

# USING ACCESSIBILITY AS A BRIDGE BETWEEN DESIGNERS AND ENGINEERS

Abstractions · Pittsburgh, PA · August 22, 2019





Hi! 🖐️

I'm **Damien**.

I am a queer **designer who codes**,  
specialised in **accessibility**.

I work for **Castor EDC** in Amsterdam, NL  
as a **Design systems & Accessibility Lead**.

I use **they/them** pronouns.



# Content warnings ⚠️

I am known to use a colourful language,  
so please pardon my French! 🙊

There will be videos (with sound). 🎹

Usually, my sense of time is really bad. ⌚

And also, raccoons? 🦨



A bit of  
**background** first







**I built my first website in 2004**



**I sold my first web project and worked as a freelancer since 2012**



**I designed & developed a SaaS product between 2014 and 2015**

for a political startup helping local leftist candidate to get elected



**I worked equally as a designer and as an engineer since**



**I moved to the Netherlands as an engineer end of 2016**



**I decided to focus only on design and stop shipping code in 2017**



**I changed my mind at least 10 times since**





**And buckle up, it's getting worse 🙌**







I joined my current job as Product Designer, in the product team



~~I complained a lot about~~ “*raised issues*” regarding the HTML/CSS and the lack of accessibility of our product



I became an IAAP certified Web Accessibility Specialist

take this in the face impostor syndrome 🐊



I started fixing the low-hanging fruits in the code base



I am now building and leading the design system for our products 😅





**In the end, there is only one thing I'm sure about:  
I've always struggled to **define what my job is.****





**And you know what?**  
**I am not the only one.**



But, **what's the point** with today's talk?





Design & Engineering:  
how is it going?



**I am about to show you the footage  
of a designer trying to interact with engineers.**

**No engineers were harmed in the making of this recording.**









Our industry **accepted people working in silos**  
making the collaboration harder.





Our industry accepted a situation where designers and engineers started **speaking two different languages.**



**The languages we speak  
are shaping our perception of reality.**





**People can't collaborate efficiently  
if they are not sharing a common language.**



**People can't collaborate efficiently  
if they are not sharing a common ~~language~~.  
reality.**



It is common for **growing companies** to go through a **storming phase** where **teams are isolating**.





It is common for **growing companies** to lack alignment between product & engineering teams.



But also, our industry accepted a situation where “*It is not your job*” can be an answer.



# Design vs. Engineering

## Is there really a winner?









How do we define  
what is our job?



**Should designers code?**  
**Should engineers design?**



Stop.





~~Should designers code?~~  
~~Should engineers design?~~  
**Who cares?**



**Engineers are not defined by their code.  
Designers are not defined by their sketches.**



Our job is to **deliver value to our customers.**

Our job is to **protect our users** and their data.

And if this is done, to **make our products successful.**





Eh! It was a talk about  
**accessibility**, right?.



# Do you think accessibility is important?



**Yes.**



Let's be clear, the only possible answer is **yes**.





**How is it possible that so many people care about it,  
when most of the websites are still excluding users?**



**You can find all needed information online.  
There is courses, tutorials, articles, guides.**



**Being accessible is not taking more time.**  
**Fixing accessibility later on is what's costing a lot.**



**But in the end,  
we are still not able to do even the bare minimum.**





**Collaboration** is the key to  
accessible products.



**A fully accessible design will exclude people  
if not implemented correctly.**



**A fully accessible interface will exclude people  
if not designed correctly.**



**Remember the thing about languages and reality?**  
**Accessibility is an answer here.**



**Accessibility is the common language that can help designers and engineers to **work together** (again).**





It requires both **involvement** from scratch.

It creates a space for people to **speak together**.

It pushes people to **take ownership & make decisions**.

It **empowers people** to share their opinions.



Let's build a **bridge**.







**Yay! Finally, a raccoon!**

Abstractions • August 2019 • @iamhiwelo



1.

We need to start  
with **strong foundations**



**Accessibility needs to be part  
of your **teams' incentive system.****





**Accessibility is not an acceptance criteria.**  
**You can't test 61 WCAG success criteria each time.**



You need to find people caring or being  
knowledgable that can take ownership of this topic.



**For that, you need a culture allowing and  
pushing people to take ownership of their work.**



**Improving accessibility is the perfect opportunity  
to take a step back and look at the bigger picture.**





Identify **accessibility champions** in your teams



Add accessibility and inclusivity in your **incentive system**



Offer access to **courses and resources** for your teammates



**Empower people to share their knowledge** cross-team



Promote accessibility by **using assistive technologies in demos**



Start with a quick, informal and **internal audit** of your products





# Dear managers, the next one is just for you.



Accessibility is not an option. 🦊

People should not fight for it. 🦊

It's your job to be a shield for your team members. 🛡️



**Also, accessibility is a **never ending effort**.**  
**It needs a lot of discussions, planning & alignment.**  
**It is **never done**.**



2.

We will need  
a **good tooling** to help us



Getting feedback is always easier  
when you get it from an **automated tool**.





Implementing a strong tooling system  
will help preventing any knowledge-related stress.



**Automated tools are limited,  
they only catch 20-35% of potential issues.**



But in the end, with or without tooling,  
the best thing to do is always to **test your work**.



Testing **keyboard navigation** and existing **labels** for **assistive technologies** should be part of a PR review.



- icon
- Link
- Radio Option
- Slider
  - overview
- Text Input
- Validation Messages

3. MOLECULES

- Input Group
- Radio Group

## Component checklist

### Quality Assurance

- ☐ It is having at least one snapshot test
- ☐ It is having at least a test with `jest-axe`
- ☐ It is having a series of relevant tests

### Accessibility manual tests

- ☐ Support [keyboard navigation](#)
- ☐ Use [optimal HTML landmarks](#)
- ☐ It was tested with [Accessibility Insights for Web](#)
- ☐ It was tested with at least one assistive technology
- ☐ It seems to be compliant with [WCAG 2.1 AA](#).

### Documentation

- ☐ At least one overview story exists in Storybook

### Other

Relates to [#22](#).



# Checklists in pull requests for the win!

Abstractions · August 2019 · @iamhiwelo

## Quality Assurance

- ☐ It is having at least one snapshot test
- ☐ It is having at least a test with `jest-axe`
- ☐ It is having a series of relevant tests

## Accessibility manual tests

- ☐ Support [keyboard navigation](#)
- ☐ Use [optimal HTML landmarks](#)
- ☐ It was tested with [Accessibility Insights for Web](#)
- ☐ It was tested with at least one assistive technology
- ☐ It seems to be compliant with [WCAG 2.1 AA](#).

## Documentation

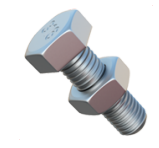
- ☐ At least one overview story exists in Storybook







**Set a series of automated tools** to avoid human focus only on code



**Add accessibility in your unit/functional tests** (Jest DOM, Jest-aXe, etc.)



**Promote the usage of browser plugins** (like Accessibility Insights for Web)



**Add assistive technologies in your device lab**



**Add keyboard navigation testing** in your daily workflow



**Include users with impairments** during user testing sessions



**Empower people** with checklists and guidelines



# Accessibility Guidelines

Making work accessible creates a better experience across the board. Use this checklist to help build accessibility into your process no matter your role or stage in a project.

## The Checklist

**Check** the boxes of the guidelines that apply to your project

**Preview and copy** the checklist to your clipboard

**Paste** the checklist into a document, Slack, Trello—or however else your team organizes projects

DESIGNERS

ENGINEERS

PROJECT MANAGERS

QA

EDITORIAL

### Designers

☐ Make sure there is enough contrast between text and its



Having checklist is great.  
(<https://accessibility.voxmedia.com/>)

## Designers

- ☐ Make sure there is enough contrast between text and its background color

According to the WCAG, the contrast ratio between text and the text's background should be at least 4.5 to 1. If your font is at least 24 px or 19 px bold, the minimum drops to 3 to 1 (though, worth noting that this is a little fuzzy because numbered sizes aren't always reflective of the visual size type). Be especially careful of text over images.

Source: Salesforce UX

- ☐ Don't indicate important information using color alone

There should be another indicator (such as icons to accompany color coding, or an underline on linked text) so that people who cannot easily distinguish colors will be able to understand and use your content.

Source: UCLA

- ☐ Pair values of colors together (not only hues) to increase contrast

When you strip out the color information from two hues and compare the values, there may not be much difference. Pairing different values creates contrast.

Source: UCLA

- ☐ Don't rely on sensory characteristics as the sole indicator for understanding and operating content

You should not rely solely on images, shape, size, visual location, orientation, or sound to indicate important instructions for operating or understanding content (ex. "See the image above"). Instead, use a combination of positioning, color, and labeling to identify content.

Source: W3C



And, y'know, it is useful for everybody.  
(<https://accessibility.voxmedia.com/>)



# Engineers

## ☐ Use the correct HTML element for your content

HTML elements communicate to the browser what kind of content they contain and how the browser should render or treat that content. The [accessibility tree](#) is built off of assumptions about the elements and their structure. This is the browser feature which powers screen readers. Choosing the correct element for the current context is a simple way to create a good foundation for your experiences.

Source: [WebAIM](#)

## ☐ Support keyboard navigation

Browsers support tabbing through link, form, button elements. This is an easy way to move around the page, but it's very easy to accidentally hinder this functionality: by relying on CSS to move elements around instead of actually reordering the HTML; hiding elements from tab flow, either by faking buttons (with Javascript) in lieu of using button elements or disabling tab via `tabindex="-1"`; and hiding form elements (on a tabbed interface) but not removing them from the tab flow (so that hidden form elements can be focused).

Source: [WebAIM](#)

## ☐ Understand and use HTML landmarks

People who use assistive devices may not want to peruse your content linearly, instead preferring to use an outline view (generated from markup) to find the relevant content for their current needs. These landmarks, like main and form and navigation help define specific areas that a user might want to seek out and can save them a lot of time.

Source: [A List Apart](#)

## ☐ Write good alt text for your images

Users with low vision often make use of a talking browser to "read" the web. These specialized browsers convert text to speech so that a user can hear the words on a site. When a talking



You can help people with resources, warnings, etc.  
(<https://accessibility.voxmedia.com/>)

## Project Managers

- ☐ Familiarize yourself with the work associated with making content accessible

We have guidelines for [design](#), [engineering](#), and [QA](#) testing. Read through the requirements, and be aware of the checks and standards that QA will be looking out for in your products.

- ☐ Build in time for accessibility during project planning and sprint planning

Accessibility is not a checklist item that only needs to be considered in some projects, or at the end of a process. These practices should be woven into every step of a project and role in a team. Make sure you are accounting for that while planning projects and sprints.

- ☐ When sharing good work done by your team, praise efforts to increase accessibility

We all like to hear our work recognized. In addition to celebrating beautiful designs, elegant code, fast turnarounds, stellar experiences—draw attention to and praise the work done by your team to make content more accessible. Talk up high contrast, a smart keyboard navigation system, or clear copy and hierarchy.

- ☐ The tools and products that you create should make accessibility easier to achieve

Accessibility should be at the forefront of your mind when considering which projects to take on and how to shape the products. Whatever you create should support accessibility efforts, and not hinder them. For instance, if your team creates a publishing tool that allows users to upload images, and the tool does not allow for alternate text descriptions, you have made it



It helps ICs but it is also a great political tool.  
(<https://accessibility.voxmedia.com/>)



# Editorial

## ☐ Write good alt text for your images

Non-sighted users often make use of a talking browser to “read” the web. These specialized browsers convert text to speech so that a user can hear the words on a site. When a talking browser lands on an image, it looks for alt text that it can read aloud; if it finds none, it will often just say “image,” leaving the user in the dark as to what the image is and how it matters to the story.

Describe all of the elements that explain what’s happening in the image, rather than just setting the alt text to be something like, “photograph”

If you have to use an image of text, be sure to describe the design if relevant, as well as all of the words in the image. (Ex: Whiteboard drawing of the quote “This is a quote”)

Source: [WebAim: The Basics of Good Alt Text](#)

## ☐ Editorial Engineers: think about how custom experiences can be made accessible

For instance, SVGs are often used as icons on interactive elements in sites. When this is the case, the SVG should have a title or embedded text (that is visually hidden but available to assistive tech) to ensure that the information of the icon is available programmatically.

Sometimes you don’t have time or resources to make certain components into accessible experiences. In those cases, consider creating an additional screen reader-friendly experience, or at the very least, describe the experience to users in alt text.

## ☐ Be as consistent and clear as possible

Be clear in copy

Avoid jargon and idioms

Use section headings to organize content.



And it is a good start to get everybody on board.  
(<https://accessibility.voxmedia.com/>)



3.

We will need  
to **create alignment** (a lot)



# How are you handing over design specifications?



# **Why** are you handing over design specifications?



# **What** are you handing over as design specifications?



The best handover is **not having to do one.**





**Engineers, this is your fight:**  
**Your opinion is valuable.**



**Engineers, this is your fight:**  
**There is no design without implementation.**



**Engineers, this is your fight:**  
**You are legit.**



**Designers, this is also your fight:**  
**You have to care about implementation.**



**Designers, this is also your fight:**  
**You have to think about interactions.**







Specifications **can't be limited to static designs.**



**Labels and "invisible" texts** should be discussed and defined



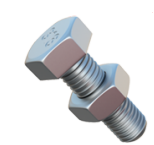
There is **no such things as a perfect design** specification



Designers' involvement should **never end**



Engineers' involvement should **never wait**



Tooling will **not save you** from failure



damien\_castor 4mth

Description

Got any notes?

Share screen

Web: [zpl.io/VDB6PLg](https://zpl.io/VDB6PLg)

App: <zpl:///screen?pid=5b83afe0fedd...>

Tags

Manage

# Date picker specifications

## Examples

Record creation date

Record creation date

<

March 2019

>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

## Variations of the component

Default date picker — Single day selection

Record creation date

2

<

March 2019

>

3

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24

<

March 2019

>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24

Interactive area:  
Loads the previous/next month without transition.

Interactive area:  
Fills the input with the selected day and close the date picker.



This should not be a handover, but a conversation starter

# Variations of the component

## Default date picker — Single day selection

Record creation date

1

2

3

4

March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

<

March 2019

>

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Interactive area:  
Loads the previous/next month without transition.

Interactive area:  
Fills the input with the selected day and close the date picker.

Focuses on current day if there is no value.  
Focuses on entered day if there is value.  
Navigation within the numbers by using the arrow keys.  
Loads prev/next month when going out of current month.  
Page up / Page down to load the next / previous month.  
Home / End to move quickly to the first / last day of the month.



This should not be a handover, but a conversation starter

4.

We will need  
to **create availability.**



**In the beginning, our conversations were like:**  
**“hey, I’m missing something in the ticket”**



**In the beginning, our retrospectives were like:**  
**“We are not getting enough information.”**





**In the beginning, our retrospectives were also like:**  
**“Tickets should have more info.”**



**The thing is:**  
**Each time we added information,**  
**the situation worsened.**



**So, how did we resolve it?**





Engineers are **involved before refinement**.



Before doing any UI design, we **discuss the minimum set of features** to have and the **main limitations** to consider



We **debate HTML and CSS** between design and engineering



We discuss **future needs to consider**, even if not implemented yet



Nobody's owning the codebase



We are **making adjustment on the fly**, every day.



# Table Actions component #154

[Edit](#)[New issue](#)[Open](#)

hiwelo opened this issue 3 days ago · 0 comments



hiwelo commented 3 days ago



This *organism* component is aimed to be used within a table view and contains a series of smaller standalone components.

After a review of the designs with @pmartin35, we aligned the implementation as described below. This content is also available in a story in JIRA for the EDC board.

I keep this information here to document our discussion and the API of these new components for the moment where we will move these to the library 👍

## Component breakdown

- `TableActions` component
  - `Stack` component – rendering the vertical list of icons
  - `Button` component – link to the target of the icon (if no specific target, open study to the relevant phase / step)
    - `Badge` component – rendering the small counter badge (hidden if equal to zero, should have option to display without numbers (small dot variation))
    - `Icon` component – can be empty, should display empty cell in the row in this case
  - `Progress` component

## New component quick specs

### TableActions

- The component is not clickable, despite having a `:hover` / `:focus-within` variation.
- Links to each available actions should be wrapping each icons.

For a better integration within the design system, it would help if this component:

- Get as a prop the list of requested actions (icon, link target, aria-label, badge content)

Assignees



hiwelo

Labels



a11y

documentation

new component

Projects



Needs triage in Component request...

Milestone



No milestone

Notifications

[Customize](#)

🔊 Unsubscribe

You're receiving notifications because you're watching this repository.

2 participants



Lock conversation

Pin issue



# Now, it is not a handover anymore, but a documentation of decisions



I keep this information here to document our discussion and the API of these new components for the moment where we will move these to the library 👍

## Component breakdown

- **TableActions** component
  - **Stack** component – rendering the vertical list of icons
  - **Button** component – link to the target of the icon (if no specific target, open study to the relevant phase / step)
    - **Badge** component – rendering the small counter badge (hidden if equal to zero, should have option to display without numbers (small dot variation))
      - **Icon** component – can be empty, should display empty cell in the row in this case
  - **Progress** component

## New component quick specs

### TableActions



Now, it is not a handover anymore, but a documentation of decisions

ally  
docum  
new co

Project:

Needs t

Milesto

No mile

Notifica

You're r  
you're v

2 partic



## Progress

- This component is having two main variations:
  - The record is excluded, we are displaying a specific variation with two purple bars around the exclusion icon
  - Otherwise, we are displaying a regular progress bar

For the exclusion "state", the DOM of this component should:

- only include the exclusion icon, with an `aria-label` containing *Record excluded*
- the side bars can be CSS pseudo elements with flexbox as discussed with Patrick

For the regular progress bar:

- Using a div for the visual progress with a width based on the `percent` prop
- Using a `<progress max="100" value={percent}>` HTML element with a *visually-hidden* CSS to make this information accessible (<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/progress>)



Now, it is not a handover anymore, but a documentation of decisions

- Using a `<progress max="100" value={percent}>` HTML element with a *visually-hidden* CSS to make this information accessible (<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/progress>)

## Resources

- Designs on Zeplin:
  - <https://zpl.io/aNz3zZN>
  - <https://zpl.io/2ZnA5pE>

  hiwelo added **new component** **documentation** **a11y** labels 3 days ago

  hiwelo self-assigned this 3 days ago

  hiwelo added this to **Needs triage** in **Component requests & bugs** via **automation** 3 days ago



Which means that now, designs on Zeplin are **just additional resources**, not the core element of our collaboration anymore.

But this can only work if  
people are **creating availability**.



This can only work if  
people are **discussing together**.



This can only work if  
people **value their colleagues' knowledge.**





**Because in the end, who cares if I am  
a designer who codes or an engineer designing UIs**



**@iamhiwelo**



**We are here to offer the most inclusive experience for all humans,  
not to fight about “who's job is it?”.**



**@iamhiwelo**



**Merci beaucoup ! 🇫🇷**      **Bedankt! 🇳🇱**

**Thank you! 🇬🇧**      **Tack så mycket! 🇸🇪**

**Vielen Dank! 🇩🇪**      **Tusen takk! 🇳🇴**



@iamhiwelo



# Damien Senger

Queer designer who codes, specialised in accessibility.

[raccoon.studio](https://raccoon.studio) • [noti.st/hiwelo](https://noti.st/hiwelo)



@iamhiwelo