

Northstar

Insurance Design System

Turning fragmented product interfaces into a shared delivery platform across claims, policy, billing, service and analytics.

My role	Scope	Platform	Duration
Design System Lead	5 products · 8 squads	Enterprise web	Approx. 12 months

Project focus

I led the design strategy, system architecture and adoption model. The work connected user needs, accessibility, design decisions and front-end delivery—not just a Figma component library.

CONFIDENTIALITY Company and product names are omitted. Screens and figures are reconstructed; metrics are rounded, anonymized ranges.

Ajay Kamat

Senior Product Designer · Enterprise UX · Design Systems

Every squad was paying the inconsistency tax

Five insurance products had evolved through separate roadmaps, vendors and engineering stacks. Similar workflows looked and behaved differently. Product teams rebuilt common patterns, accessibility fixes arrived late and customers had to relearn interactions across journeys.

70+	8 squads	5 products	3 themes
Overlapping UI patterns	Different delivery practices	One fragmented experience	Uneven implementation

Business problem

Inconsistency was no longer a visual-quality issue. It increased delivery time, QA effort, accessibility exposure, UI rework and onboarding cost.

User impact	Business impact
Unpredictable forms, tables, alerts and navigation increased cognitive effort during complex insurance tasks.	Repeated design and development reduced roadmap capacity and made cross-product change expensive.

DESIGN CHALLENGE How might we create a governed system that accelerates delivery without removing the flexibility products need?

I led the system as a product

- Defined the design-system vision, principles, roadmap and success measures.
- Planned and ran the cross-product interface and accessibility audit.
- Created foundations, token architecture, component specifications and complex patterns.
- Paired with design engineering to align Figma variables, code tokens and component APIs.
- Facilitated prioritization and trade-off decisions with product, engineering and accessibility.
- Created contribution, release, migration and governance workflows.
- Coached product designers and led adoption reviews with eight squads.

Core team

Design system lead	Design engineer	Front-end leads	Accessibility partner
Strategy + UX quality	Component implementation	Product integration	Standards + testing

Leadership boundary

I owned experience quality and system decisions. Engineering owned technical architecture and releases; product squads owned migration within agreed roadmaps. Shared decisions were documented through lightweight decision records.

Research combined users, interfaces and team behavior

Interface audit · where inconsistency created cost

	Claims	Policy	Billing	Service	Analytics
Buttons	6 variants	4 variants	5 variants	4 variants	3 variants
Form fields	8 variants	7 variants	6 variants	5 variants	4 variants
Tables	5 variants	4 variants	3 variants	4 variants	6 variants
Alerts	7 variants	5 variants	6 variants	4 variants	5 variants

Finding Teams repeatedly solved the same UI problem, producing 70+ overlapping patterns and uneven accessibility.

Audit count represents reconstructed, anonymized ranges across five product surfaces.

Methods

- Audited 200+ representative screens and mapped repeated patterns, states and defects.
- Interviewed product designers, front-end engineers, QA and product managers across squads.
- Observed build handoffs and reviewed defect, accessibility and rework themes.
- Mapped high-volume workflows: intake, search, forms, review, exception handling and data tables.
- Prioritized components using reach × inconsistency × risk × implementation effort.

Key insight

THE SYSTEM GAP Teams did not lack components. They lacked shared decisions: when to reuse, when to extend, who approves change and how quality is verified.

Scale required deliberate boundaries

Legacy UI	Dense workflows	Multiple stacks	Accessibility	Themes	Capacity
Migration could not stop delivery	Tables and forms carried risk	Code adoption varied	WCAG 2.2 AA target	Brand and contrast modes	Small central team

Non-negotiables

- No big-bang redesign; adoption had to fit live product roadmaps.
- No component entered release without keyboard, focus, error and screen-reader behavior.
- Tokens carried design intent across themes; products could not consume raw color values.
- System changes required migration notes and versioned documentation.
- Product exceptions were visible, time-bound and reviewed—not hidden forks.

2026 readiness

The component model included loading, empty, error, permission, stale-data and AI-assisted states. AI suggestions used consistent labels for source, confidence, review status, correction and fallback so future intelligent workflows inherited the same trust contract.

Standardize decisions, not every screen

Decision	Trade-off	Why
Build a shared core + product recipes	More governance work than a loose library	Core components stay stable; workflows remain contextual
Tokens before broad component expansion	Slower visible progress at first	Prevents theme and code divergence
Prioritize forms, tables and feedback	Marketing patterns came later	Highest operational reach and risk
Composable APIs with controlled variants	Less local freedom	Reduces one-off combinations and testing cost
Federated contribution with central review	Requires coaching and triage	Scales knowledge without losing quality
Measure adoption and outcomes	Instrumentation effort	Proves business value beyond component count

Principles

Clear	Predictable	Inclusive	Composable
Intent before decoration	Same action, same behavior	Accessibility by default	Stable parts, flexible workflows

DECISION RULE A pattern entered the system only when it solved a repeated user problem across products and could be supported in design, code, documentation and testing.

Tokens connected design intent to production

A token architecture that connected intent to code



Figma variables ↔ token package ↔ coded components

Three layers made themes and controlled component exceptions manageable.

Foundation scope

- Color roles and contrast-safe interaction states.
- Typography for operational scanning and data density.
- 4-point spacing, responsive grids and layout primitives.
- Elevation, border, radius and focus treatment.
- Motion guidance including reduced-motion behavior.
- Content conventions for actions, status, errors and recovery.

System contract

Design variables and code tokens used the same semantic names. A change to an intent such as `action.primary` or `surface.warning` could be reviewed once and released through a versioned token package.

Each component shipped with behavior, not just pixels

Production component anatomy · form field

Employee ID
Required

CA-48291

Use the ID shown on the claim record.

Default

Focus

Error

Disabled

Definition of done

✓ Keyboard sequence

✓ Visible focus

✓ Label and description

✓ Error recovery

✓ Screen-reader name

✓ Zoom and reflow

✓ Light/dark modes

✓ Usage guidance

✓ Automated tests

Example specification covering anatomy, states and accessibility acceptance criteria.

Release package

Design	Code	Guidance	Quality
Figma component + variables	Versioned implementation	Use, avoid and content	A11y, unit and visual tests

Components followed WCAG 2.2 AA-oriented requirements and relevant WAI-ARIA Authoring Practices. Automated checks supported quality, while keyboard and assistive-technology verification remained human review.

Components became useful through workflow recipes

Insurance teams needed more than buttons and fields. I designed reusable recipes for decisions that repeated across products.

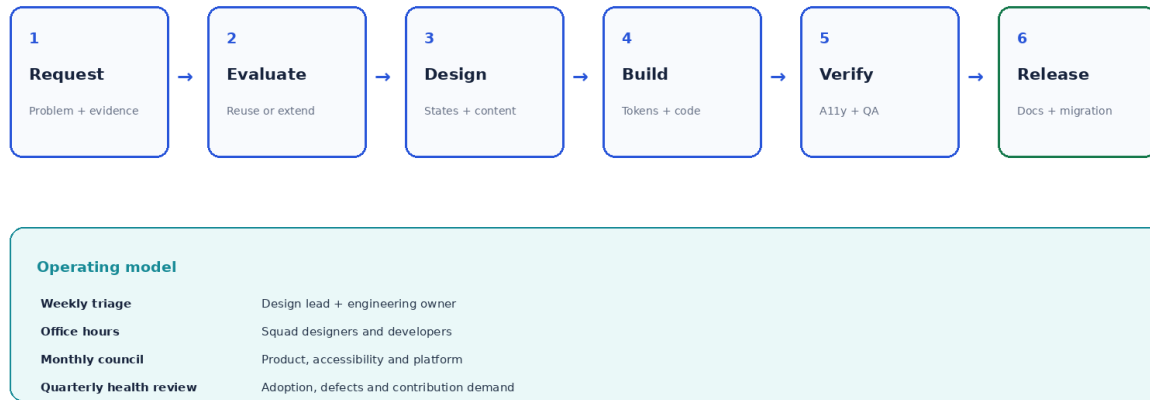
Recipe	Problem solved	Required states
Guided intake	Long, conditional data capture	Save, resume, validation, dependency
Search + filter	High-volume record finding	No result, stale data, permission
Review + submit	Consequential confirmation	Changed values, warning, audit
Data table	Dense comparison and action	Sort, select, bulk, empty, error
Exception resolution	Blocked operational work	Reason, owner, recovery, escalation
AI-assisted review	Suggested content or action	Source, confidence, correct, reject, fallback

Human review in AI states

The system never treated AI output as ordinary content. Suggestions displayed provenance, review status and a correction path. Consequential actions required explicit confirmation, and unavailable evidence produced a safe fallback.

Governance made the system scalable

Contribution lifecycle • from product need to governed release



A lightweight operating model distributed contribution while keeping release quality centralized.

How I led adoption

- Established a design-system council with product and platform representation.
- Published a transparent backlog scored by reach, risk and delivery value.
- Ran office hours, pairing sessions and migration clinics.
- Created component owners and service-level expectations for critical issues.
- Used release notes, deprecation windows and migration checklists.
- Shared a monthly health view: adoption, exceptions, defects and contribution flow.

Adoption followed product value, not mandates

Pilot	Prove	Expand	Sustain
Claims intake	Forms + feedback	Tables + navigation	Contribution + health

Migration strategy

- Selected a high-volume claims workflow as the pilot and measured effort before and after.
- Aligned migration with active feature work to avoid separate redesign projects.
- Provided compatibility wrappers for legacy surfaces where immediate replacement was unsafe.
- Tracked product adoption by coded component usage—not Figma instances alone.
- Retired duplicates only after migration guidance and owner confirmation.

Validation

Design reviews checked comprehension and workflow fit; engineering reviews checked API and performance; accessibility reviews covered keyboard, zoom, reflow and assistive technologies; product analytics measured adoption and downstream quality.

Measurable business outcomes

Figures are rounded, anonymized ranges intended to communicate scale and direction while protecting confidential company data.

82–88%	35–45% faster	25–35% faster	45–55% fewer
Adoption in target surfaces	Design assembly	Front-end delivery	Accessibility defects

Measure	Baseline	After rollout	Business outcome
Target-surface adoption	Approx. 20%	Approx. 82–88%	Shared platform became default
Design assembly time	8–10 days	5–6 days	Approx. 35–45% faster
Front-end UI delivery	12–15 days	8–10 days	Approx. 25–35% faster
Duplicate patterns	70+	26–32	Approx. 55–65% lower
Accessibility defects / release	18–24	9–12	Approx. 45–55% lower
UI-related rework	Baseline index 100	60–70	Approx. 30–40% lower

Outcome

The organization gained a reusable delivery platform: squads released common interactions faster, users encountered more predictable workflows, accessibility defects moved earlier in the lifecycle and cross-product change became less expensive.

MEASUREMENT NOTE In a real interview, I would explain the source for each metric—delivery logs, defect categories, component telemetry and team surveys—and retain only figures I can defend.

A design system succeeds when teams make better decisions

- A component library is an output; the product is a shared way of deciding and delivering.
- Start with repeated user and business problems, not a long inventory of UI pieces.
- Accessibility scales when acceptance criteria live inside components and release gates.
- Tokens create leverage only when design and code share semantics and ownership.
- Adoption is a product problem: reduce migration cost, prove value and measure usage.
- AI-ready patterns need provenance, uncertainty, human correction and safe failure by default.
- Leadership means resolving incentives and ownership—not personally designing every component.

My 2026 UX contribution

- Connected user evidence, operating constraints and business measures.
- Designed standards for humans and AI-assisted workflows.
- Led cross-functional decisions across design, engineering, product and accessibility.
- Created governance that allowed teams to contribute without fragmenting the system.
- Measured impact through delivery, quality, adoption and user-experience signals.

Reference standards

- W3C Web Content Accessibility Guidelines (WCAG) 2.2 — [w3.org/TR/WCAG22/](https://www.w3.org/TR/WCAG22/)
- WAI-ARIA Authoring Practices Guide — [w3.org/WAI/ARIA/apg/patterns/](https://www.w3.org/WAI/ARIA/apg/patterns/)
- Design Tokens Community Group Format Module — design-tokens.github.io/community-group/format/

PORTFOLIO CLOSE I did not introduce a design system to make products look identical. I created a shared decision and delivery platform so teams could solve complex insurance problems consistently, accessibly and faster.

Ajay Kamat

Senior Product Designer • Enterprise UX • Design Systems