**Engineer** Software

# 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.

2025

## one component inventory, kept one by the linter

**Situation.** A UI without a shared visual language and a fixed set of components drifts, and it drifts fast. Every new screen reinvents its own buttons and badges and cards, each a little different, and those small inconsistencies pile up until the product looks incoherent and every change means touching five bespoke versions of the same thing.

**Task.** The design system had to be coherent enough to look deliberate, and enforceable enough that it wouldn't erode the moment the team was busy.

**Action.** The visual language and the component system were designed as one thing. The language is an iOS 26 "Liquid Glass" look — translucent panels with backdrop blur, layered shadows, a bit of refraction, spring animations — built out of Tailwind utilities. On top of it sits a canonical set of components every screen is supposed to compose from: Card, Label, Button, GlassIconButton, DirectoryGrid and the rest. The part that makes it stick is the linting: rules that reject a hand-rolled badge or chip and point you at the canonical component instead, so the system is held up by tooling rather than by whoever's reviewing that day remembering to care. And the docs are paired with concrete mistake-to-resolution notes, so the guidance is "here's the wrong way and the right way," not an abstract principle.

**Result.** The UI stays consistent and on-brand, and the one-off components that would otherwise spread get caught before they do. Visual consistency stopped being a matter of everyone's discipline and became something the tooling holds the line on, which is the only version of it that survives a deadline.

### Bring this experience to your project

Use “Designed an iOS 26 'Liquid Glass' design system” as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. 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 →

### Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Designed an iOS 26 'Liquid Glass' design system — one component inventory, kept one by the linter

**Explore our services** Get in touch →

## Need a consistent design system?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Design Systems Engineer. It draws on Frontend Engineering, Design Systems & UI, UX / UI Design and Documentation. And while you are here, we are rather good at Frontend Development and UI/UX Design & Design Systems.

**Get in touch** Explore our services

## One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

**See the whole record**

### Define the improvement you need

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 →

# Need a consistent design system?

Ask about the decisions behind “Designed an iOS 26 'Liquid Glass' design system”. You will speak with the people who can explain the work and assess your own project.

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/portfolio/designed-an-ios-26-liquid-glass-design-system/