Userflow Gap-Check Skill

An agentic Claude skill that audits the documented user flow in Figma against the live web portal and the final UI designs — and reports every gap.

ROLE
Concept, skill definition, prompt engineering

TRIGGER
/check-userflow-web-vs-figma

MODEL
Claude Opus 4.8

TOOLS
Claude in Chrome, Figma Dev Mode MCP, FigJam

PLATFORM
Claude Desktop + Figma

Problem → Skill Design → Technical Setup → Automated Audit → Impact

Design Ops

1. Discovery & Problem Analysis

🔍 Flow documentation silently drifts out of date

Keeping the global user-flow board in sync with the shipped product was manual and irregular: designers had to click through the portal and compare every screen, modal, and state against Figma by hand.

CORE PROBLEM

Shipped screens and states are missing from the flow board

Features ship, modals and error states get added — but the documented user flow only updates when someone remembers. Gaps accumulate silently.

AFFECTED ROLES & FRICTIONS

UI Designers

Manual audits are tedious and constantly deprioritized under delivery pressure.

Original statement:
"I never know if the flow board still matches the app."

Product & UX

Outdated flow documentation misleads onboarding, planning, and design discussions.

Original statement:
"We planned against a flow that didn't exist anymore."

EVIDENCE

8
screens & states in one portal area alone

25 %
of them completely missing in the flow documentation

Systems Thinking

2. Skill Design & Concept

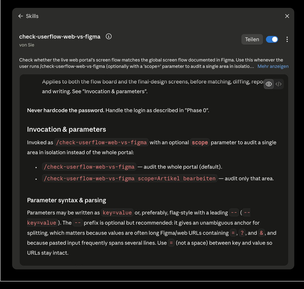
🧠 From manual audit to a parameterized Claude skill

Instead of ad-hoc prompting, a reusable skill definition (/check-userflow-web-vs-figma) encodes the whole audit: scope resolution, browser walk, Figma extraction, matching heuristics, and report format.

WORKFLOW STEPS

0. **Scope resolution (gate)**
Resolves `--scope` on both sides: the matching Figma section and the web entry point. Ambiguity → ask, don't guess.
1. **Extract the documented flow**
Enumerates the frames on the flow board — one frame per documented screen, incl. components and transitions.
2. **Walk the live portal (mode=web)**
Breadth-first walk capturing every distinct screen, modal, and state — non-destructively.
3. **Read the final UI designs (mode=design)**
Alternatively compares "ready for dev" UI screens against the flow board — no browser needed.
4. **Match by content, not by name**
Frames are classified by rendered content (brand, state); name/content conflicts are flagged for human review.
5. **Report the gaps**
Structured report: missing screens, missing components, diverging transitions, stale documentation.
6. **Render the gap diagram**
A color-coded FigJam diagram shows at a glance what's documented, partial, or missing.

SKILL PARAMETERS



Automation

3. Technical Setup

⚙️ One command, three parameters, three modes

The skill runs inside Claude with browser and Figma access. Parameters make it reusable across portal areas (--scope), brand variants (--case), and comparison targets (--mode).

TECHNICAL SPECS

Invocation	/check-userflow-web-vs-figma --scope=… --case=… --mode=…
Modes	web (portal vs. flow) · design (final UI vs. flow) · both
Browser tooling	Claude in Chrome — find, click, form input, batch actions
Figma tooling	Dev Mode MCP (read) + generate_diagram (FigJam gap diagram)
Guardrails	Test environment only, non-destructive, no credential handling
Output language	German — structured gap report directly in the chat thread
Fixed output	Summary table + color-coded FigJam gap diagram per run

AI Testing

4. Automated Audit — Core Feature: Gap Diagram

✅ From audit run to actionable worklist — in one command

Gaps aren't just described — they're rendered as a color-coded FigJam diagram plus a summary table, so designers see instantly which frames to add.

A · MODE=WEB — PORTAL VS. FLOW

Every screen and state of the live portal, checked

Breadth-first walk through the area incl. modals, empty and error states — matched against the flow board by content, not by frame name.

User-Flow-Abgleich: Web vs. Figma

Geprüft am 30.07.2026 · **Webportal:** `https://web.loql-test.com/` · **Figma-Flow:** Einstellungen ✅
Scope: Einstellungen (einziger Bereich, auf den `userflow-section` verweist) · **Abgedeckte Screens (Web):** 1 Hauptseite + 7 Modal-Bereiche · **Dokumentierte Frames (Figma):** ~40 (5 von 7 Bereichen)

Zusammenfassung

Screen	Web-URL	Figma-Frame	Status	Hinweise / fehlende Komponenten	Figma-Link
Bestell- und Lieferhinweise (Modal)	`/settings`	`115:21591`	✅	Textarea + Speichern/Löschen entsprechen dem Frame	Link
Abwesenheitsnotiz — Empty/Error	`/settings`	`115:21783`, `115:22539`	✅	Toggle + Textarea, Pflichtfeld-Validierung stimmen überein	Link
Abwesenheitsnotiz — Toggle „An"	`/settings`	—	⚠️	Startdatum*/Enddatum*-Felder erscheinen erst nach Aktivierung, kein eigener Frame dafür gefunden	—
Sortierung der Bestell-PDFs (Modal)	`/settings`	— (nur Connector zu Success)	⚠️	4 Radio-Optionen im Web: das Modal selbst ist als Frame nicht dokumentiert, nur der Erfolge-Snackbar	Filled State
Cut-off-Zeit	`/settings`	—	❌	Komplette Sektion (Zeitauswahl-Modal) fehlt in Figma	—
Liefertage	`/settings`	—	❌	Komplette Sektion fehlt: pro-Partner Wochentags-Grid (10 Partner à 7 Tage)	—
Unternehmensinformation (Modal)	`/settings`	`115:22034`, `115:22082`, `115:22130`, `115:22333`	✅	Textarea-Modal stimmt überein	Link
Bestell-E-Mails (Modal + Errors)	`/settings`	`115:22190`, `115:22385`, `115:22433`, `115:22238`, `115:22286`	✅	„Ungültige E-Mail" & „Trennung mit Komma"-Fehler beide reproduziert und bestätigt	Link

Status-Legende: ✅ vollständig dokumentiert · ⚠️ teilweise (Komponenten/Übergänge fehlen) · ❌ Screen fehlt in Figma · 🗑 in Figma vorhanden, aber nicht mehr im Web

Fehlende Screens (im Web, nicht in Figma)

- **Cut-off-Zeit** — `/settings` (Modal) — Schlüsselkomponenten: Uhrzeit-Feld (Type „time"), Speichern/Abbrechen/Löschen
- **Liefertage** — `/settings` (Modal) — Schlüsselkomponenten: Pro-Partner Wochentags-Buttons (Mo–So) für 10 Partner, scrollbare Liste, Speichern/Abbrechen

Fehlende Komponenten je dokumentiertem Screen

- **Abwesenheitsnotiz** (Empty-Frame `115:21783`): fehlend in Figma → Zustand „Toggle aktiviert" mit Startdatum*/Enddatum*-Feldern
- **Sortierung der Bestell-PDFs** (nur über Connector referenziert): fehlend in Figma → das eigentliche Auswahl-Modal mit den 4 Radio-Optionen als eigener Frame

Abweichende / fehlende Übergänge

- Einstellungen-Übersicht → Cut-off-Zeit-Modal: im Web vorhanden, in Figma nicht dokumentiert
- Einstellungen-Übersicht → Liefertage-Modal: im Web vorhanden, in Figma nicht dokumentiert
- Abwesenheitsnotiz-Modal → Toggle-An-Zustand: im Web vorhanden, in Figma nicht dokumentiert

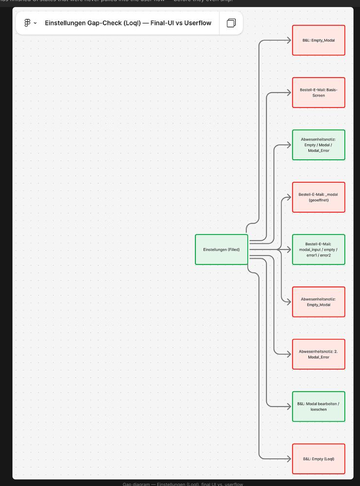
Mögliche veraltete Dokumentation

- Keine gefunden — alle geprüften Figma-Frames hatten eine erkennbare Entsprechung im Web (keine verwaisten Frames im untersuchten Bereich).

B · MODE=DESIGN — FINAL UI VS. FLOW

"Ready for dev" designs checked against the documented flow

Finds finished UI states that were never pulled into the user flow — before they even ship.



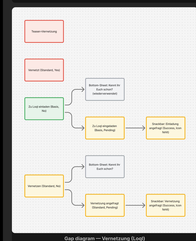
Gap diagram — Einstellungen (Logi), final UI vs. userflow

C · GAP DIAGRAM & RESULTS TABLE (CORE FEATURE)

One glance shows what's documented and what's missing

Green = documented in the flow, red = missing. The chat report adds Figma deep links per frame, missing components per screen, and data-quality findings such as name/content mismatches.

```
- BBL: Empty_Modal – missing in the userflow
+ Abwesenheitsnotiz: Empty / Modal / Modal_Error – documented
```



Summary table (chat report)
Per-screen status, action needed, and Figma deep links

Gap diagram — Vernetzung (Logi)
Color-coded FigJam output: documented, partial, missing

Business Impact

5. Impact & Validation

💎 Flow documentation that stays trustworthy

Keeping the flow board current no longer depends on discipline — one command audits an entire area and returns a worklist of exactly the frames to add.

IMPLEMENTATION VS. DESIGN — TWO AUDIT RUNS

18
screens & states audited across two runs

6
implemented 1:1 as specified in Figma

5
partially implemented — missing components, cases or states

7
not implemented at all

PRINCIPLE

Documentation quality isn't secured by extra manual effort, but by an audit that's repeatable, parameterized, and embedded in the design workflow — from live product to an actionable Figma worklist in one command.