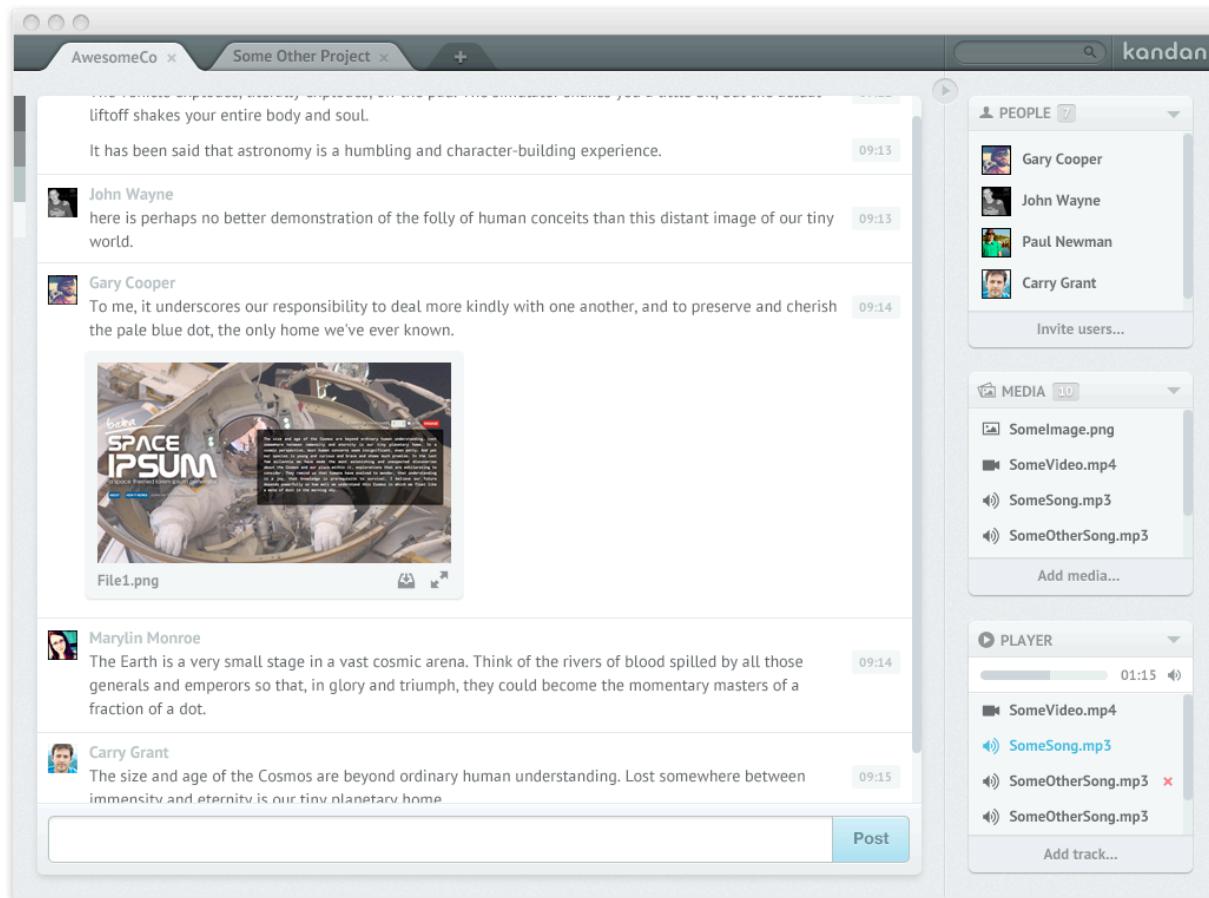
Let's take a break to review what we have so far: we've designed a flexible sidebar layout that can accommodate various widgets while at the same time move out of the way when it's not needed. Pretty neat!

So it's now time to finally design that media player. Let's review our requirements. The media player should:

- Include basic controls for volume
- Show a playlist, and tell you which track is playing
- Show track length and progress
- Let you remove tracks from the playlist (if you're the one who's added them)
- Provide a way to add a track to the playlist

Notice that we don't need a control for stopping the music or moving on to the next or previous track. That's because our player will be a *shared* player, meaning that everybody will listen to the same thing at the same time.

That kind of functional detail is obvious once you're actually using an interface, but can be easy to overlook when you're still at the mock-up stage. This is why it's so important to be able to project yourself in the future and imagine how real people will use what you've designed.



Our player is pretty simple: a progress bar, a time indicator, a mute button, and that's it. The rest of the space is occupied by the playlist and the “Add track” button.

The red “x” to delete tracks would be distracting if it was displayed at all times, and users might also click it by mistake, so we'll only make it appear on hover.



Discoverability

When possible, it's a good idea to hide UI elements until they're actually needed.

This will make the UI simpler, and also draw focus to these elements when they eventually do appear.

Discussion



Add a comment...

Discussion



Bold *Italic* ⬤ Bullets ⬤ Numbers QUOTE

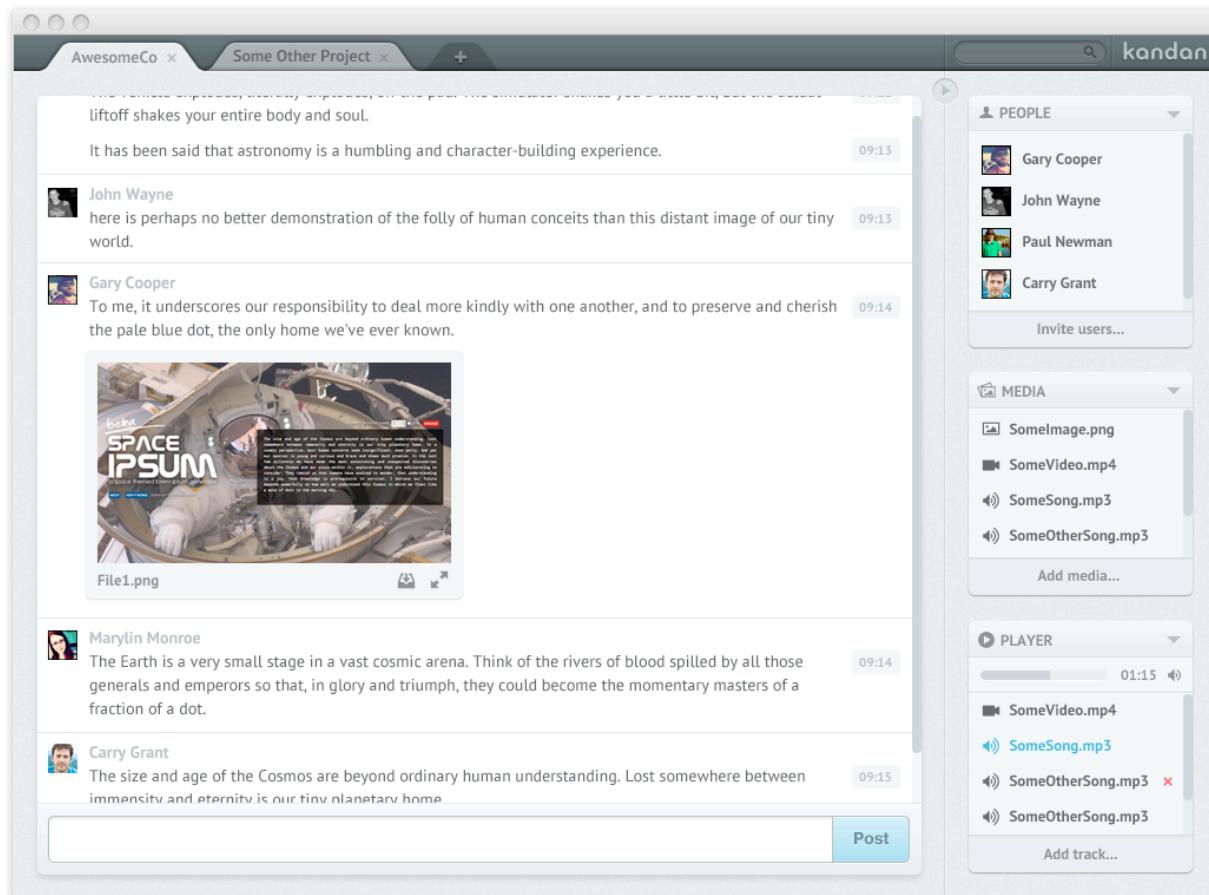
Attach files by dragging & dropping them here or choose them manually...

For example, **Basecamp**'s text formatting controls only appear once you click inside a textfield.

Of course, the trick is deciding which elements you can hide, and which ones should be visible at all times...

Step 17: Cleaning Up

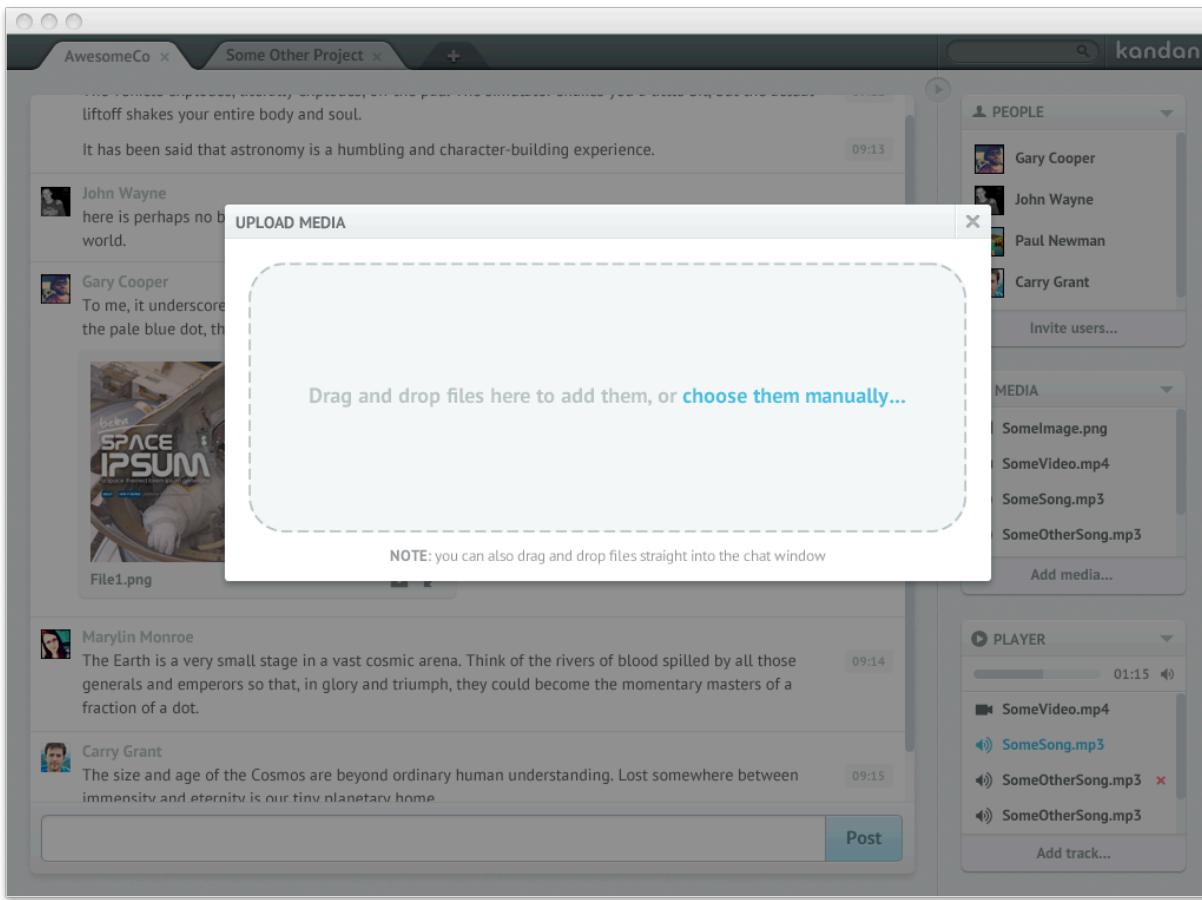
Let's clean this up a bit by getting rid of the numbers next to the widget title. They don't really add anything, and look too much like the red bubbles that pop up when the sidebar is collapsed.



UI design is mostly about those small iterations and improvements. So you shouldn't feel discouraged if you sometimes spend hours tweaking a small detail only to eventually decide your initial design was the best. At least the whole process will have taught you *why* it's the right choice.

Step 18: Add Media Overlay

Let's flesh out the app's interactions with the overlay that lets you add files.

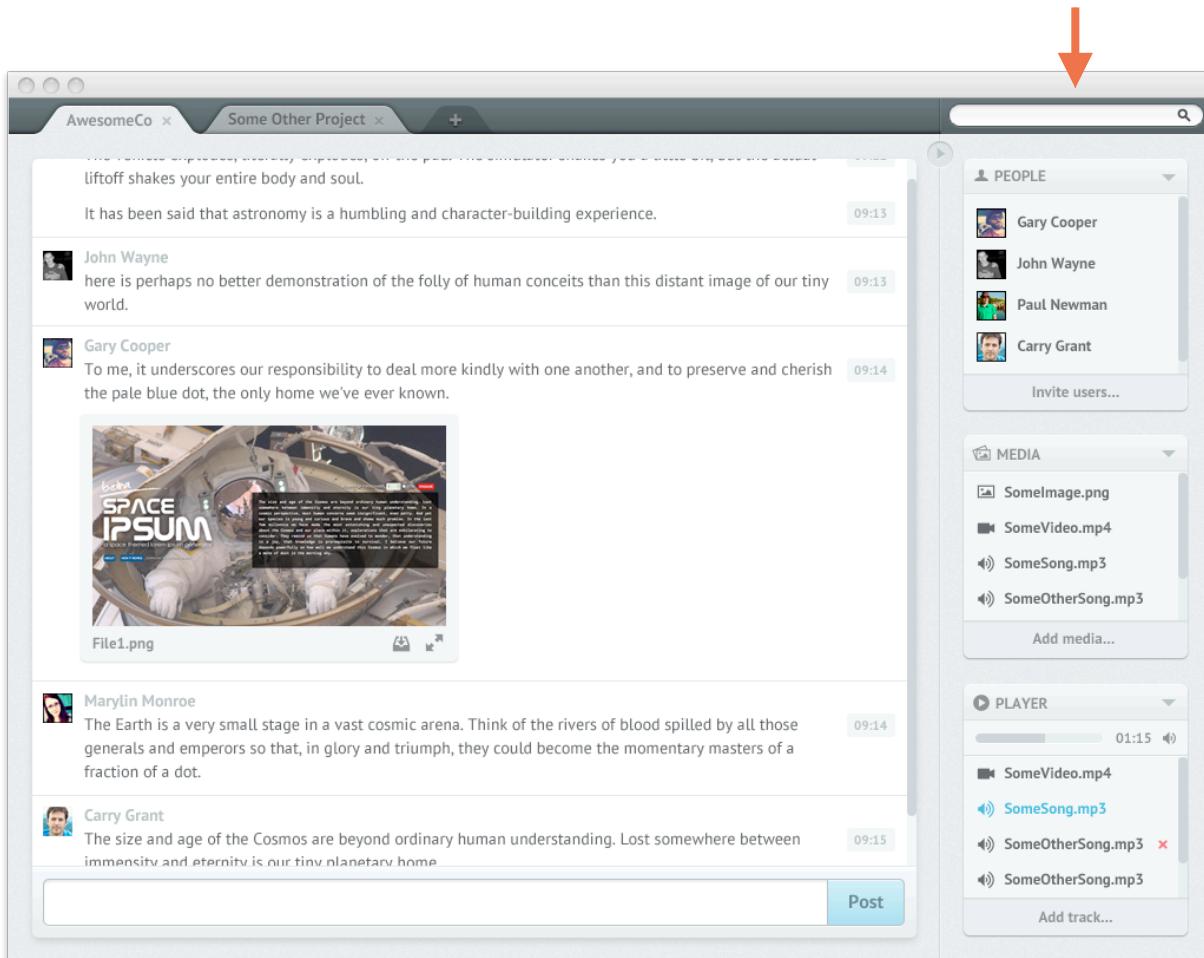


Drag and dropping files into a browser window to upload them is still a relatively new feature (unless you're reading this in 2017, in which case, is IE 6 still around?), so we'll also give users the possibility of doing so through the standard “browse files” dialog.

We'll also remind users that they can skip this dialog altogether by drag and dropping files straight into the main chat window.

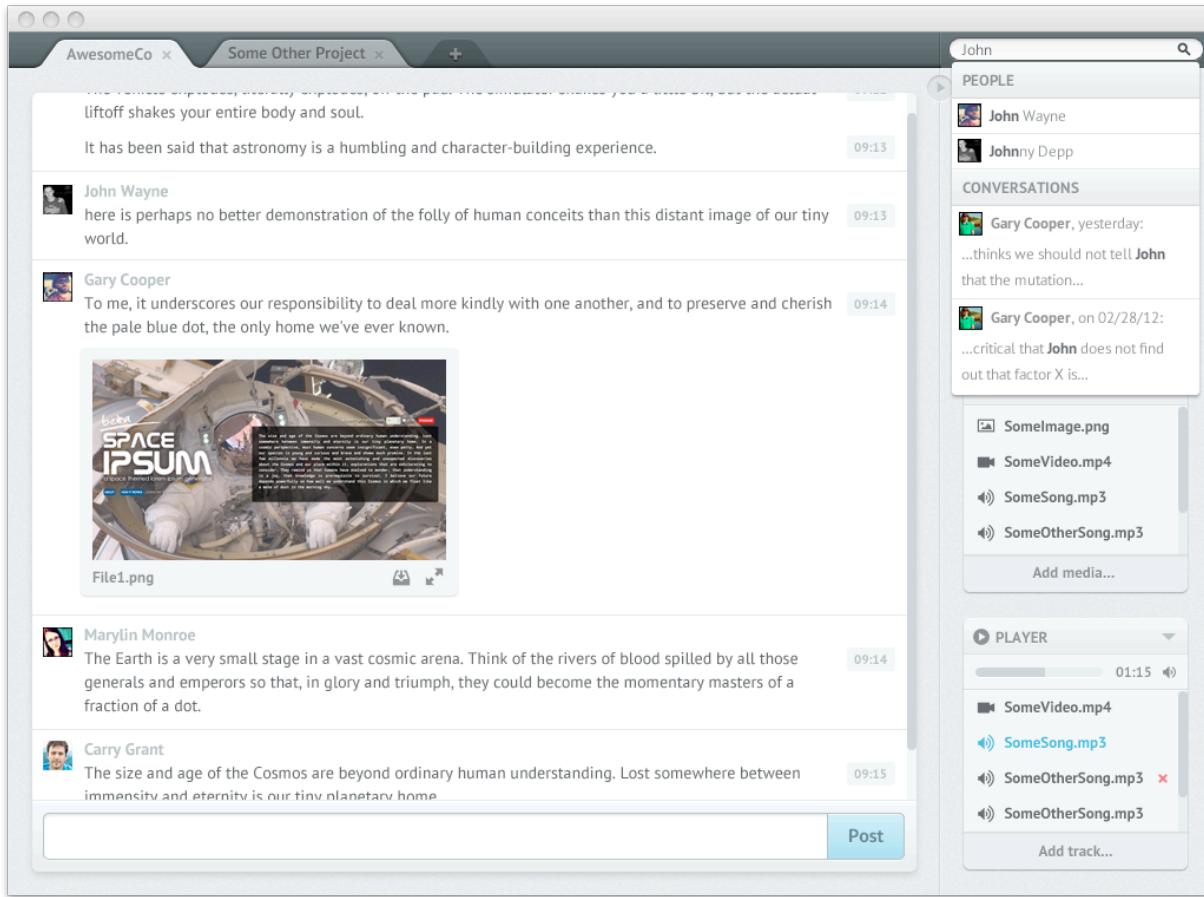
Step 19: Search

The last interaction we'll work on will be the search. Our search will have three states: inactive (the default state), focused (when you click the search field), and active (when you're typing).



When users click the search field, it will expand to show more text (hiding the logo) and also become white instead of transparent.

Note that the placement of the search field outside the active tab helps indicate that this will search across all conversations, and not just the current one.



Once the user starts typing, the search results will appear in a drop-down below the field.

These search results are grouped by type (people, conversation, media, etc.) and loaded via AJAX without requiring to refresh the page.

Clicking a conversation result would open up a new tab with the specific word highlighted in the conversation.

Step 20: Conclusion

I hope you enjoyed this short design exploration, and feel ready to take on your own projects. Here are a couple things to take away from all this:

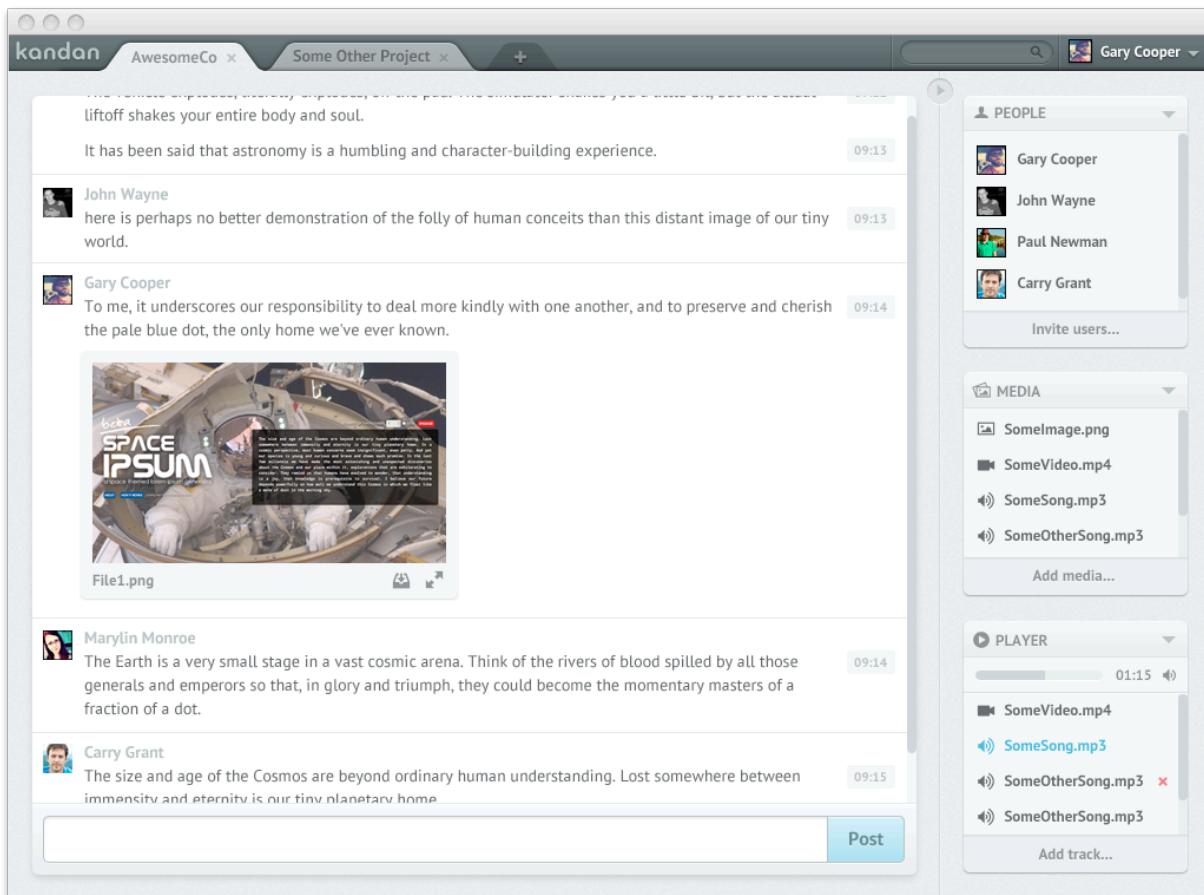
- UI design is an iterative process. It takes time and you'll make mistakes, but that's entirely normal.
- There are a couple principles that you should always keep in mind. Check your design against each of them as you go along, and you should be fine.
- The most important thing is to always keep in mind the final purpose of your UI. Every time you're at a crossroad, ask yourself which choice helps the user achieve this purpose best.

So there you go. UI design is a lot of hard work, but a great UI can really make or break a product, so don't underestimate the importance of good design.

If you want to learn more about all this, I encourage you to [check out my blog](#), [follow me on Twitter](#), or [drop me a line](#) to say hello!

Step 21: Oops!

You might have noticed something very important is missing from my mockup... Instead of trying to hide my mistake, I'll come clean and show you. But first, can you find it yourself?



That's right, I completely forgot the user menu. Without it, how are you supposed to log out or modify your profile?

But no big deal, we can rearrange elements and move the logo to the left. UI crisis averted!

Appendix: Design Resources

There's a ton of design resources out there, so the problem is not so much finding them as not being overwhelmed.

Personally, I find that the most helpful articles are the ones who talk about a real project, have a very specific focus, or are written by very good designers.

By these standards, you should probably avoid round-ups and list posts (i.e. “147 Free Photoshop Brushes”). They’ll waste your time and won’t teach you much.

Instead, here are a few good guides:

- **Crash Course: Design for Startups**
<http://paulstamatiou.com/startup-web-design-ux-crash-course>
- **Design for Hackers: Why Monet Never Used Black, & Why You Shouldn't Either**
<http://www.kadavy.net/blog/posts/d4h-color-theory/>
- **The UI Guide | Part I: Buttons**
<http://galp.in/blog/2011/08/02/the-ui-guide-part-1-buttons/>

These sites have a ton of great design advice:

- **Bjango Articles**
<http://bjango.com/articles/>
- **The Elements of Typographic Style Applied to the Web**
<http://webtypography.net/toc/>

- **Method & Craft**

<http://methodandcraft.com/>

And here are a couple of my own articles:

- **Google Webfonts That Don't Suck**

<http://sachagreif.com/google-webfonts-that-dont-suck/>

- **Just a Big List of Free Design Stuff**

<http://sachagreif.com/the-design-freebies-list/>

- **Tools of the Trade**

<http://sachagreif.com/tools-of-the-trade/>

And some case studies about my own work:

- **How I Designed CodeYear.com in One Hour**

<http://sachagreif.com/how-i-designed-codeyear-com-in-1-hour/>

- **Making Of: Sharypic.com**

<http://sachagreif.com/making-of-sharypic-com/>

- **Making Of: Sharypic for iPhone**

<http://sachagreif.com/making-of-sharypic-for-iphone/>