

How white space killed an enterprise app (and why data density matters)

Co-Written by [Patrick Deuley](#).



Christie Lenneville [Follow](#)

Feb 5, 2019 · 5 min read

Spacious. Minimalist. Clean. Bountiful white space has become the de facto design aesthetic in consumer apps.

And I'm not here to hate on the trend. Used effectively, white space is attractive and can greatly improve the usability of a simple interface. Long live bountiful white space!

But what about complex interfaces? Enterprise software designers know the ones I mean: technology-enabling control panels, data-intensive logistics systems, and number-heavy accounting systems. The tools our business users rely on to get their jobs done every day.

Let me tell you a sad story.

The protagonist is a well-intentioned UX Designer at a large high-tech company who was given a new project: Redesign an internal control panel that was ugly, hard to learn, and stuffed full of content on every screen (so much data). Everyone agreed it needed modernization—it looked like it was from the early 2000s, after all!

So this designer set out to solve the problem, taking cues from modern consumer apps.

The new design simplified every screen. It broke apart huge pages into smaller, more focused ones. It used progressive disclosure to hide presumably insignificant information. And since today's users don't mind scrolling (ahem), the design

incorporated white space in all of the usual places—around headers, content blocks, and in table rows. The breathing room was glorious.

It lasted one month before the company was forced to retire it.

Users absolutely hated the new system. Sure, the old system was ugly, but it had everything they needed, right at their fingertips! Their jobs were incredibly fast paced—they worked in a tech support call center and were rated on productivity metrics. They didn't have time to click or scroll to find information while the clock was *literally* ticking.

In their eyes, this new system wasn't an upgrade, it was a boondoggle. It wasn't just a little frustrating—it *made them mad*.

What's the lesson here?

A large business by its nature has massive-scale data and usually thousands of users who directly interact with it—searching, manipulating, reporting, and more. They need to move through that data quickly, without a lot of digging around in the interface.

Think of it this way: Just like you wouldn't appreciate a dictionary with only 10 words per page (so many pages to flip through!), enterprise users don't want systems that make them work to find the things they're looking for.

But that doesn't mean your interface has to be ugly.

Well-designed data density can present massive amounts of information on every screen, while still maintaining a clear and scannable content hierarchy. Done right, users can quickly and accurately access the data they need, without impacting their ability to read text or interact with the application as a whole.

Following are some recommendations on how to increase data density without decreasing the modern aesthetic of your application.

First, try reducing data volume

Before you start shrinking your font sizes to cram more data on every screen, see if all of that data is even necessary. Ask yourself:

- **Can I eliminate redundant data?** So many systems accidentally display the same data multiple times with slight variations. (Massive tables are the first place to start looking for duplicative information.)
- **Are there things my users really don't need to see right now?** If you don't know, ask! Are they really using everything that's available to them?
- **Can I do a better job of grouping related information?** Not every discrete data element always needs its own table cell, even if that's how it's stored in the back-end system. See where you might reduce visual complexity by combining elements — for example, by putting a customer's first and last name in the same field.

Last Name	First Name	Street	City	State	Zip Code	Phone
Doe	John	1234 Main Street	Los Angeles	California	54321-1234	555-555-1212

Harder to scan

Name	Street Address	City/State/Zip	Phone
John Doe	1234 Main Street	Los Angeles, California 54321-1234	555-555-1212

Easier to scan

Use good typographic standards

“Spacing is essential for rapid reading of long, fundamentally meaningless strings, such as serial numbers, and it is helpful even for shorter strings such as phone numbers and dates.”

– Robert Bringhurst, *The Elements of Typographic Style*

Appropriate typography is essential in an enterprise application. Font weights, kerning, and spacing all affect how quickly and easily your users can scan information.

- **Consider using monospaced numbers** when comparing digits between rows matters.
- **Consider decimal-alignment of currency** when dollars and cents are important.

- **Keep line heights narrow**, but be careful to use enough white space to separate objects cleanly.

Use color deliberately.

Reduce the color of less-important content wherever possible. Using a conservative palette in your design makes the color you *do* use more impactful and open to interactive or visual meaning—like in error messages (usually red) or link text.

Customer Name	Status	Balance	Account
Christal O'Connor	Active	\$0.00	Business
Georgette Nivens	Active	\$0.00	Business
Lenna May	Inactive	\$462.40	Individual
Nettie Applebaum	Active	\$0.00	Custom

Customer Name	Status	Balance	Account
Christal O'Connor	Active	\$0.00	Business
Georgette Nivens	Active	\$0.00	Business
Lenna May	Inactive	\$462.40	Individual
Nettie Applebaum	Active	\$0.00	Custom

In the above example, note the prominence of the single red highlight in the gray-themed table, thanks to the lack of other color. Each row suffers slightly in its horizontal scannability, but at the gain of increased table-wide scannability. Consider what is more important for your application.

Use less "furniture"

In discussing making more readable tables, Robert Bringhurst notes,

“There should be a minimum amount of furniture (rules, boxes, dots, and other guide rails for traveling through the typographic space) and a maximum amount of information.”

– Robert Bringhurst, *The Elements of Typographic Style*

When increasing the density of your application, think about which elements on your page are ultimately aesthetic and not valuable to the internal logic and visual structure of the page.

Let users export data, instead

Sometimes you just can't be dense enough. When dealing with large data sets, long lists, complex tables, or page after page of results, sometimes the best solution is to enable users to take that data to another tool where they can interact with it differently.

Consider adding affordances for your users to export the data they're looking at (or a superset of it! what if your export could include *more* data than you can fit on the screen?) via XML, XLS, JSON, or CSV.

Don't forget about touch

Unless you can safely rule out touch-based interaction, don't forget about minimum sizes for touch targets. For example, Material has good guidelines for the layout of touch targets on their accessibility page that you should definitely check out.

Look at Material's Updates

In recent updates to Material, Google included new standards that increase data density, making it significantly more usable for enterprise applications. They have well-considered, specific recommendations that come with a ton of wonderful examples of how density can affect your layout.

Function over form, always

Enterprise designers can take a lot of great cues from consumer design, but functionality must always be our primary concern. That's because our users don't have a choice, they *have* to use the tools we design—if they don't think a system is usable, they can't decide to just go download a different one. So we owe it to them to put their daily productivity first—always.



Join our weekly newsletter

UX-related articles (like this one) every week in your inbox

- If you are ok with us sending you links via email, please tick the box
- I agree to leave Uxdesign.cc and submit this information, which will be collected and used according to [Upscribe's privacy policy](#).