

Diane Party Rentals

A website and operating platform for a party rental company. Equipment, quotes, payments, and the work of getting everything there.

Design engineer March 2026–present

[Visit Diane Party Rentals ↗](#)

Wiel Zouantcha · Read and interact with this case study at <https://zouantcha.com/case-studies/diane-party-rentals>

Where it started

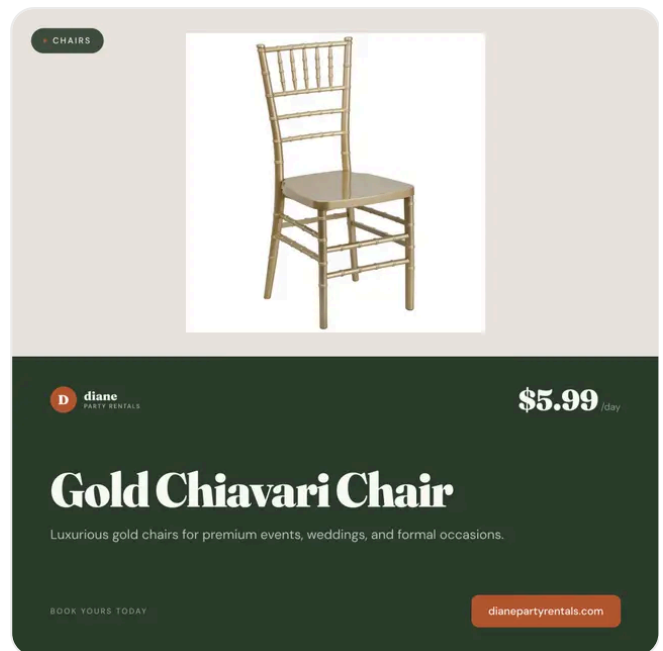
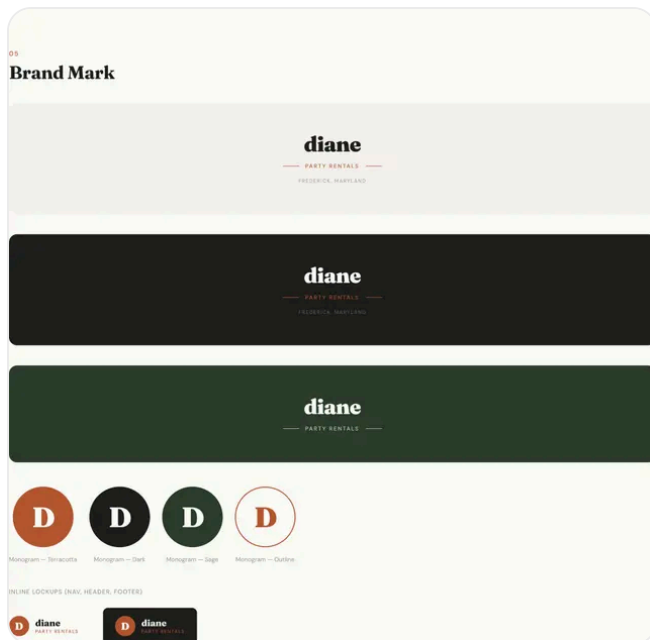
Diane Party Rentals is a family-operated rental company in Frederick, Maryland. They reached out to me in March 2026 with a site on Wix's free plan. Bookings and quotes were handled manually. Payments came in as cash, Cash App, Venmo or Zelle.

I started with the brand and assets, then designed and built the website, quote and payment flow, and the admin behind each booking. The work runs from how someone first encounters Diane to what the crew sees when it arrives at their event.

Giving Diane an identity

Before the website, I worked on the identity in Paper: the diane wordmark, a D monogram, the type, colors and reusable assets. I wanted it to feel welcoming enough for a celebration and clear enough for an invoice.

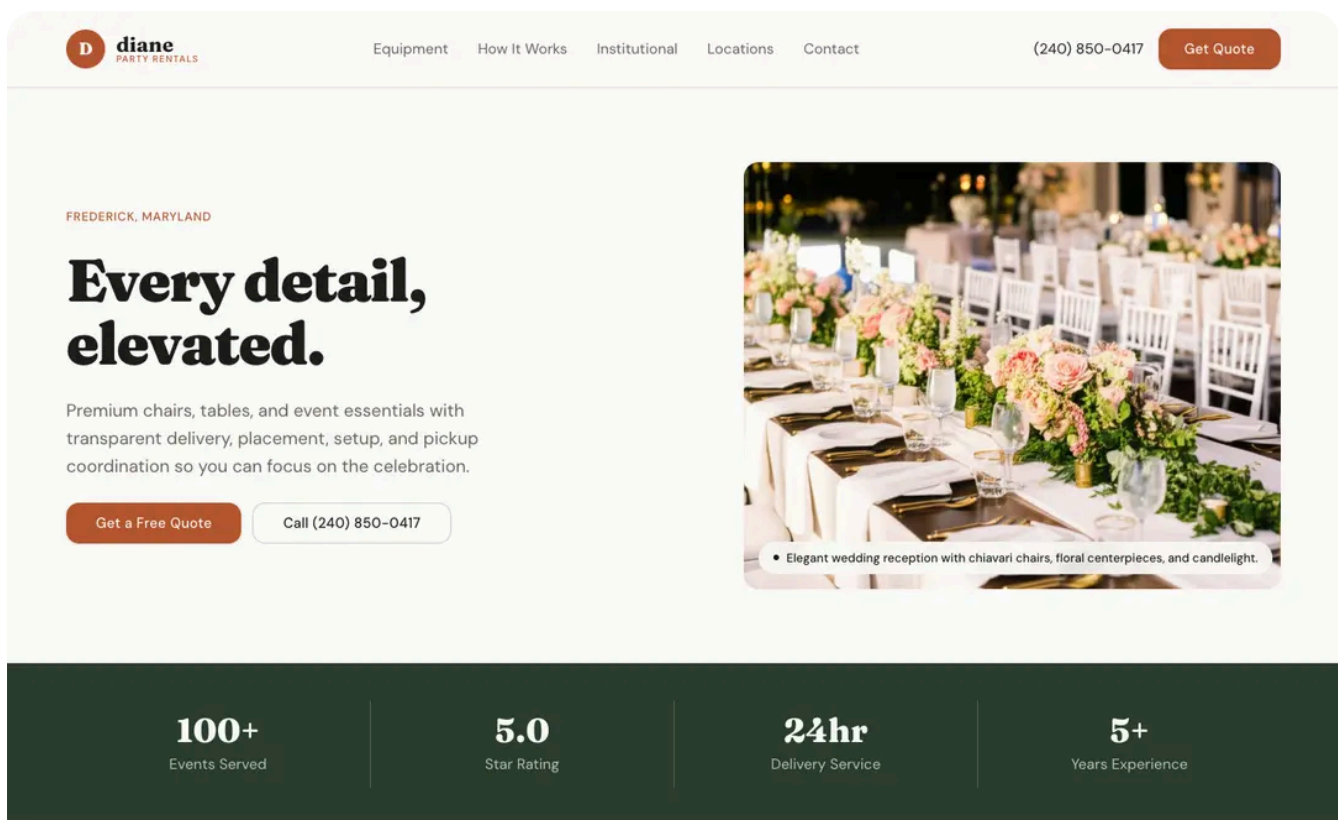
Fraunces gives the name and headings their character. Cream and warm neutrals leave room for equipment photography; terracotta picks out the mark and actions. The smaller D works where the full name would be too much.



The identity and one of its product assets, from Paper. The same pieces carry into the website, customer emails and documents.

I made product templates for Instagram and Facebook, alongside quotes, contracts, receipts and customer emails. These are often the things a customer keeps or forwards. They needed to feel like they came from the same company as the website.

Choosing what to rent



The website, with the catalog and quote flow a click away.

Most customers arrive with an occasion in mind. A birthday, a wedding, a school event. I wanted the site to help them work out what they needed: which chairs, how many tables, whether delivery and setup were available. Equipment photographs do much of that work.

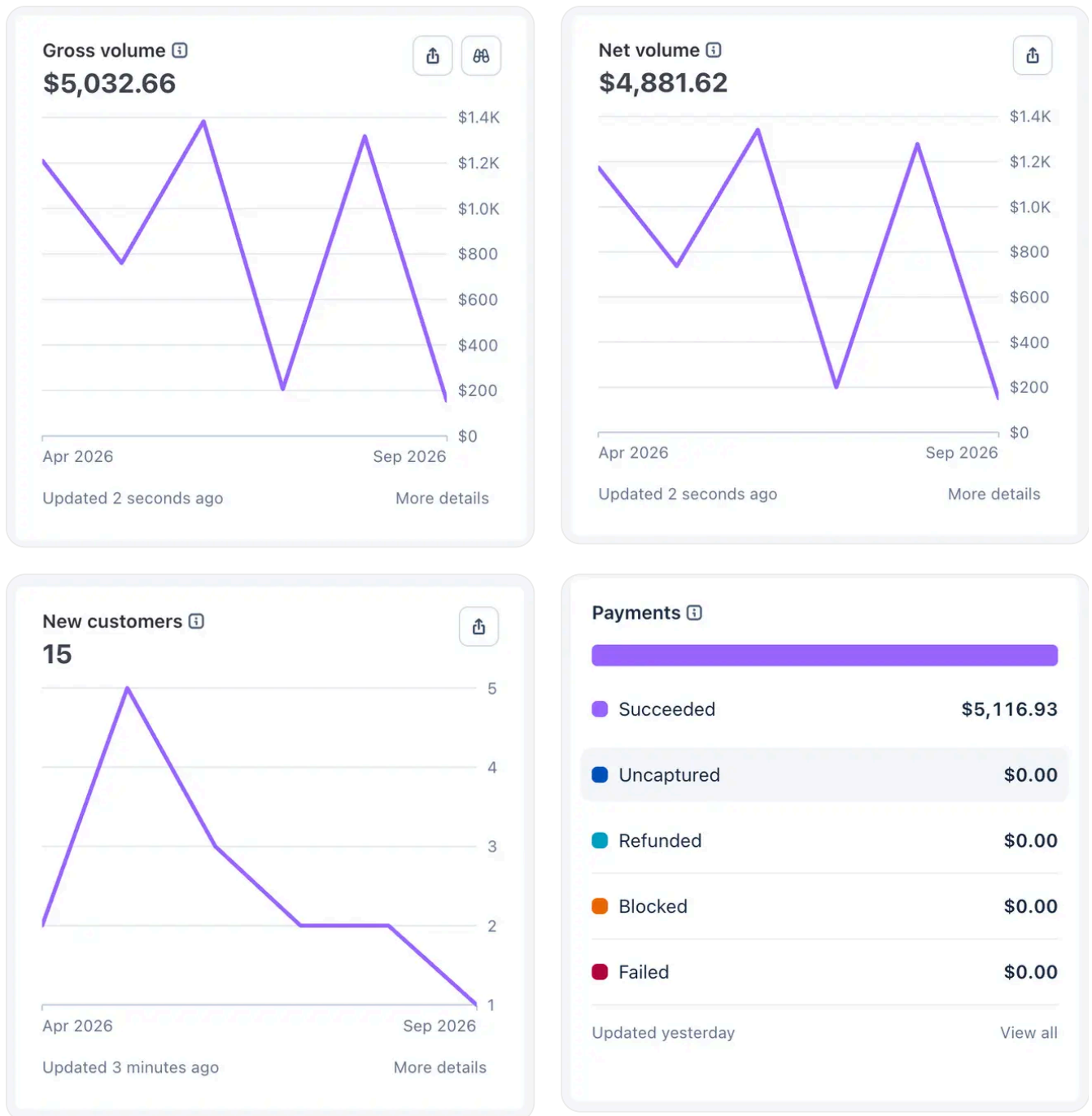
The location pages carry the same catalog into the areas DPR serves, with local delivery information. Schools and other institutions have their own path too. A purchase order comes with different questions from a birthday booking.

From a quote to a paid booking

The old payment methods worked, but the booking lived elsewhere. Someone still had to connect the money to the event and keep track of what remained to be paid. Adding a checkout meant bringing those pieces together.

The customer reviews the equipment and dates, accepts the quote and follows the payment link. The payment then belongs to that booking, alongside its balance and receipt. The team has a record to return to when the customer calls.

Stripe handles the online payment. By September 8, the dashboard records \$5,032.66 in gross volume, \$4,881.62 in net volume and 15 new customers across April–September 2026.

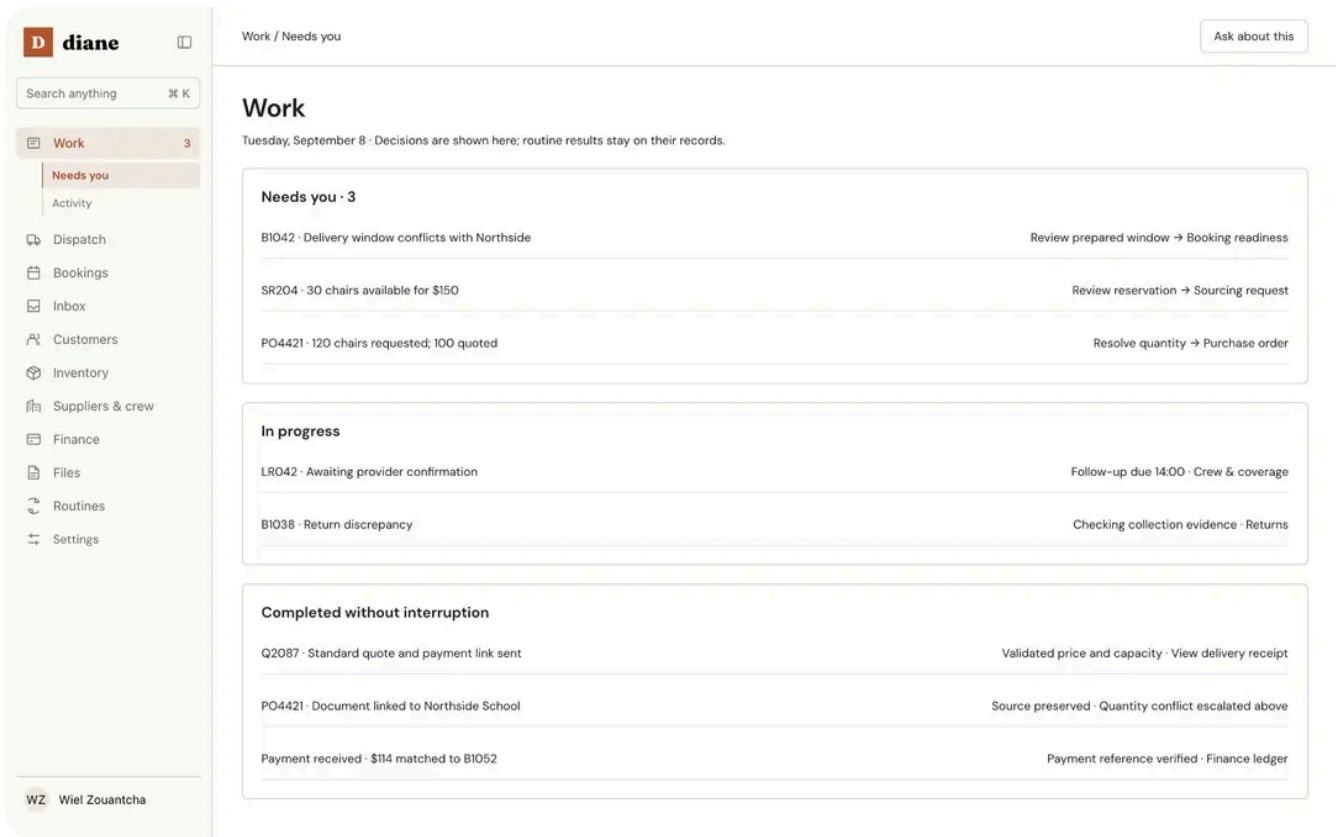


Stripe, September 8, 2026. The volume and customer charts cover April–September; September is partial. The payments overview has its own total, with no date range shown.

Getting everything there

With the customer’s booking and payment in one place, I followed the work into fulfillment. There is still a truck to load, a crew to assign and a customer waiting at the other end. A late pickup can affect the next delivery. A damaged table can leave tomorrow’s booking short.

That is where I spent most of the time on the admin. It opens on the work that needs a decision: a delivery conflict, a supplier reservation, a purchase order with the wrong quantity. Open an item and you are in the booking or request it belongs to. Work already under way and completed actions sit below it.



[View full-size design ↗](#)

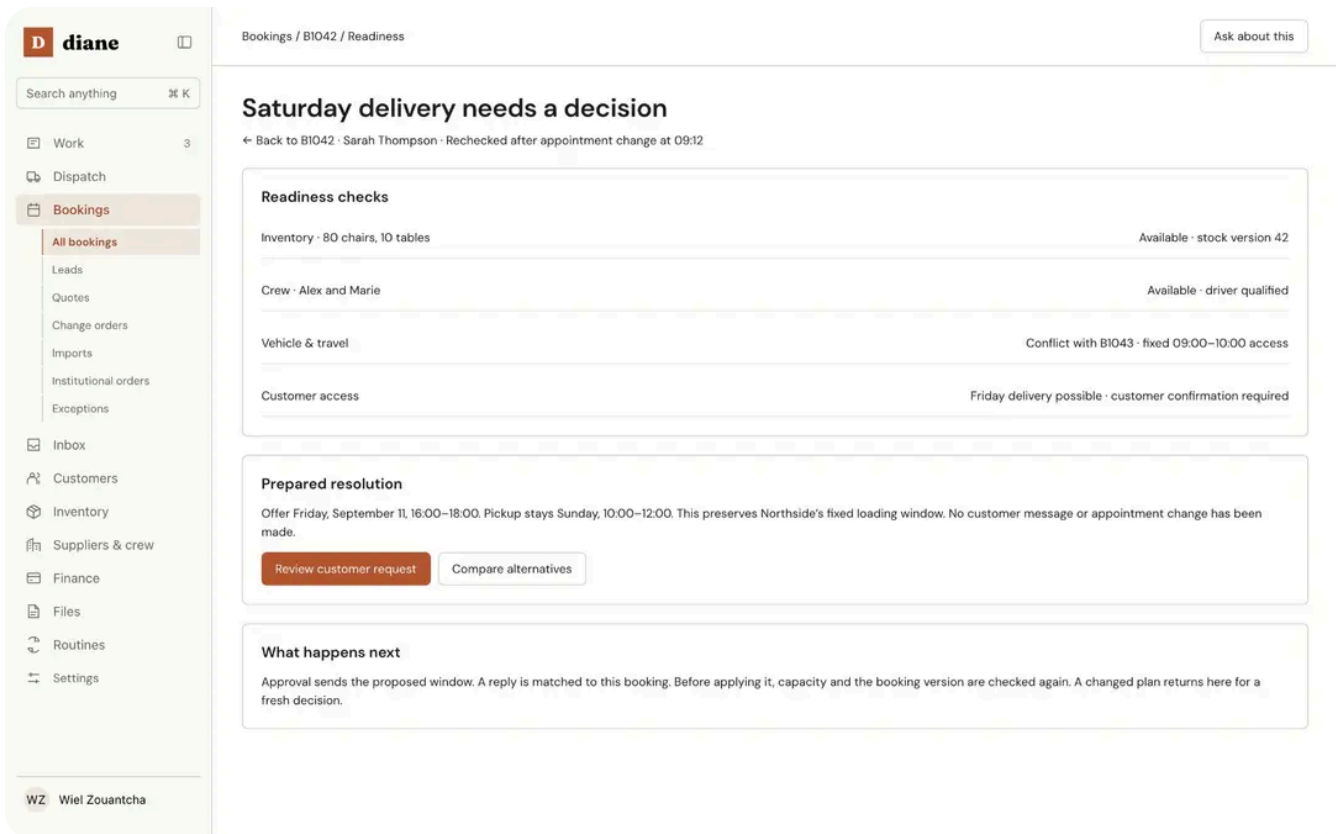
The work view. Three decisions, with the ongoing work and receipts underneath.

Agents handle the checking and follow-through. A booking change starts a readiness check. A shortage starts sourcing. An unanswered request gets a follow-up. The team can ask a question about a record, but it does not need to ask for these checks to happen.

Moving a delivery

Two customers need equipment at the same time. One has a fixed loading window; the other may be able to take delivery the evening before. That is a common sort of problem here, and solving it involves more than dragging a calendar entry.

The platform checks stock, crew and the vehicle, then prepares another window. The operator sees what changes, what stays put and the message the customer will receive. Approving sends the request. The appointment moves after the customer agrees and capacity is checked again.



[View full-size design ↗](#)

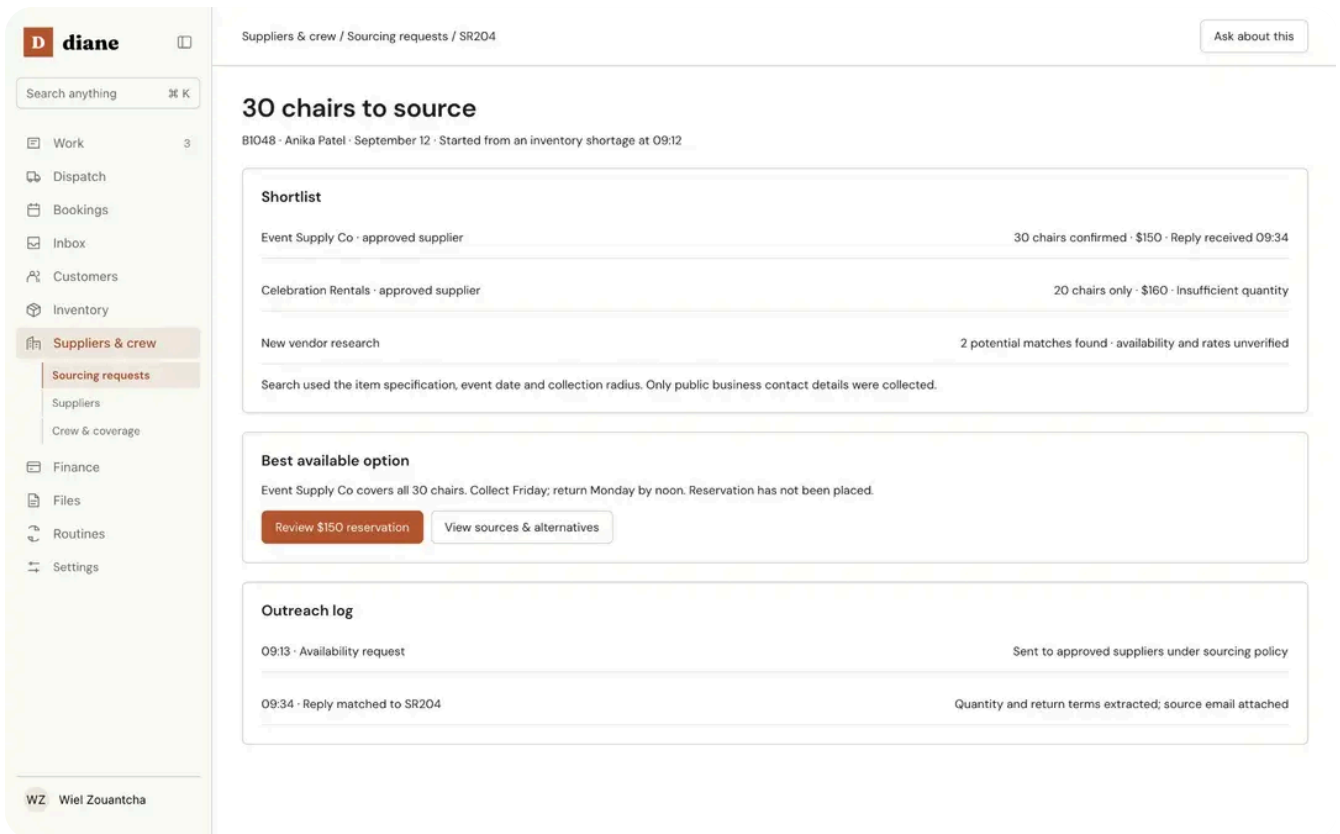
A Friday delivery frees the truck for Saturday's fixed appointment. The Sunday pickup stays the same.

I kept those steps visible. A sent message is not an agreement. If someone changes the booking while a reply is coming back, the old approval needs another look. The record holds the conversation and the decision together, so the next person can pick it up.

Finding thirty more chairs

DPR also brings in equipment from other suppliers. For larger setups, it hires temporary help. Both involve finding someone, checking availability, agreeing on terms and making sure they actually confirm.

For equipment, the agent starts with approved suppliers and researches alternatives when needed. It collects quantities, prices and collection terms in one request. A chair listed on a website stays unverified until the supplier confirms it is available for the date.



[View full-size design ↗](#)

Thirty chairs to source. Confirmed quantities and prices sit beside the options that still need checking.

Availability requests and reminders can go out under the team's policy. Committing money needs approval. If a supplier does not answer, the request stays open and the follow-up has a deadline. It does not disappear into an email thread.

At the venue

The office view is only half of the handoff. The person unloading needs the venue, item count and setup instructions on a phone. I designed the mobile flow around that moment: check what arrived, take a photograph, record the condition and capture who accepted it.

DELIVERY SIGNOFF

#DPR-0147 — Mitchell

On-Site

Venue	Ceresville Mansion
Arrived	9:15 AM
Items	56 of 56 delivered
Setup	Complete
Sourced items	None

Delivery photo

 Take setup photo

Condition notes

All equipment in good condition. Ceremony layout per customer request.

Accepted by

Sarah Mitchell

Signature

S.M.

Complete Delivery Signoff

The delivery signoff keeps the physical handoff with the booking.

Pickup and inspection continue the record. A missing chair or a damaged table needs evidence before it becomes a stock adjustment or a customer charge. The agent can gather the relevant records and flag the discrepancy; the crew still has to count and inspect the equipment.

Keeping the context

A booking accumulates things: a quote, a payment, a purchase order, delivery photographs, a note about which gate to use. Those details need to travel with the work. The agent can find them, and the person checking its answer can open the source.

I use [Eve](#) to coordinate the agents and work that waits for a reply or approval. [Cloudflare Workers](#) runs the application logic around the operational records. I keep prices, stock and permissions in that layer so every action passes the same checks, whether it starts with a person or an agent.

I used [shadcn/ui](#) for the interface, giving me established components to build on and more time for the behavior around them. Details open in sheets beside the current record, and a question stays with the work it concerns. Closing a sheet does not cancel the task. Coming back to it does not start the same action again.

The point is to give the team less to chase. A customer knows when to expect the delivery. A supplier has confirmed the extra chairs. The crew can see what needs loading. The useful work is in those details.

The design and engineering decisions are closely tied here. A delivery approval needs to show the customer's message, but it also needs to notice if the booking changed while that message was being reviewed. I work through both sides of that interaction: what a person needs to understand, and what the system has to guarantee.

If you're hiring a design engineer or have an operational problem like this, I'd be happy to talk.