

2wāv

Design Guide for Programmers





Design Guide for Programmers

A user interface designer specializes in the zone where computer code meets a live human being. Good interface design can mean the difference between success and failure of a software product. The line between back-end and front-end programming is not always distinct.

The most popular front-end frameworks have some built-in styles that are less than ideal. To make matters worse, so many developers use these frameworks that many websites look the same. The first thing designers learn is *not* to use defaults. Every choice should be intentional.

This guide outlines my processes and best practices. Thinking through the user experience will help you make good choices from the start, and being familiar with my design and typography tips will help with good choices on the fly.

—Gretchen Wieshuber

Founder of Studio 2D and Creative Director at 2wav

- Copyright © 2021 by 2wav inc.
- All rights reserved. This book may
- be shared freely in its entirety.
-
- No portion may be reproduced without
- the express written permission of
- 2wav inc., except for the use of brief
- quotations in a book review.
-
- Contact us at info@2wav.com.
-



CONTENTS

- 3 The User Experience
- 5 Elements of Style
- 11 Interface Design
- 16 Design Tips
- 18 Typography Tips
- 24 The Art of Delight
- 25 Accessibility
- 26 We Are 2wav

The User Experience

Don't write a line of code until you know who you are writing it for.

An app or website exists to allow a user to accomplish a goal. Always keep the user foremost in your mind. Before the first line of code is written, or the first mockup designed, understand the user story. (A project might have more than one user story.)

The path to an effective user story is to ask a lot of questions. Don't make assumptions. Those deeply involved in the project often have difficulty pivoting from their own objectives to those of the user. A designer has the mindset of a curious outsider. These are some of the questions I ask:

- Who is the target audience?
- What is the problem or pain point that your app solves?
- How do you want them to feel about your app?

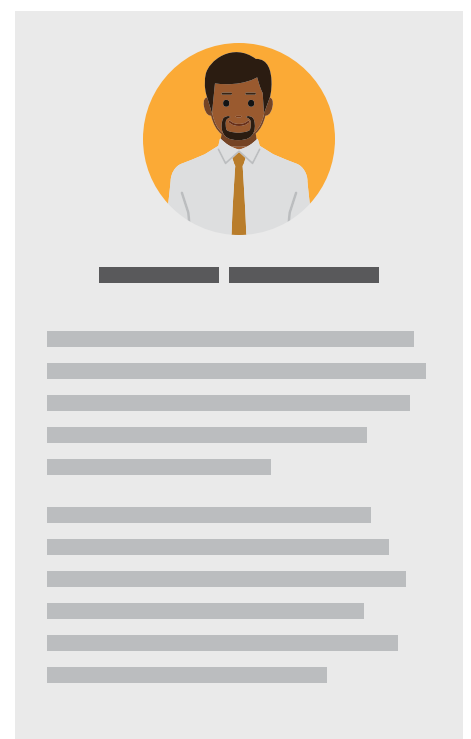
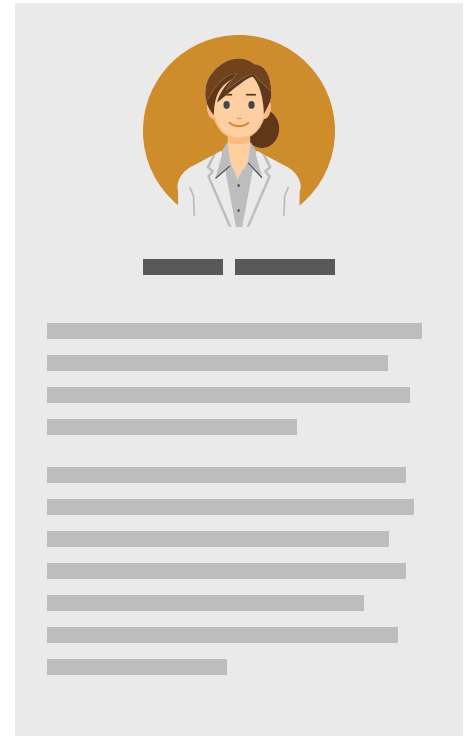
USER STORY

The best user story starts by describing a person with demographic information: age, income, education level, perhaps city, profession, and family as well. Give this person a name. Then add psychographic information: values, attitudes, and aspirations.

Write a story starting with how your user becomes aware of your app and how she first arrives there. What is the first thing she sees? What are the first choices she has to make? Do you want to put up some barriers to entry? Or reduce barriers?

Be crystal clear on what your user hopes to accomplish on the app or website. Think about how many clicks will be necessary to achieve the desired goal, and minimize them. What will the user do once the goal is accomplished?

What if the user becomes confused or has a question when using the app?



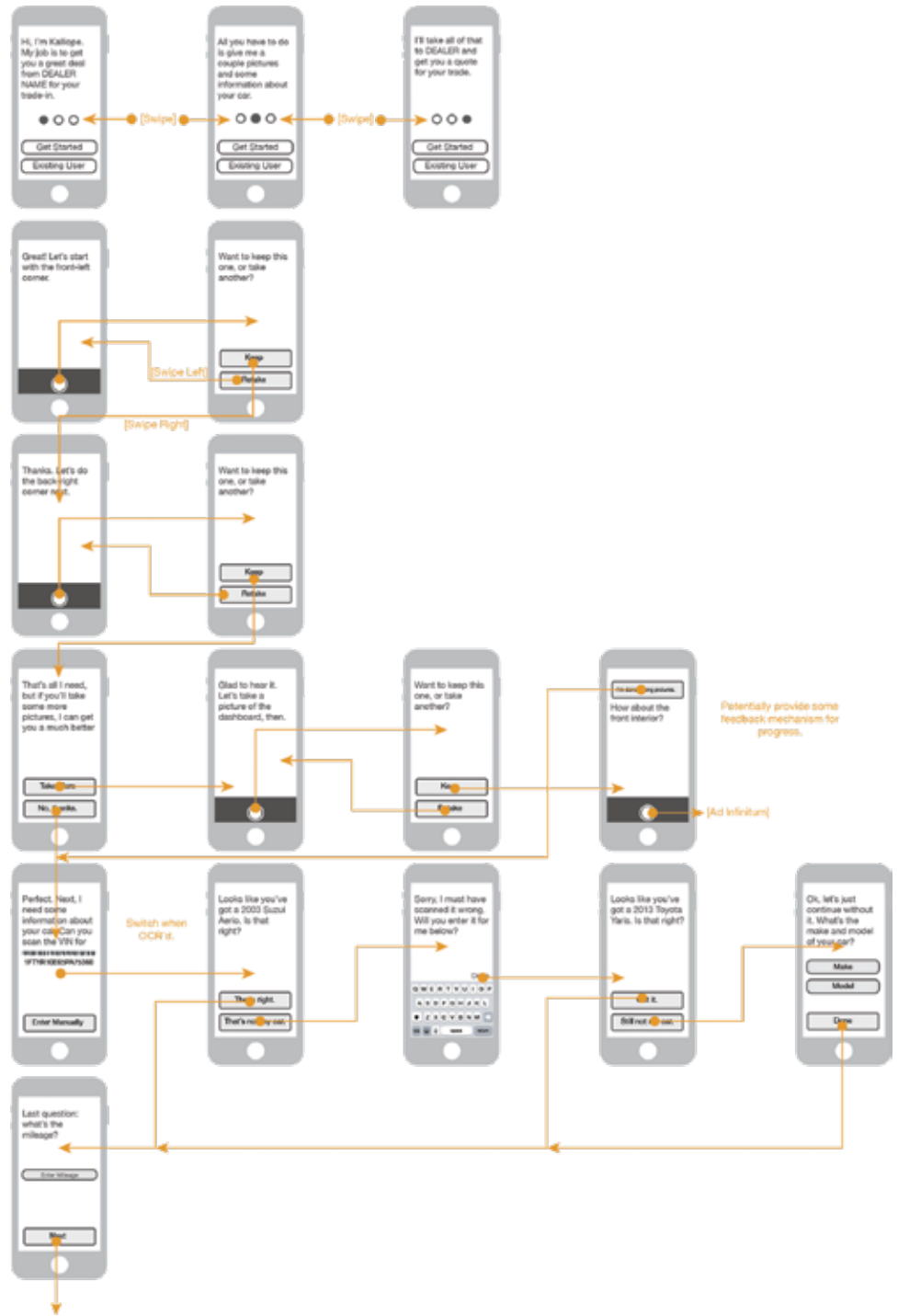
WIREFRAMES

Create a flow chart showing progress through the app that follows the user story. Think about every possible path. Represent each page with a rough sketch that indicates navigation, buttons, and major content elements. Don't be designing yet.

The ideal number of menu items is 7 or less. Each menu item can have a dropdown menu if necessary. In other words, find a way to group all of your pages of information into 7 or fewer top-level topics.

Avoid a laborious initial onboarding process unless absolutely necessary. You can gather more account information later.

For a large website, you might also have a site map with a hierarchical listing of every page that will be created.



This is a portion of a wireframe diagram for a complex app that estimated the trade-in value of a used car based on pictures the user snapped. Wireframes can be sketched in pencil on paper or any graphics program you are comfortable with. You don't need to use a full-featured wireframe application.

Elements of Style



These are the main components of the visual aspect of the user interface.

COLOR PALETTE

Based on your user profile, what colors are most appealing? Hot, saturated colors generate excitement, and cool, subdued colors are calming. Many resources exist to help you come up with a pleasing and professional palette.

Put each color name and hex code in a comment at the top of your style sheet for reference. With **CSS Custom Properties** you can now easily refer to colors by name in your CSS.

A color palette needs at least the following:

- Dominant color (will very likely come from the logo)
- Three complementary colors (to the dominant color)
- Dark neutral
- Light neutral

Build your color palette from that base by adding tints and shades of the main colors, and adding more colors as needed, dictated by your content. If you need ad hoc colors, make sure they are compatible with the defined color palette. Add the new colors to the style guide for reference.



A very basic cool color palette.



A very basic warm color palette.

Color Palette



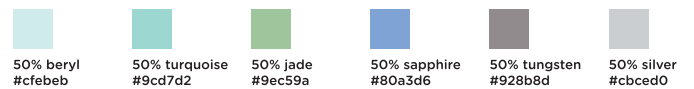
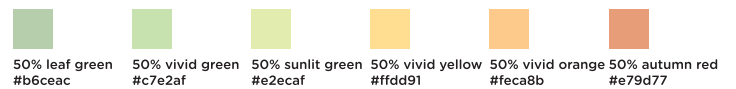
On the left is a color palette developed for a business application. In the top row are the colors for most of the interface. In the second row are the accent colors used mostly in charts and graphs. The last row contains tints of the accent colors where more variety is needed.

On the right is a color palette for a data-driven city transportation dashboard. Different types of data are color coded with hues that have the most differentiation. The other colors are used for additional features. The tints are used for hover states and backgrounds. The more complex the app, the more colors you will need.

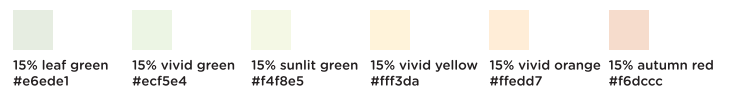
VIVID COLOR PALETTE



50% TINTS



15% TINTS



TYPOGRAPHY

Typefaces have personalities. What kind of personality do you want your website or app to have? Programmers tend to limit themselves to the fonts on their system. 2way subscribes to two type houses that give us thousands of quality web fonts to choose from. Google Fonts has some nice font families—and they're free.

See
Typography
Tips starting
on page 18 for
professional
guidance.

In general, serif typefaces are traditional and more credible. Sans serifs are more modern and easier to read in small sizes. Script type should be used sparingly to add flair. Handwritten fonts give a personal touch.

Choose one typeface (font family) as your base. Optionally, add one complementary typeface for a richer and more complex look.

traditional *add flair*
CREDIBLE
modern small sizes
personal touch

A Google Fonts typeface with 4 fonts.

Lora Regular

Lora Italic

Lora Bold

Lora Bold Italic

Start with one typeface as a base, and add a complementary typeface for richness.

About Us

Lorem ipsum dolor sit amet

Consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui.

Dolore te feugait nulla facilisi

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Sed diam nonummy nibh euismod tincidunt

Laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut

ICONS

Gather an icon set for your app and make sure that they are all the same style. For many apps, the FontAwesome icons are sufficient. For more individuality, icon sets can be purchased for a small fee on stock image sites. The vector files can be customized to the requirements of your interface. Additional icons can be drawn in the same style. If using line icons, be sure that the line weights are the same throughout the set, especially if some are resized.

Beware of free icons. They are the lowest quality. Not only that, you will see them everywhere because they are free.

Exporting each icon to svg is the best way to use icons in your app. They are scalable and can take on CSS styles.

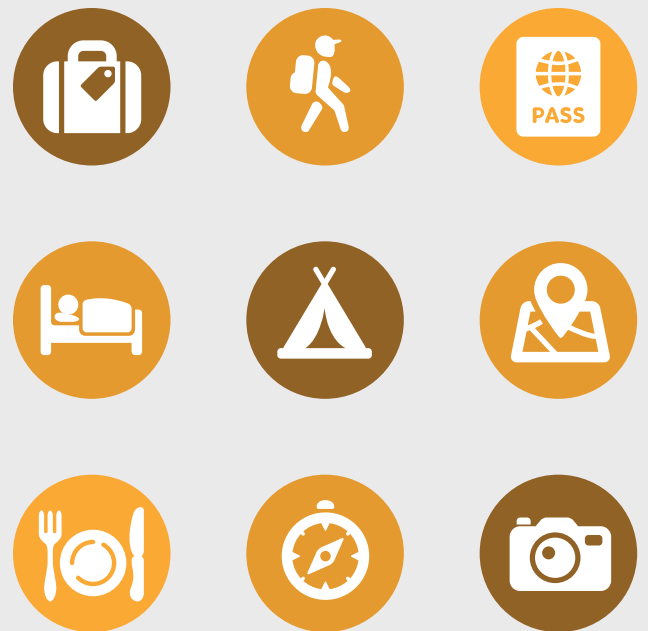
FontAwesome is sufficient for a lot of applications.



If you are using line icons, make sure that all are the same weight and all have either square or round corners and terminals.



Maintain consistency throughout your icon set.



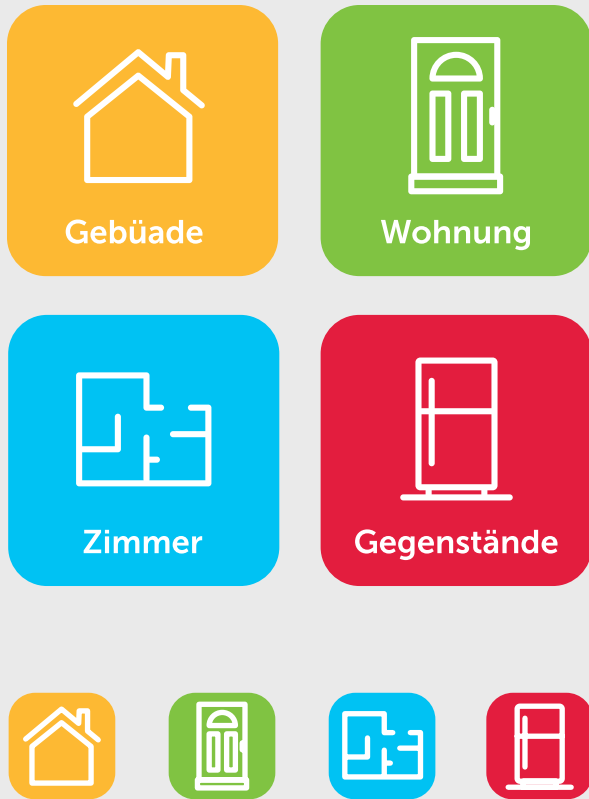
BUTTONS

For mobile devices, a button should be at least 40 x 40 pixels. Will your buttons be squares, rectangles, or circles? Will they have rounded corners? Will they be capsules? Define consistent button styles in your CSS. Also define hover states (for web).

Buttons that do different things should look different. For example, a navigation button should look different from a form submit button. It can be a subtle difference, like a size or color change. (See *Develop a Visual Language* on page 11.)

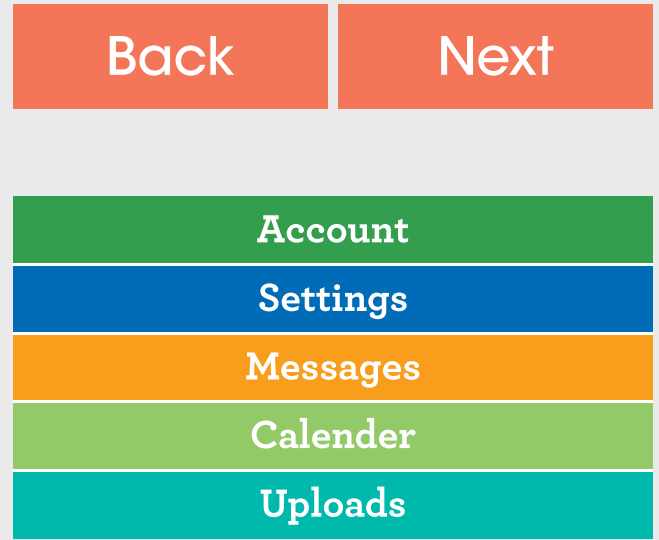
Use an icon or text on your buttons, but not both. If your icons are not perfectly clear, then opt for text. One exception is shown below. The first time the user sees the buttons it is with text. Thereafter it is the icon only.

The user is introduced to these buttons on the home screen with color, icon, and text. Within the app, text labels are dropped.



Keep capitalization consistent throughout the app. Will your buttons be all caps, initial cap, title case, or all lower case?

Rectangle buttons with square corners.



Rectangle buttons with round corners.



Capsule buttons.



Circle buttons with icons only.



PHOTOS

Choose photos carefully. Poor photo choices degrade the appearance of quality. Be aware of photo composition. Sometimes cropping helps the composition enormously. Crop out background distractions that draw attention away from the subject.

As with icons, free photos tend to be clichéd. They are the bottom of the barrel. If your app is released to the public, make sure you have the copyright to every photo.

A home page hero image will be about 1800 pixels wide.

If the client is providing photos, strongly suggest they use a professional photographer. Give them minimum pixel dimensions of 2000 x 1200. Start with the high-res photo and adjust the brightness, contrast, highlights, shadows, and color balance. Use the Photoshop rubber stamp tool or spot healing brush to remove flaws. Use the gradient tool to add or intensify blue in an overcast sky.

Save the photo in a nondestructive format such as psd or tif. Then adjust the pixel dimensions for its final size and save it as jpg. You can always go back to the psd or tiff file to make further adjustments, then resize as resave as jpg.

Whatever you do, please don't stretch or squish photos! Maintain proportions.



The photo on the left is straight from the camera. Although a beautiful scene, it's rather dark and overcast.

Below, I have cropped out the dead space above and below, brightened the colors, and added blue to the sky. This is a more attractive banner image.



Interface Design

PUTTING THE PIECES TOGETHER

What kind of home page will be most inviting? What implicit or explicit promises will be made visually and verbally? As always, keep the user story in mind as you begin the page mockups.

First, determine what screen size will be used most often: desktop, tablet, or phone. Start your design with the most likely pixel width.

Most pages will have a header, body, and footer. You might also have sidebars, preheaders, prefooters, etc. Be clear about what areas you need to design before starting.

DEVELOP A VISUAL LANGUAGE

Consistency is key in web and app design. Navigation in particular should remain stable throughout the experience. Think about how the navigation will look in various widths.

I find it helpful to divide the page into columns. Sometimes only two columns are needed, but you might need as many as 12. A column grid layout brings underlying consistency to all pages in your app or website. It also helps conversion to different viewport sizes. Define a max width for your content area. Most of the time, around 960px creates the optimal line length.

Use visual cues to guide the user down the most productive path. For example, one, two, or three large colorful buttons that force a preliminary choice and invite a first interaction.

Use color consistently. For example, all links should be the same color. Different sections can be color coded as a navigation aid.

Take advantage of CSS to create styles that can be reused and easily changed globally when the inevitable revisions come in. Do not create styles ad hoc or, heaven forbid, inline. Inline styling is only for one off exceptions or overrides. If you use a style more than once, add it to the style sheet.

STYLE GUIDES

The style guide consists of the logo, color palette, typography specs, icon set, and specs for all repeated visual styles. It might contain image guidelines and supplemental graphics.

On the next few pages are examples of actual style guides that I have developed for my programming team. You can see how the elements are brought together to support a look and feel.

BRC Style Guide



This style guide has the basic elements on one page: colors, typography, buttons, and icons.

H1 25/28 Tungsten Semibold, dark blue, with .1 em letterspacing

H2 16/20 Gotham Bold, white

Large Text 14/20 Gotham Medium cool gray 11

Small Text 12/16 Gotham Medium, cool gray 11

Buttons: 30px high, 15px rounded corners, width adjusted to fit text

not ready

Calculate Now

ready

Calculate Now

hover/active

Calculate Now

button-text: 20pt Tungsten Semibold, with .1em letterspacing





This style guide has the basic elements on one page: logo, colors, typography, and buttons. A second page contains the icon set and additional type specs.



NEIGHBORHOOD
jade
#339F4E



BUILDING
sapphire
#006CB7



DWELLING
vivid orange
#F99D1C



ROOM
vivid green
#93CA68



OBJECT
turquoise
#00B9AD



REMINDER
vivid yellow
#FFC20E



tungsten
#49494C



silver
#A4ABAF



leaf green
#73AB6B



sunlit green
#C8DC62



beryl
#A2DADB



autumn red
#CF381E

h3 **14px Archer 700**
margin-top = 20px
margin-bottom = 10px

p 14px Archer 500
line-height = 18px
margin-bottom = 10px
color = tungsten
align = left



button is 20px high and 100px wide
10px rounded corners
button-text = 14px Archer 700; white;
all caps; letter-spacing = .1 em
background = tungsten; hover = silver

↑ in general, 10px between elements

property

“hint” = 14px Archer 500 Italic; silver
box = 30px high; 1px silver border
rounded corners = 5px



each bar is 30px high, 100% width

h2 **18px Archer 700**
color = white
align = center

border-bottom = 1px white solid

truncate overflow



CICADA

Conservation Inclusive Construction
and Development Archive

COLOR PALETTE



Almost Black
0 60 100 85
74 29 0
#4a1d00



Brown
0 60 100 65
115 57 0
#733900



Autumn
34 0 100 60
86 108 17
#566c11



Summer
55 0 100 25
98 155 51
#629b33



Spring
29 0 100 10
175 196 42
#afc42a



Sunshine
0 24 94 0
255 196 37
#ffc425



Pumpkin
0 66 100 7
227 111 30
#e36f1e



Cranberry
0 100 34 40
158 0 69
#9e0045



Sky
68 34 0 0
80 145 205
#5091cd



Cloud
6 3 0 1
232 236 245
#e8ecf5



Residential



Residential



Commercial



Commercial



Industrial



Industrial



About

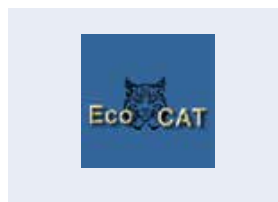


About

.small-button
20px Good Karma Wide
line-height = 24px
color = white



.social-button



.gem

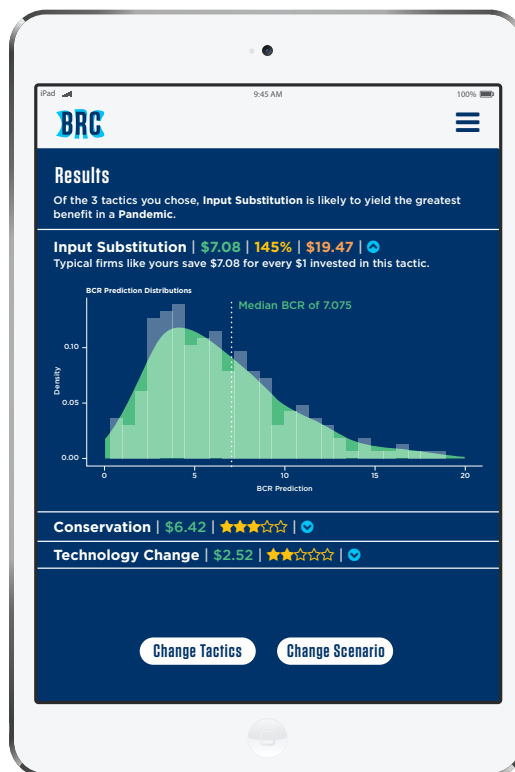
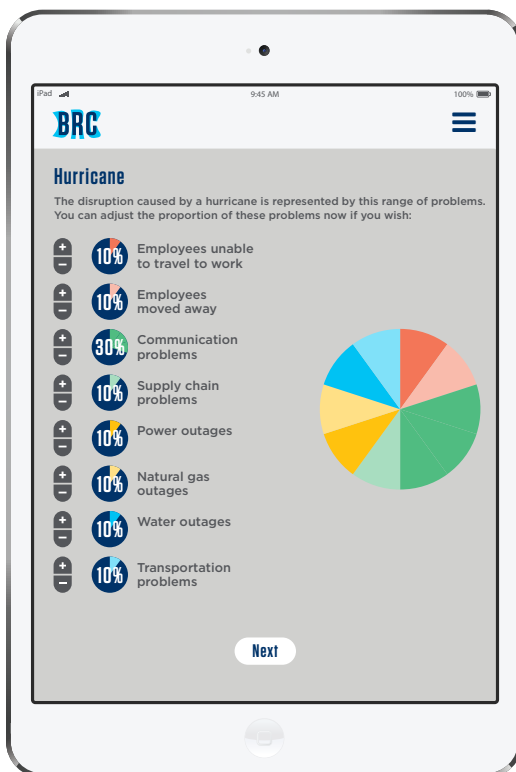
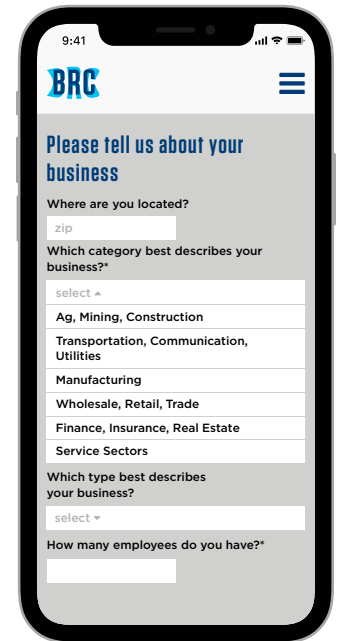
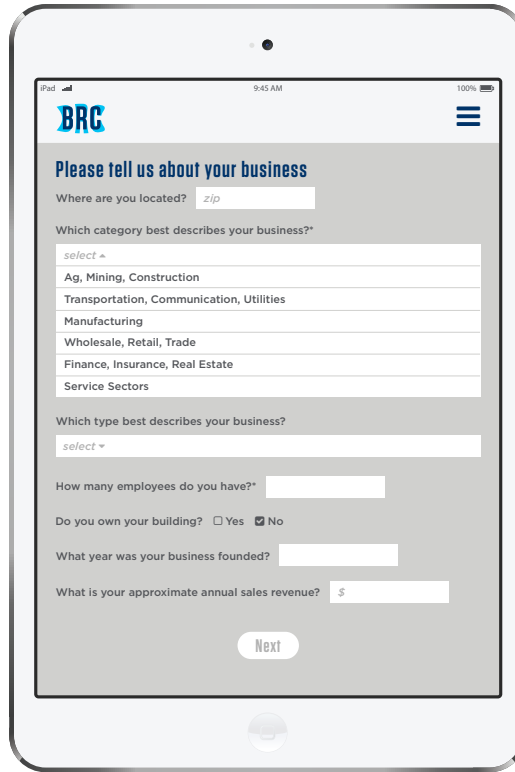
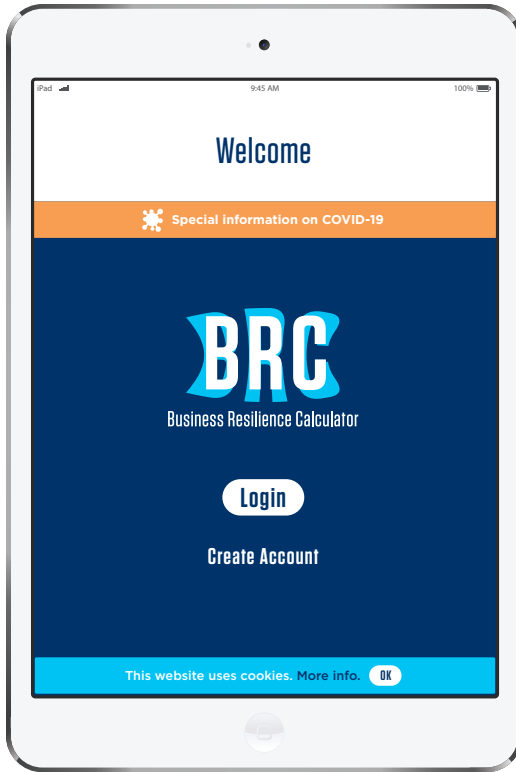
This style guide has the basic elements on one page: logo, colors, buttons, and icons. A second page contains the detailed type specs.

KEYFRAMES

Not every page needs a mockup. I mock up keyframes and use repeating styles for the rest. When I come across a page that has new elements, I mock that up.

Once you are happy with the page layout, redo it in another page width to make sure it still works. For example, if you designed for mobile first, make sure it can be adjusted for desktop without looking terrible.

This project was designed for a tablet, but I made sure it looks just as good on a phone.



Design Tips

Below are my tips for a clean, modern interface design.

Do not outline shapes

If you can't see something without an outline, increase the contrast instead. If you outline a box or shape, have a very good reason. Putting an outline on everything looks defaulty. If you have proper contrast throughout your page, outlining is unnecessary and only adds clutter.

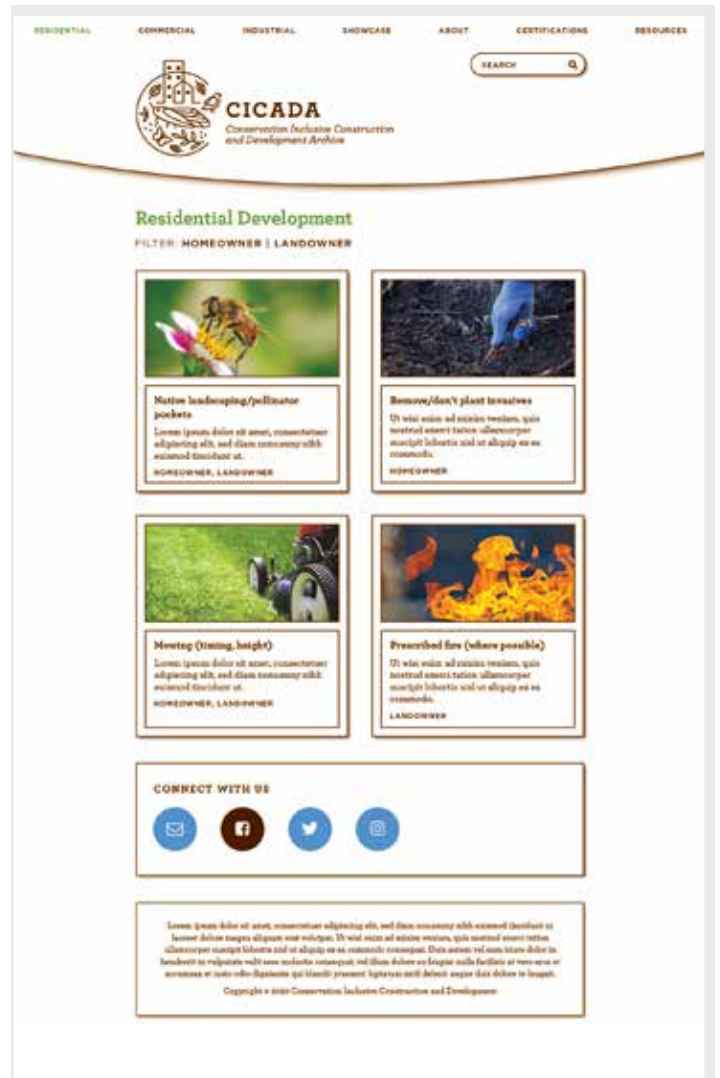
Use background colors (not outlines) to divide the page into areas. A subtle color change will keep areas more connected. A dramatic color change will indicate the section is less connected. Background colors should fill the width, even when the content is boxed. (See next page for example.)

Do not nest boxes

Boxes inside of boxes creates unnecessary complexity. Only box something if it really needs to be set apart in that way.

Use drop shadows sparingly

Sometimes a little drop shadow adds dimension. I keep my shadows at 15–30%. A heavy shadow is overbearing and draws attention away from the important content.



BEFORE: In the example above, every space is defined by an outline and a drop shadow. It has a cluttered feeling. (See next page for the makeover.)



AFTER: Color is used to define areas. The overall feeling is calm and open. Emphasis is on the content.

Spacing

Keeping in mind the law of proximity, things that belong together should have a consistent amount of space between them. Keep things that don't belong together a consistent, but larger, amount of space apart.

For example, if things that are related are always 20px apart, the next section should be at least 40px away. Keeping these small and large spaces consistent makes it very clear to the user what belongs together.

Don't be afraid of negative space! It gives a viewer's eyes a place to rest. Do not feel you have to pack everything tightly or spread it out to fill the available space. Keep spacing between elements consistent, and let the negative space fall as it may.

The law of proximity: objects that are near to one another are perceived as belonging together.

Readability

The most important items should be large and high contrast. Keep the most valuable information "above the fold."

Avoid long blocks of explanatory text. Provide a link to it (or menu item) for when the user is ready for deeper information.

Top align by default

Any time you have columns or boxes or cells, top align the type within them. We read left to right, top to bottom, so top alignment helps the viewer follow content across the page.

Typography Tips

In the best case scenario, you will have a style guide and mockups from a professional UI designer as guidance, or you will have followed the process described in the front of this guide. Even the most thorough style guide will not anticipate every detail. Use the following rules when in doubt. As with all rules, there are exceptions—but leave those up to professional designers.

Use flush left alignment by default

If you adopt only one of these tips, this is it. Flush left alignment will immediately look neater and cleaner.

Paragraphs should left align under their headings. Lists should left align under the paragraph above. Customize your CSS to align lists neatly under the paragraph above instead of allowing the default indent.

If you center something or indent it, do it for a good reason. It makes sense to indent a sublist under the main list. Sometimes a block quote within a column is indented to set it apart.

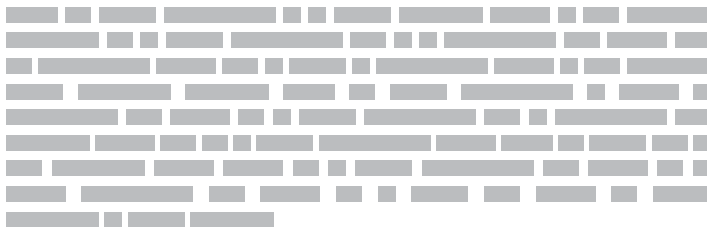
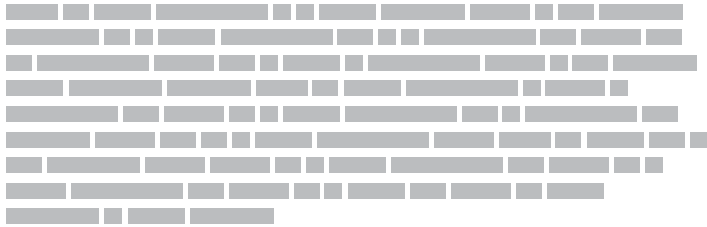
Center alignment works best on a single-column page, such as a home page or welcome page. In that case, everything should be centered on one axis. Nested columns on that type of page can also be centered.

LEFT: Everything is centered within its column, yet the header and footer have flush left and flush right alignments. The page has an overall chaotic feel. RIGHT: Everything is flush left. The page feels ordered, and your eye move more easily down the page.

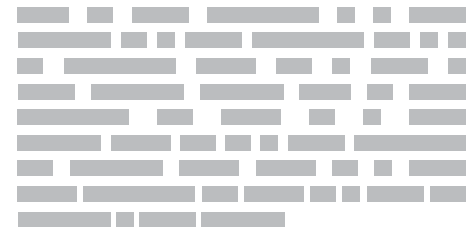
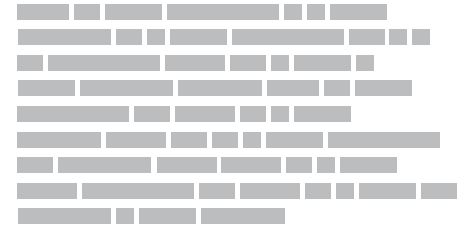


Do not justify text

Justified text requires a sophisticated compositor that adjusts word-spacing, letter-spacing, and hyphenation to create a nice, even texture throughout a paragraph. On a web page, justifying text simply adds word-spacing to force even lines. The extra space is unsightly and makes the paragraph harder to read. The problem gets worse with narrower columns.



LEFT: The top block represents flush left alignment. After it has been justified, notice how irregular the wordspacing is in the lower block. **RIGHT:** The problem becomes worse with a narrower column. Big gaps have formed.



Watch your line lengths

When your content div goes all the way across a page, lines of text can be hard to follow. The ideal line length is between 45 and 75 characters. Divide your text area into columns to achieve optimal line lengths. Or set a max-width on the div so that it can't increase to more than ~75 characters wide. For typical websites, that is somewhere between 780 and 960 pixels.

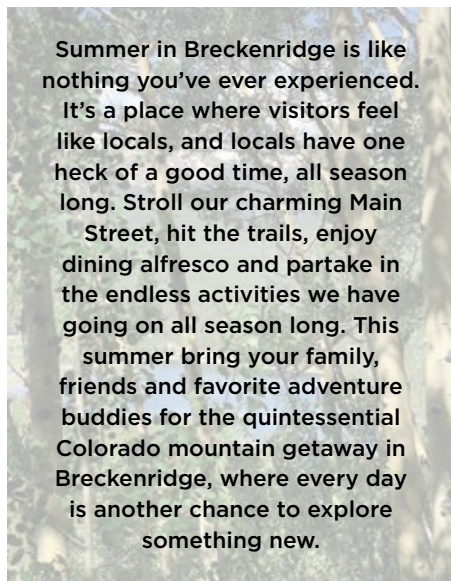
Notice how much harder it is to read this paragraph when it stretches all the way across the page. The more lines there are, the harder it is to follow. Set a max-width on your text divs so that they do not exceed a comfortable line length. Characters per line in this block comes to about 120. The rest of this paragraph is a copy of what you read above for comparison. When your content div goes all the way across a page, lines of text can be hard to follow. The ideal line length is between 45 and 75 characters. Divide your text area into narrower columns to achieve optimal line lengths. Or set a max-width on the div so that it can't increase to more than ~75 characters wide. For typical websites, that is somewhere between 780 and 960 pixels. Notice how much harder it is to read this paragraph when it stretches all the way across the page. The more lines there are, the harder it is to follow. Set a max-width on your text divs so that they do not exceed a comfortable line length. Characters per line in this block comes to about 120. The rest of this paragraph is a copy of what you read above for comparison. When your content div goes all the way across a page, lines of text can be hard to follow. The ideal line length is between 45 and 75 characters. Divide your text area into narrower columns to achieve optimal line lengths. Or set a max-width on the div so that it can't increase to more than ~75 characters wide. For typical websites, that is somewhere between 780 and 960 pixels.

Avoid type over a photo

It is rarely a good idea to put type over a photo. If you must, use a photo that has low-contrast space for type. Sometimes recropping a photo can open up the necessary space.

Another option is an overlay screen. A semitransparent color laid on top of a photo can reduce contrast enough for type to be readable.

Lightening (or ghosting) a photo as a full page background sounds like a good idea, but it rarely is. The viewer will either not be able to see the photo or not be able to read the type. One has to be sacrificed to the other.



Do not ghost a photo as a full page background. No matter how good an idea it seems, it does not work.



It's hard to read the white type over the clouds.



Recropping the photo is all it takes.



If recropping is not possible, drop a transparent color over the photo. In this case, I used 25% black set to "multiply."

Never outline type

Outlining type distorts the letterforms. If you can't read something over the background image or color, increase contrast instead of outlining the type. (See previous page.)

Outlined type is an amateur move.
Don't do it.



Use more space above a head than below it

Objects should always be closer to related objects than unrelated objects. (See the law of proximity on page 17.) The space above a heading should be twice the space below. As a rule of thumb, margin-top = 1 em and margin-bottom = .5 em. Space after paragraphs and lists should also be adjusted to .5 em.



LEFT: This page of text has equal spacing throughout—1em above and below.

RIGHT: This page of text has been adjusted so that there is .5em after each paragraph and heading, and a full em above each heading. Now you can see three distinct blocks. Each heading is grouped with its text.

Avoid tables

Tables don't work well in responsive layouts. Running information together in one line will adjust effortlessly to any viewport width. Put separators between data types, such as a bullet or a vertical bar. Using a different color for the separator is nice touch.

If a tabular format is required for your content, use left alignment as much as possible. Keep the table itself simple. Do not have double outlines around each cell.

City	Date	Address
Palatine	Nov. 18 (Mon.)	Harper College, Wojcik Conference Center 1200 West Algonquin Road
Palos Hills	Nov. 19 (Tues.)	Moraine Valley Community College, Building M Moraine Business and Conference Center 9000 W. College Parkway
Quincy	Oct. 24 (Thurs.)	Town & Country Inn and Suites 110 N. 54th St.
Rockford	Nov. 7 (Thurs.)	Giovanni's Restaurant & Convention Center 610 N. Bell School Road
Skokie	Nov. 21 (Thurs.)	Holiday Inn Chicago North Shore 5300 Touhy Ave.
Springfield	Nov. 21 (Thurs.)	Crowne Plaza Hotel 3000 S. Dirksen Parkway
St. Charles	Nov. 12 (Tues.)	Pheasant Run Resort 4051 E. Main St.
Tinley Park	Dec. 3 (Tues.)	Tinley Park Convention Center 18451 Convention Center Drive

ABOVE LEFT: Looks great on a full-size screen. ABOVE RIGHT: Looks lousy on a phone.

BELOW LEFT: This run-in method looks great on a full-size screen. BELOW RIGHT: Also looks great on a phone!

City	Date	Address
Palatine	Nov. 18 (Mon.)	Harper College, Wojcik Conference Center 1200 West Algonquin Road
Palos Hills	Nov. 19 (Tues.)	Moraine Valley Community College, Building M Moraine Business and Conference Center 9000 W. College Parkway
Quincy	Oct. 24 (Thurs.)	Town & Country Inn and Suites 110 N. 54th St.
Rockford	Nov. 7 (Thurs.)	Giovanni's Restaurant & Convention Center 610 N. Bell School Road
Skokie	Nov. 21 (Thurs.)	Holiday Inn Chicago North Shore 5300 Touhy Ave.
Springfield	Nov. 21 (Thurs.)	Crowne Plaza Hotel 3000 S. Dirksen Parkway
St. Charles	Nov. 12 (Tues.)	Pheasant Run Resort 4051 E. Main St.
Tinley Park	Dec. 3 (Tues.)	Tinley Park Convention Center 18451 Convention Center Drive

City	Date	Address
Palatine	Nov. 18 (Mon.)	Harper College, Wojcik Conference Center 1200 West Algonquin Road
Palos Hills	Nov. 19 (Tues.)	Moraine Valley Community College, Building M Moraine Business and Conference Center 9000 W. College Parkway
Quincy	Oct. 24 (Thurs.)	Town & Country Inn and Suites 110 N. 54th St.
Rockford	Nov. 7 (Thurs.)	Giovanni's Restaurant & Convention Center 610 N. Bell School Road
Skokie	Nov. 21 (Thurs.)	Holiday Inn Chicago North Shore 5300 Touhy Ave.
Springfield	Nov. 21 (Thurs.)	Crowne Plaza Hotel 3000 S. Dirksen Parkway
St. Charles	Nov. 12 (Tues.)	Pheasant Run Resort 4051 E. Main St.
Tinley Park	Dec. 3 (Tues.)	Tinley Park Convention Center 18451 Convention Center Drive

City	Date	Address
Palatine	Nov. 18 (Mon.)	Harper College, Wojcik Conference Center 1200 West Algonquin Road
Palos Hills	Nov. 19 (Tues.)	Moraine Valley Community College, Building M Moraine Business and Conference Center 9000 W. College Parkway
Quincy	Oct. 24 (Thurs.)	Town & Country Inn and Suites 110 N. 54th St.
Rockford	Nov. 7 (Thurs.)	Giovanni's Restaurant & Convention Center 610 N. Bell School Road
Skokie	Nov. 21 (Thurs.)	Holiday Inn Chicago North Shore 5300 Touhy Ave.
Springfield	Nov. 21 (Thurs.)	Crowne Plaza Hotel 3000 S. Dirksen Parkway
St. Charles	Nov. 12 (Tues.)	Pheasant Run Resort 4051 E. Main St.
Tinley Park	Dec. 3 (Tues.)	Tinley Park Convention Center 18451 Convention Center Drive

HTML Superheroes

Hero	Power	Nemesis
The XMLator	Standards compliance	Sloppy Code Boy
Captain CSS	Superlayout	Lord Deprecated
Browser Woman	Megacompatibility	Ugly Code Monster

ABOVE: This default table looks terrible.

BELOW: Try a table style with cell borders that match the background color. Simpler is better.

HTML Superheroes

Hero	Power	Nemesis
The XMLator	Standards compliance	Sloppy Code Boy
Captain CSS	Superlayout	Lord Deprecated
Browser Woman	Megacompatibility	Ugly Code Monster

Only underline links

Users expect underlined text to link to something. Do not underline a head—or anything else—for emphasis. That being said, there are more beautiful ways to indicate links than underlining. A bold font with a distinct color is a nice option. Underlining on hover helps the user see a link before clicking; of course this doesn't work on mobile.

Adjust your line-heights

The bigger the type size, the smaller the line-height. For most headings, the line-height will equal the font-size. (Designers call this solid leading.) Don't always default to auto-leading.

On the web, paragraph text needs more leading than in print. A line-height of 150% is usually good.

All caps requires letter-spacing

Anything set in all caps requires about .1em of letter-spacing. That's because all caps has less differentiation between each letter. A capital letter has no ascenders and descenders, and takes up most of its horizontal space. Adding space between each letter allows them to breathe, and makes it easier for the user to take in.

All caps should not be used for any text that runs more than three lines. It becomes very hard to read. Notice that if someone does not want you to read a paragraph, it is set in all caps.

Use proper en and em dashes

Use an en dash to show a range, such as 8 am–5 pm. Use an em dash to show a break in thought—and no spaces around dashes, please.

Using a hyphen in place of a dash can lead to confusion. Good design is all about communicating with clarity.

Ways to add emphasis

- Make it bigger is only one way
- Increase contrast
- Make it bolder
- Change color
- Change font to italic
- Use a different typeface altogether
- Isolate it (surround with negative space)
- Change the shape—if everything is rectangular, a round shape will stand out
- Violate the grid

This is a Heading, Not a Link

I'm clicking on it but nothing happens!

This Large Head Requires a Line-Height Equal to Its Font-Size

THIS ALL CAPS HEAD REQUIRES .1EM OF LETTER-SPACING

All caps requires more line-height as well. You want the space between lines to be greater than or equal to the space between words.

Your eye keeps getting drawn here, doesn't it?

The Art of Delight

The most delightful experience is free of frustration. Beyond that are little touches that surprise a user and elevate the app to something more. A delightful app generates loyalty and word-of-mouth recommendations.


Extra touches are subtle and unexpected. It might be wording with a sense of humor. It might be saving the entries in a long form so they don't have to be retyped if submission fails. It might be a little animation upon completing a task.

Over the top animations or sounds or pop-ups are annoying! If you keep your user foremost in your mind, delightful touches will emerge naturally.



In this race app, Emma pops up to announce a new leader. Her eyes move ever so subtly.

Accessibility



Try to stick to WCAG 2.0 with minor exceptions. Total compliance doesn't always make sense. If the project contract requires compliance to stricter standards like WCAG 2.1, then compromises will need to be made to the design.

There you have it, a designer's guide for programmers. I'm sure I missed a few things, but you now have plenty of tips to start implementing right away.

If this guide seems overwhelming, it has given you an appreciation for the job that designers do.

This will be a living document. I'd love to hear from you if you have feedback or suggestions.

Gretchen@2wav.com

We Are 2wav



Anderson

Boss

Software architect, team leader, code craftsman. I have delivered projects in military intelligence, video processing, eCommerce, mapping, and mobile computing. My years as an entrepreneur, employer, and consultant have honed my instincts and understanding of stakeholders' needs. In my occasional time away from software, you will find me outdoors: whitewater kayaking, rock climbing, and running mountain marathons.



Joseph

Senior Software Engineer

My initial experience was developing 3D games in DirectX/OpenGL and I specialized in graphics and advanced shader systems along with artificial intelligence. Over the past 3 years I've focused on mobile platforms ranging from the iPhone/iPad to ARM boards, as well as database management and web design. I've also worked on mobile game development on both the iOS and Android systems.



Gretchen

Creative Director

I am a graphic designer that enjoys problem solving, strategizing, and organizing complex concepts. As creative director at 2wav, I design interfaces that are both appealing and logical. My favorite part of the process is seeing the programmers bring my ideas to life on screen. When not in the studio, I find my peace and inspiration hiking in the mountains.



Matthew

Senior Software Engineer

I am interested in software craftsmanship, TDD, building information systems on the web and exploring all kinds of software. I work with many different technology stacks on the web including PHP, ASP.NET, Rails, Node.js and Meteor. I am big fan of the text editor Vim. I like to read and write when I have some spare time.



Aleksandra

Developer

I am an energetic, passionate web developer who is always hungry to learn. I love contemporary layouts and captivating responsive design. Some of my favorite things to explore are different technology stacks and css techniques. I enjoy reading, going to the beach, and hanging out with my two adorable dogs.



Michael

Human Relations Director

Being able to see our customer's ideas and vision come to fruition is what inspires me. I work tirelessly to bridge communication from vision, through development, all the way to delivery. I am a lover of nature, enjoy gardening, hiking, and living a healthy lifestyle.



André

DevOps

I do software—whether that software is the hypervisor that sits on your bare metal server, shell scripts to manage that server, an open source application that helps to manage a team, or custom code developed from scratch. I currently specialize in DevOps, which harnesses my development skills to solve infrastructure operations problems. When I'm not code bending, I can be found under some weights in the gym, dancing salsa, or reading a science fiction novel.

We are passionate about the craft of software and pour that energy into every project. With emphasis on real-time mobile web, ontology based information systems, and product development for the Internet of Things, we deliver practical and elegant solutions.

About Us

We are more than software developers—we are a team of innovators, strategists, and partners at every step. We help identify the best target markets, design a powerful brand identity, and ensure top-notch user experience while integrating with our partners' internal teams during and after development to ensure a seamless product lifecycle.

Contact us at
info@2wav.com

For more than 20 years, Anderson Wiese and the 2wav professionals have been delivering diverse projects to clients across North America and internationally. We are based in Manitou Springs, Colorado, between Pike's Peak and the city of Colorado Springs. Our team members live in Georgia, Illinois, California, and Toronto, Ontario.

While there is no magic recipe, we've got a lot of experience with things that work—and things that don't. We use this experience to help guide our customers' decision making, crafting products that have the greatest chance of success.

2wav is Your Partner

The 2wav team fully invests its energy in every project. Some of our clients have been working with us for over a decade. We even support a product whose company went out of business years ago. Your trust in us is taken seriously.

With over 20 years of experience working mostly with small to medium sized businesses, and a lot of startups, we have learned valuable lessons. Our practical business advice is shared freely as we consult with our clients on their software project.

Join Us

We are looking for interesting projects that fit our skill set and challenge us to grow. See our portfolio at **2wav.com**.

2wāv

