

Service Blueprint Template

Formats: Markdown (canonical) | Word (DOCX) | PDF **Updated:** 2026-03-22 **License:** CC BY 4.0 --
Kate Makrigiannis / k8mak.com

A fill-in-the-blank template for mapping what users see, what happens behind the scenes, and where handoffs break down. Service blueprints go deeper than journey maps by adding frontstage, backstage, and support process lanes.

When to use this

Your user experience involves multiple teams, systems, or manual processes behind the scenes. A journey map shows the user's experience. A service blueprint shows that experience plus everything invisible to the user that makes it work (or fail). Use this when you're diagnosing handoff failures, identifying automation opportunities, or redesigning an operational workflow.

Template structure

A service blueprint maps each phase of the user experience across five lanes. The "line of visibility" separates what the user sees from what they don't.

Phase:	[Phase 1]	[Phase 2]	[Phase 3]
User actions	(What the user does)	(What the user does)	(What the user does)
Frontstage	(Visible touch-points: UI, email, chat)	(Visible touch-points)	(Visible touch-points)
== LINE OF VISIBILITY ==			
Backstage	(Staff actions invisible to the user)	(Staff actions invisible to the user)	(Staff actions invisible to the user)
Support processes	(Systems, APIs, databases)	(Systems, APIs, databases)	(Systems, APIs, databases)
Physical evidence	(Tangible artifacts: receipts, PDFs)	(Tangible artifacts)	(Tangible artifacts)

How to fill it in

Step 1: Define your phases

List the major stages of the experience from the user's perspective. Keep it to 4-6 phases. More than that and the blueprint becomes unreadable.

Step 2: Fill user actions and frontstage first

Start above the line of visibility. For each phase, capture what the user does and what they see or interact with.

Step 3: Map backstage and support processes

Below the line of visibility, capture the staff actions, systems, and processes that enable each frontstage touchpoint. This is where service breakdowns live.

Step 4: Add physical evidence

What tangible artifacts does the user receive or interact with at each phase? Confirmation emails, receipts, reports, documents, notifications.

Step 5: Mark failure points and opportunities

After filling all lanes, mark:

- **Failure points** (where handoffs break or delays occur)
- **Wait points** (where the user waits and the backstage is working)
- **Automation opportunities** (where manual backstage work could be replaced)

Worked example: Online food delivery

Lane	Browse & Select	Place Order	Preparation	Delivery	Post-Delivery
User actions	Browse menu, customize items, add to cart	Review cart, enter address, choose payment, tap "Place Order"	Wait for status updates	Track driver, meet at door	Rate experience, contact support if issues

Lane	Browse & Select	Place Order	Preparation	Delivery	Post-Delivery
Frontstage	App menu screen, item detail pages, cart drawer	Checkout screen, payment confirmation, "Order placed" screen	Push notification: "Your order is being prepared"	Live map with driver location, ETA updates	Rating prompt, support chat
LINE OF VISIBILITY	---	---	---	---	---
Backstage	Menu management team updates items and prices daily	Order routed to nearest available restaurant; payment processed	Restaurant confirms, starts cooking, marks items ready	Driver assigned, picks up order, follows GPS route	Support agent reviews complaint, issues refund if needed
Support processes	Menu CMS, pricing engine, image CDN	Payment gateway (Stripe), order routing algorithm, restaurant POS integration	Kitchen display system, order status API	Driver app, GPS/mapping service, delivery time prediction model	CRM ticket system, refund processing, rating aggregation
Physical evidence	Menu photos, nutritional info	Order confirmation email, receipt	Status update notifications	Delivery bag, receipt in bag	Refund confirmation email, rating thank-you

Failure points identified:

- Order routing sometimes sends to a restaurant 20 minutes farther away (routing algorithm issue)
- Kitchen display delays cause "preparing" status to show for 5+ minutes before the restaurant actually starts
- Refund process requires 3 support agent clicks -- could be automated for orders under \$20

Facilitator tips

- **The line of visibility is the most important concept.** Everything above it is the user's world. Everything below is yours. When something fails above the line, the cause is almost always below it.
 - **Backstage lanes reveal the real problems.** Teams that only map user actions and touchpoints miss the operational breakdowns that cause user pain. The blueprint's value is in making invisible work visible.
 - **Keep it to one row per lane.** If a lane needs multiple rows, you have too many phases or too much detail. Summarize and link to detailed documentation elsewhere.
 - **Use this to find automation opportunities.** Any backstage action that's manual, repetitive, and rule-based is a candidate for automation. Mark them.
 - **Present the blueprint horizontally.** Blueprints are wide. Use landscape orientation, a whiteboard, or a Miro/FigJam board. Portrait documents don't work.
-

How did it go?

- ☐ All five lanes are filled for every phase
 - ☐ The line of visibility clearly separates user-facing from internal operations
 - ☐ Failure points are marked with specific descriptions, not vague labels
 - ☐ At least one automation opportunity is identified
 - ☐ The blueprint is readable by someone who wasn't in the room
-

Part of the [k8 Agent Toolkit](#). Download other formats at k8mak.com/resources.