Engineer Software

Design Systems & UI

18 achievements

Connect the parts of your system

The work collected under “Design Systems & UI” shows one part of a working system. We can help connect that part to the rest of your operation. Interfaces that load quickly, work on a phone and behave the same in every browser your customers happen to use.

Explore our services Get in touch →

Find a comparable challenge

Follow an example that resembles your situation. Its review describes the decisions behind the result, beyond the technology label. Directed full-stack GIS map development — full-stack mapping, from database to browser

Directed full-stack GIS map development Get in touch →

Presented 200 UI/UX improvements

Presented 200 UI/UX improvements for the GIS map application, boosting revenue by 10x through enhanced software features.

Contributed to user interface design processes

Contributed to user interface design processes, ensuring intuitive, visually appealing, and user-friendly project interfaces.

Designed 30 websites from Photoshop to HTML

Designed 30 websites, delivering unique and flexible solutions by converting Photoshop designs to HTML.

Designed an iOS 26 'Liquid Glass' design system

Designed an iOS 26 'Liquid Glass' design system and a canonical component inventory enforced by lint rules to prevent UI divergence.

Designed the product's UI and UX end to end

Designed the product's UI and UX end to end — directory grids, dual card/table views, live requirement validators and breadcrumb navigation.

Built a multilingual Hugo site in 3 languages

Built a multilingual static site in Hugo across 108 template files including 53 partials, publishing every page in four representations from one content tree in three languages, and again under seven focused subdomains built from that same tree.

Replaced 27 font sizes with a 6-step scale

Replaced 27 rendered font sizes whose nearest neighbours were 0.6% apart with a six-step Major Third scale, and wrote the linter that fails the twenty-eighth.

Accessibility checks across 32 pages and both themes

Implemented WCAG 2.2 Level AA checks across 32 representative pages in both colour themes, with a conformance record, one AAA criterion checked and the partial result for criterion 1.4.13 stated explicitly.

The accessibility faults your checker misses

Found and fixed 11 accessibility defects the browser reported as healthy — eight footer links still in tab order behind pointer-events, a scroll timeline clipping the colophon off three pages, and a rule engine running 70 of its 105 rules.

Closed the colour system at 28 colours

Closed the colour system at 28 documented colours with a linter that fails a value painted but undocumented, documented but unpainted, mis-measured, respelled as a literal, or within a perceptual distance of 0.02 of one already there.

Found the light theme missing a whole layer

Found that the light theme had been missing a full-page overlay layer since the day it was written, by asserting that both themes paint every layered surface with the same number of layers.

Made the reduced-motion setting honest

Made reduced-motion honest after finding eleven selectors still animating under the preference, because a universal transition-none rule loses on specificity to any rule carrying a class.

Built a 637-line print stylesheet

Built a 637-line print stylesheet against four documented rendering-engine behaviours, including run-in headings that printed white on white for any reader whose browser preferred dark.

Cut the stylesheet bundle from 87 KB to 31 KB

Cut the stylesheet bundle from 87 KB to 31 KB by taking a base64 font out of it, and dropped 756 KB across 22 files that were published on every deploy and referenced by nothing.

Built a native macOS client in Swift 6

Built a native macOS messaging client in Swift 6 and SwiftUI — 11,141 lines across 53 files — over the C interface of a Rust core linked as a static archive from a pinned revision.

Named all 17 icons for screen readers

Named every icon-only control in the interface for screen readers after finding the send button announced as "arrow up circle, button", and wrote the linter that requires the label within eight lines of the icon.

Built a 48-token macOS design system

Built a 48-token design system on the platform's own glass material and wrote the linter that rejects a magic number, a hardcoded font size, or an animation that ignores the reduce-motion preference.

Built the visual identity from 2 drawings

Built the company's icon and favicon sets from two hand-made drawings, with every derived asset regenerated by script and a check that fails when a derived file was committed before its source.

Focus the engineering effort

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Put the subject in your context

Tell us how “Design Systems & UI” appears in your own system. We can help choose the relevant service and define the work around it.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://software.engineer.company/categories/design-systems/