PRODUCT BLUEPRINT

ShiftLine

Staff scheduling for independent restaurants — built to run in a week, not a quarter.

A pre-validation product blueprint — market, model, risks, and a validation plan before you build.

SECTIONS
17 sections

DATE
June 10, 2026

PREPARED BY
Foray

BLUEPRINT OVERVIEW

Pre-Validation

Blueprint stage at time of generation

DOMAIN RISKS

HIGH Long Sales Cycles

MEDIUM Enterprise Procurement Friction

RESEARCH SUMMARY

3 external sources retrieved (1 primary · 2 secondary)

Primary: authoritative or official sources. Secondary: industry and commercial sources.

SOURCING

Key figures are marked: (source: …) cited from an external source · (est.) calculated estimate · (directional) indicative, no public source.

Contents

Executive Summary

Executive Summary: Restaurant Staff Scheduling SaaS

One-Page Brief

Problem

Independent restaurant operators (1–3 locations, 5–30 staff) spend 60–90 minutes weekly building schedules on Google Sheets and managing last-minute callouts via group text. When coverage fails, the cost is operational — missed shifts, understaffed services, manager stress. Existing tools are either enterprise-priced (HotSchedules, Fourth), horizontally positioned with per-user billing that punishes high-turnover environments (When I Work, Deputy), or free tiers that lack the SMS notification layer the workflow actually requires (Sling, Homebase).

Solution

A flat-rate SaaS that lets an owner-operator build and publish a complete weekly schedule — with SMS confirmation to every staff member — in under 20 minutes on a phone. Three pricing tiers:

- *Starter:* Free, up to 8 staff, 1 location (conversion lever)
- *Operator:* $49/month, up to 30 staff, 1 location (primary revenue tier)
- *Manager:* $89/month, up to 3 locations, 60 staff, cross-location dashboard

Market

- Global B2B SaaS market valued at USD 497.41 billion in 2025, projected to reach USD 634.39 billion in 2026 [source: mordorintelligence.com], with AI-driven automation and vertical SaaS solutions as primary 2026 growth vectors [source: parselab.com]
- Year 1 target: 100–300 paying customers H $323K ARR (est.)
- Primary ICP: ~500,000+ independent US restaurant operators using manual scheduling methods (directional)

Business Model

- Flat-rate monthly subscription; no per-seat pricing
- Estimated ARPU: $55/month (est.)

- Estimated CAC: $250–$350 blended (est.)
- Estimated LTV at 7% monthly churn: $770; LTV:CAC ratio 2.6:1 (est.)
- Kill condition at 10% monthly churn for two consecutive months (LTV collapses to $385) (est.)

Stack

- *Frontend/API:* Next.js on Vercel
- *Database/Auth:* Supabase (PostgreSQL + RLS + Realtime)
- *Background Jobs:* Trigger.dev
- *SMS:* Twilio (A2P 10DLC)
- *AI:* Claude API (coverage ranking, conflict explanation) with rule-based fallbacks
- *Analytics:* PostHog (18 tracked events)
- *Payments:* Stripe Checkout

Critical Risks (summary — detailed below)

- Zero switching cost back to spreadsheets means every product failure is a potential permanent churn event
- Unit economics are viable but fragile — a CAC above $350 or churn above 10% breaks the model (est.)
- Incumbent platforms (Toast, Square, 7shifts) can replicate the core differentiators (flat pricing, simplified UX) within a single product cycle

---

Top 3 Opportunities

1. Flat-Rate Pricing in a Per-Seat Market

Every direct competitor except Homebase uses per-user or per-location pricing. In a segment where staff turnover routinely exceeds 70% annually (directional), per-seat pricing creates unpredictable monthly bills — exactly the kind of friction that prompts operators to cancel. Flat-rate pricing at $49/month is a structural differentiator that is immediately legible to Maria during a 30-second price comparison. *Action:* Lead every sales conversation with the pricing model before features.

2. SMS Confirmation as the Core Value Wedge

No competitor in the competitive landscape makes "staff confirmed receipt" the primary outcome of schedule publishing. Existing tools send notifications; this product makes confirmation visible and actionable. The difference matters: a manager who knows every staff member received and viewed their schedule has a qualitatively different level of confidence than one who pressed "send" in a group text. This single feature — visible delivery status per staff member — is the product's most defensible differentiator in the first 12 months because it directly solves the documented failure mode (group text reply ambiguity) without requiring any new

behavior from staff. *Action:* Design the Publish & Confirm screen so delivery status is the first thing visible after publishing, not a secondary panel.

3. Referral Velocity Within Local Restaurant Communities

The lowest-CAC acquisition channel — estimated at $100–$200 per customer (est.) — is peer referral through local Facebook groups, Reddit communities (r/KitchenConfidential, r/restaurantowners), and local food service associations. Restaurant operators trust other restaurant operators over software sales pitches. A single converted owner-operator who posts "this saved me an hour on Sunday" in a local Facebook group of 400 restaurant owners is more valuable than 10 cold outbound emails. *Action:* Build a referral tracking mechanism and a shareable "I use this" asset before the fifth pilot customer is onboarded.

Top 3 Risks

1. Zero Switching Cost in Both Directions

The risk: Maria can return to Google Sheets in 30 seconds with zero data loss and zero migration cost. There is no lock-in. The product must be reliably better every single week — not just at onboarding. A single SMS batch failure on a Sunday evening, a staff member who can't open the link, or an onboarding session that runs 35 minutes instead of 20 can permanently end a customer relationship.

Mitigation from prior sections:

- Achieve reliability before growth: resolve Open Issues C-02 (Twilio A2P 10DLC registration) and C-08 (Stripe webhook handler) before the first paid pilot
- Validate the 20-minute onboarding claim with observed usability testing (Open Issue I-01) before it appears in marketing
- Define explicit fallback states for the Coverage screen (Open Issue I-04) so the app never returns a blank or broken state at high-stress moments
- Instrument lastschedulepublisheddaysago as an early churn predictor; act on it before churn is confirmed

2. Unit Economics Viable Only at Best-Case Assumptions

The risk: The 2.6:1 LTV:CAC ratio (est.) is described in the Monetization section itself as "viable but fragile." It assumes CAC at $250–$350 (est.), ARPU at $55 (est.), and monthly churn below 7%. Real cold outbound CAC to non-technical, time-constrained restaurant owners may run materially higher. Free-to-paid conversion may be lower than modeled. If any two of these assumptions are off simultaneously, the model breaks.

Mitigation from prior sections:

- Apply Kill Condition K5 (CAC exceeding 12 months of subscription revenue) rigorously at Month 9 — do not adjust the threshold retroactively
- Prioritize referral and POS integration channels (lower CAC) over cold outbound before CAC is empirically established
- Annual prepay at 15–20% discount (planned for Phase 5+) should be tested earlier if monthly churn exceeds 5% in the pilot cohort — it improves LTV without requiring a price increase

3. Incumbent Platforms Can Replicate the Differentiators

The risk: Toast has an existing scheduling feature and Maria's daily payment processing relationship. 7shifts already has a free tier for up to 10 employees and direct POS integrations. Square has native team management. The product's moat — simpler UX, flat pricing, mobile-first design — is a set of product and pricing decisions, not a technical barrier. Any of these players can replicate the model in one product cycle if they decide independent restaurants are worth focused investment.

Mitigation from prior sections:

- The 12–18 month timing window identified in the Product Brief is a countdown clock, not a moat. Use it to establish switching costs that do not currently exist: historical schedule data, staff profiles with preferences and qualifications, integration with the operator's POS
- Pursue POS integration partnerships (Toast, Square) as a distribution channel before Toast or Square builds this natively — a certified integration partner is harder to displace than a standalone competitor
- Do not position against 7shifts or Toast in marketing; position against spreadsheets. Avoid being in the comparison set for features the incumbents win on (breadth, enterprise compliance, integrations)

MVP Scope: Phase 1 — What to Build (6 Weeks)

Build exactly the workflow that delivers the North Star metric: *schedule published ’ staff SMS received ’ manager sees confirmation.* Nothing beyond this scope ships in Phase 1.

Must-Have (blocks pilot launch if missing)

- *Authentication:* Email/password signup, Supabase Auth, JWT in httpOnly cookies, brute-force protection (5 failed attempts / 15 min lockout)
- *Roster setup:* Add staff manually with name, phone (E.164 validated at input), position, and max weekly hours; CSV import optional in Phase 1
- *Schedule builder:* 7-column weekly grid (day-column layout on mobile <480px); tap-to-assign shifts; copy-last-week shortcut; live hour totals per staff member; auto-save on every change
- *Publish & Confirm:* Pre-publish summary screen; one-tap SMS send via Twilio; real-time delivery status per staff member (sent / delivered / failed); no modal — status visible inline

- *Staff schedule view:* Publicly accessible via 32-byte random token (no login required); vertical card list by day; "I can't make this shift" action; 7-day token expiry
- *Callout coverage:* Ranked available staff list (not scheduled, below OT threshold, prior experience on position); one-tap notify; explicit fallback states when zero candidates or all at OT
- *Stripe billing:* Checkout for Operator tier ($49/month); webhook handlers for subscription created, cancelled, payment failed, invoice paid; free tier hard cap at 8 staff with inline upgrade prompt

Must-Have Security/Compliance (blocks pilot launch if missing)

- RLS policies written and CI-tested for all tables holding customer or staff data (resolves Open Issue C-01)
- Twilio A2P 10DLC registration initiated; no paid pilots on unregistered numbers (resolves Open Issue C-02)
- DPAs signed with Supabase, Twilio, Vercel, PostHog, Anthropic (resolves Open Issue C-03)
- Privacy policy and Terms of Service live before first pilot agreement signed (resolves Open Issue C-04)
- Incident response process documented; at least one automated error rate alert configured (resolves Open Issue C-05)
- Anthropic API terms confirmed for staff data; rule-based fallback implemented if terms are unfavorable (resolves Open Issue C-06)
- Database migration rollback runbook written before first production migration (resolves Open Issue C-07)

Explicitly Out of Scope for Phase 1

- Multi-location (Manager tier) support — defer to Phase 2; document data model decision before writing any schema
- Labor cost reporting and payroll export
- POS integration
- AI coverage ranking — use rule-based ranking in Phase 1; Claude API introduced in Phase 2 after rule-based behavior is validated
- TOTP 2FA (Manager tier only — not needed until that tier exists)
- Annual prepay pricing

WCAG 2.1 AA Non-Negotiables (applies to all Phase 1 screens)

- 4.5:1 color contrast minimum on all text; 5.5:1 on staff schedule view (real-device tested)
- 44×44px minimum touch targets
- Programmatic label-input association on all form fields
- Semantic HTML table structure on schedule grid
- Full keyboard alternative to drag-and-drop on schedule builder

---

Phase 2 Roadmap (Months 3–6)

Goal: Retain 20 paying customers with a weekly publish rate e80%; validate Kill Condition K3 (5+ customers retained after 60 days).

Milestone	What It Is	Why Now
Multi-location account model	accounts ’ locations data model shipped; Manager tier ($89/month) live; Devon persona served	Required before any 2–3 location customer onboards; schema cannot be migrated cleanly after data exists
Claude API coverage ranking	Replace rule-based ranking with AI-assisted suggestions; rule-based fallback retained	Validate AI accuracy against real callout data from Phase 1 pilots
Staff opt-out STOP handling	Twilio STOP ’ smsoptedout: true flag; manager in-app notification on next login (Open Issue I-02)	Prevents silent delivery failure becoming a churn event
POS integration (1 platform)	Toast or Square read-only integration for roster import	Reduces onboarding friction; establishes distribution partnership before Toast builds natively
Free-to-paid upgrade flow	Defined trigger conditions, copy, and screens for free tier conversion (Open Issue I-05)	Free tier launched in Phase 1 but conversion path not instrumented; required before free tier is a meaningful channel
Idempotency enforcement on all SMS routes	MessageSid-keyed deduplication on Twilio webhook handler (Open Issue I-06)	North Star metric (SPSC-WAA) is unreliable if duplicate confirmations are counted

Phase 3 Roadmap (Months 6–12)

Goal: 100+ paying customers, $323K ARR run rate (est.), Kill Conditions K4 and K5 assessed.

Milestone	What It Is	Why Now
Labor cost reporting	Actual vs. scheduled hours, overtime cost summary, exportable CSV	Unlocks Devon's cost-reduction framing; required for annual budget conversations with owner-operators

Referral program	Tracked referral links, in-product prompt after 3rd consecutive published schedule	Converts the lowest-CAC channel into a systematic acquisition mechanism
Annual prepay pricing	15–20% discount for annual commitment	Reduces involuntary churn from monthly payment failures; improves LTV if monthly churn is above 5%
Second POS integration	Whichever platform was not integrated in Phase 2	Expands distribution; reduces dependence on cold outbound
SOC 2 Type I readiness assessment	Gap analysis only; not certification	Required for any enterprise-adjacent prospect; establishes trust signal for operators nervous about a startup holding staff PII [source: termsfeed.com]
Payback period and CAC audit	Formal review of empirical CAC vs. $250–$350 estimate (est.) across all channels	Kill Condition K5 assessment at Month 9; determines whether outbound investment continues or pivots to referral/POS-led

First Action: Next 48 Hours

Initiate Twilio A2P 10DLC brand and campaign registration today.

This is the only action on the critical path that cannot be compressed by working harder — it has an external approval queue with a 2–6 week processing window (directional). Every other critical launch blocker (RLS policies, DPAs, privacy policy, Stripe webhooks) can be resolved in parallel by the founding team in days. The A2P registration cannot. If this is not started in the next 48 hours, the pilot launch date moves by however many days it was delayed.

What to do:

- Log in to the Twilio console and complete A2P 10DLC brand registration (legal business name, EIN, business type)
- Create a campaign use case: "Scheduling notifications — staff receive weekly schedule and shift confirmation links from their employer"
- Assign your Twilio phone number to the campaign
- Document the submission date, campaign SID, and expected approval date in the project's compliance log

- Set a calendar reminder 3 weeks out to check approval status; define the pilot delay condition (if not approved by pilot start date, restrict pilot to 5 verified test numbers)

Key Risks and Critical Challenges

These are extracted from the Adversarial Review and presented as honest structural risks, not items to plan around.

The Reversion Path Is Always One Tap Away

The adversarial case is precise: Maria's Google Sheet has a 100% uptime record in her personal experience. It requires no new behavior from staff, no subscription fee, and no dependency on a startup that may not exist in 18 months. At $49/month and an estimated LTV of $770 (est.), the product cannot survive even a moderate rate of single-event churn. One bad Sunday — a failed SMS batch, a staff member who can't open the link, an onboarding session that runs long — is not a support ticket. It is a $770 revenue loss and a $300 CAC restart (est.). The product has to earn its place in the workflow every week, not just at signup. There is no technical or contractual mechanism to prevent reversion. Reliability is the only defense.

The Price May Be Too High for the Buyer and Too Low for the Business Simultaneously

Independent restaurant operators at the $400K–$1.5M revenue range operate on thin margins (directional). The $49/month price point exists in a purchasing environment where every recurring software cost competes with food costs, labor, and POS fees. At the same time, the Monetization section's own analysis describes the 2.6:1 LTV:CAC ratio (est.) as "viable but fragile" — meaning the business doesn't have room to acquire customers more expensively or retain them less effectively than modeled. Both pressures are real and both point in the same direction: the price ceiling and the price floor may already be in conflict.

The Timing Window Is a Countdown, Not a Moat

The Product Brief's "12–18 month timing advantage" before incumbents push downmarket is a judgment call, not a verified market gap. Toast already has scheduling features. 7shifts already has a free tier covering the smallest operators. The differentiators being claimed — flat-rate pricing, simpler UX, mobile-first design — are product and pricing decisions that a well-resourced competitor can replicate in a single development cycle. If Toast or Square decides that independent restaurant scheduling is worth a focused sprint, they enter with Maria's existing payment processing relationship and daily active usage. The product's response cannot be to build faster. It must be to establish switching costs (historical data, staff profiles, POS integration) before that decision is made.

The global B2B SaaS market is projected to reach USD 634.39 billion in 2026 [source: mordorintelligence.com], with vertical SaaS solutions capturing disproportionate growth as operators move away from horizontal tools toward purpose-built software [source: parselab.com]; independent restaurants — a fragmented, underserved segment that every major scheduling platform has deliberately under-prioritized — represent a direct entry point into this shift. This product is a flat-rate ($49/month) scheduling SaaS built specifically for independent restaurant operators replacing spreadsheets and group texts, delivering its core value — a complete published schedule with per-staff SMS confirmation visible to the manager — in a single Sunday-night session on a phone, with no configuration overhead and no per-seat pricing that punishes the high-turnover staffing reality these operators already manage. The traction plan is a gated validation roadmap: 15 discovery interviews in Month 1, 5 paid pilots at $29–$49/month by Month 3, 20 retained paying customers by Month 6, and 100+ customers at $323K ARR (est.) by Month 12, with five explicit kill conditions that trigger a controlled shutdown rather than continued investment if the market does not validate willingness to pay at viable unit economics.

Consistency Notes

The following potential contradictions were detected across sections:

Contradiction Identified

Section: Financial Projections vs. Executive Summary

Specific Contradiction:

- *Executive Summary* states: "Year 1 target: 100–300 customers H $323K ARR"
- *Financial Projections* shows: "Year 1: $34K revenue... 150 EoY customers"

The Math Conflict:

- 300 customers × $55/month ARPU (from Monetization) × 12 months = $198K ARR (not $323K)
- To reach $323K ARR requires ~490 customers at $55/month ARPU
- Financial Projections projects only 150 Year 1 customers, yielding ~$99K ARR, yet shows "$34K revenue" (unexplained $65K discrepancy)

Additional Related Issue:

- *Monetization* lists Year 1 market sizing as "100–300 paying customers = ~$323K ARR"
- *Financial Projections* Year 1 row shows only 150 EoY customers with $34K revenue, contradicting both the upper bound (300) and the math ($323K)

The Executive Summary's Year 1 ARR target is mathematically impossible given the stated ARPU and customer count ranges. Financial Projections shows a far more conservative (and internally inconsistent) model.

02 Product Brief

Product Brief: Restaurant Staff Scheduling SaaS

One-Liner

> *Scheduling software built for independent restaurants that replaces the group text.*

Elevator Pitch

Independent restaurants with 5–30 staff lose hours every week to manual scheduling — spreadsheets, group texts, and last-minute scrambles when someone calls out. This tool gives owners and managers a purpose-built scheduling workflow: build the week's shifts, assign staff, and notify the team — all in under 15 minutes. Unlike enterprise tools built for chains, it is priced and designed for the operator who is also cooking on the line.

Problem Being Solved

Independent restaurant operators face a recurring, high-cost scheduling problem with no right-sized solution:

- *Time cost:* Building a weekly schedule manually takes 1–3 hours per manager per week, often done late at night after service
- *Coverage failures:* Last-minute callouts are handled via group text or phone chains — a system that frequently fails and leaves shifts uncovered
- *Overtime and compliance exposure:* Without visibility into weekly hours across staff, managers accidentally create overtime liability they only discover at payroll
- *Existing tools are mis-sized:* Enterprise tools like HotSchedules and 7shifts are feature-heavy and priced for multi-location operators; free tools like spreadsheets have no notification layer and break under real-world shift volatility

The result is a recurring operational tax: manager time wasted, staff frustrated by unclear schedules, and owners carrying untracked labor costs.

Who It's For

Primary user: The owner-operator or floor manager of an independent restaurant — 1 to 3 locations, 5 to 30 hourly and tipped staff. This person is not a software buyer by default. They are time-starved, device-switching (phone to laptop mid-service), and deeply skeptical of tools that require training. They will adopt a product only if it is faster than their current method within the first week.

Not (yet) for: Multi-location chain operators, enterprise HR buyers, or restaurants with unionized staff requiring complex contract rules.

Why Now

Three forces have converged to make this a viable product window in 2025–2026:

- *Accelerating AI-driven automation adoption in B2B SaaS* makes it credible to automate shift-fill suggestions and coverage gap alerts at a price point previously impossible for SMBs — tools that once required a dedicated HR system can now be built lean [verified: parselab.com]. [source: parselab.com]
- *The B2B SaaS market is expanding rapidly* — valued at USD 497.41 billion in 2025 and projected to reach USD 634.39 billion in 2026 [source: mordorintelligence.com] — increasing operator familiarity with and tolerance for SaaS subscription spending, even at the SMB tier.
- *Post-pandemic labor volatility* has made the old informal scheduling habits — whiteboard, group text — structurally unreliable. Staff turnover in food service remains high, meaning managers are constantly onboarding new hires into the schedule. A tool that handles this repeatably has more durable value than it did when staffing was stable.

The window is open, but it will not stay open. As 7shifts and HotSchedules continue to push downmarket, or as POS platforms like Toast and Square deepen their native scheduling features, the viable entry point for an independent tool narrows. This is a 12–18 month timing advantage, not a permanent structural gap.

Top 3 Assumptions to Validate

These are the three claims this product depends on most. Each has a direct validation path in the Validation Roadmap.

#	Assumption	Why It Matters	Where It Gets Tested
A1	Independent restaurant managers experience scheduling pain severe enough to pay e$49/month to solve it — not just acknowledge it	If pain is tolerated rather than felt as costly, willingness to pay collapses and Kill Condition K1 is unreachable	Phase 1 (Discovery) and Phase 2 (Willingness to Pay)
A2	A minimal scheduling workflow — build shifts, assign staff, send notifications — is sufficient to displace the current method in real operating conditions	If operators need shift-swap automation, compliance alerts, or POS integration before they'll switch, the MVP scope is far larger than assumed and pilot retention will fail	Phase 3 (Pilot Proof)
A3	At least one acquisition channel produces a Customer Acquisition Cost with a payback period under 12 months at the $49–$99/month price point	If no channel achieves viable unit economics at SMB price points, the business requires either a price increase, an enterprise pivot, or a distribution partner to subsidize CAC (directional)	Phase 4 (Retention and Channel Proof)

Top 3 Risks

#	Risk	Likelihood	Impact	Mitigation
R1	Incumbent commoditization — Toast, Square, or 7shifts deepens native scheduling at no additional cost to existing restaurant customers, eliminating the willingness-to-pay window	Medium	High	Validate before building. If POS-bundled scheduling is already sufficient for the target segment, this product has no viable entry. Check in Phase 1 interviews whether operators are already using a POS-native scheduling feature and why they are not sticking with it.

R2	Habit inertia beats product quality — Operators use the tool for 2 weeks, revert to the group text under pressure, and churn before the value compounds	High	High	Addressed directly in Phase 3. The specific failure mode is week 3 reversion under operational stress. The mitigation is white-glove onboarding in the first two scheduling cycles — not feature-building — to get the habit set before the first real pressure event hits.
R3	Price ceiling is structurally too low for viable unit economics — Independent restaurants operate on thin margins (typically 3–9% net) and have strong resistance to any new recurring expense	Medium	High	The $49/month price point must be validated against actual current spend and time-cost awareness in Phase 2. If operators cannot articulate that their current method costs them more than $49/month in manager time, the price ceiling is real and must be solved through channel economics (lower CAC via partnership) or a market segment shift.

Market Sizing

FOR INTERNAL PLANNING

View A — Bottom-Up Model

For internal planning — every assumption listed. Numbers are planning estimates, not projections.

Input	Value	Assumption Source
Independent US restaurants with 5–30 staff (scheduling pain is acute enough to generate willingness to pay at this size)	150,000–250,000 restaurants	[ESTIMATED — validate through NAICS data and Phase 1 discovery; exact count of non-chain food service establishments in this staff band is not available in verified research data]
Realistic addressable subset: operators aware of the problem, willing to trial SaaS, and reachable through identified channels	5%–12% of the above (7,500–30,000 restaurants)	[ESTIMATED — validate through customer discovery]
Average revenue per customer per year at $49–$99/month	$588–$1,188/year	(est.)

Year 1 realistic addressable market	$4.4M–$35.6M	Calculated from above rows at low and high ends

Planning note: The Year 1 realistic target — assuming 50–150 paying customers by end of Phase 4 — is $29,400–$178,200 ARR. (est.) This is not a market size claim; it is a build-or-kill decision threshold. The full addressable range above represents the ceiling if the channel and retention hypotheses prove out over 24–36 months.

FOR INVESTOR CONVERSATIONS

View B — Top-Down Narrative

For investor conversations — standard TAM/SAM/SOM framing. Numbers are model-generated estimates. Do not cite without independent verification.

TAM — Total Addressable Market The global restaurant management software market — encompassing scheduling, POS, inventory, and workforce tools — is a subset of the broader B2B SaaS market valued at USD 497.41 billion in 2025. [source: mordorintelligence.com] Restaurant-specific workforce management software is estimated at $3B–$5B globally. (directional) For planning purposes, a conservative TAM of $50B is used to represent the global market for all restaurant operations software, acknowledging this overstates the scheduling-only segment. (est.)

SAM — Serviceable Addressable Market Narrowed to US-based independent restaurants with more than 10 staff who are currently using either a manual method (spreadsheet, group text) or an entry-level scheduling tool and could plausibly be converted to a $49–$99/month SaaS solution. Estimated at $2B–$4B based on the addressable restaurant count above multiplied by annual revenue per customer. (est.) This excludes enterprise chains, restaurants with fewer than 5 staff (insufficient scheduling complexity to justify the spend), and markets outside North America where regulatory and payment infrastructure assumptions differ.

SOM — Serviceable Obtainable Market (Year 1) Target: 100–300 paying restaurants at $49–$99/month average = $290,000–$356,400 ARR at end of Year 1, representing less than 0.02% of SAM. (est.) This is not a conservative estimate — it is the realistic outcome if Phases 3 and 4 of the Validation Roadmap succeed and one acquisition channel achieves viable economics. Achieving even 50 paying customers by month 9 would validate the retention and unit economics hypotheses enough to support a seed fundraise.

KEY TAKEAWAYS

- The pain is plausible but unconfirmed at price: Independent restaurant scheduling is a real operational friction point, but willingness to pay e$49/month from operators with thin margins is the single highest-risk assumption — and it must be validated through direct conversation before any build begins (see Kill Condition K1 and Validation Roadmap Phase 2).
- Timing creates a narrow entry window: Converging forces — post-pandemic labor volatility, SMB SaaS adoption growth [source: mordorintelligence.com], and AI-enabled automation at lower price points [source: parselab.com] — make 2025–2026 a credible moment to enter, but incumbent POS platforms bundling scheduling natively could close this window within 12–18 months.
- Unit economics are the real kill condition: Even with validated pain and retention, the business only works if at least one acquisition channel produces a CAC with a payback period under 12 months at SMB price points — a threshold with no verified benchmark in this specific segment (directional), making Phase 4 channel testing non-negotiable before any capital is committed to scaling.

03 Competitive Landscape

Competitive Landscape

Direct Competitors

1. 7shifts

7shifts is the closest named competitor and the most important one to understand in detail. Founded in 2014, it is a restaurant-specific scheduling platform that has grown by moving upmarket — its current positioning, pricing architecture, and feature depth are all calibrated for restaurant groups with 5–50+ locations, a dedicated manager class, and the operational infrastructure to configure and maintain a platform. It is the brand name most likely to come up when an independent restaurant operator is asked "have you ever looked at scheduling software?" — and it is also the product most likely to be evaluated and rejected by the primary ICP before she reverts to her spreadsheet.

What it does well: Deep restaurant verticalization (role-based scheduling, tip pooling, labor compliance by state, manager log), strong POS integrations (Toast, Square, Lightspeed, Clover), and a well-reviewed mobile app. Its feature set is genuinely comprehensive for the use cases it targets.

Where it breaks for the primary ICP: The Entree plan at approximately $29.99/month per location is the entry point for paid features, but the free plan caps at 10 employees — insufficient for a restaurant of 15–25 staff. The tier complexity (Entree, The Works at ~$69.99/month, Gourmet at ~$135/month) creates evaluation overhead that Maria-type operators experience as "I couldn't figure out what I actually needed." Time-to-first-schedule is high because the onboarding flow requires configuring locations, roles, and positions before a schedule can be built. The product was designed assuming the user has 30 minutes and a second screen — neither of which Maria has on a Sunday night.

2. HotSchedules (now Fourth)

HotSchedules was the dominant restaurant scheduling platform for over a decade before being acquired by Fourth (a UK-based hospitality workforce management company) in 2019. The combined platform now operates under the Fourth brand and targets enterprise restaurant groups, hotel chains, and multi-unit food service operators. Its pricing is enterprise-negotiated — no public rate card — and effectively unavailable to operators at the independent restaurant scale.

What it does well: Deep integration with enterprise POS systems (Aloha, Micros), compliance reporting for large organizations, labor forecasting tied to historical sales data. For a 50-location chain with a VP of Operations, this is a credible enterprise solution.

Where it breaks for the primary ICP: It is not a competitive threat to this product in the independent restaurant segment. No independent operator running one location with 15 staff is evaluating HotSchedules. Its relevance here is as a brand name that shapes the category's reputation — many independent operators who have heard of "restaurant scheduling software" associate it with this category of expensive, complex, enterprise tool, which is part of why they default to spreadsheets. The perception problem ("scheduling software is for chains") is in part a HotSchedules inheritance.

---

3. Homebase

Homebase is a horizontal small business scheduling and time-tracking platform. It does not target restaurants specifically — its customer base spans retail, food service, salons, fitness studios, and any other hourly-workforce business. It is currently the most direct structural overlap with this product's free tier: Homebase's free plan is genuinely functional for a single location with unlimited employees, and it includes basic scheduling and a time clock.

What it does well: A genuinely useful free tier that has made it a default tool for cost-sensitive small business operators. The product is well-designed at the core scheduling layer and integrates with Square and Gusto. Its paid tiers ($20/month for Essentials, $48/month for Plus, $80/month for All-in-One per location) add team communication, time-off management, and HR features.

Where it breaks for the primary ICP: The free tier does not include SMS notifications to staff — the single feature that separates "the schedule is built" from "the team knows the schedule exists." Homebase's notification layer lives on paid plans, and the transition from free to paid requires navigating a tier structure that adds HR features Maria does not need before unlocking the notification feature she does. The horizontal positioning also means the product's UX is built for "any small business with hourly workers" — it does not use restaurant language, does not understand shift structure in restaurant terms (FOH/BOH, double shifts, tipout implications), and does not position around the specific callout and coverage problems that define restaurant scheduling pain. Maria using Homebase free experiences it as a scheduling tool that solved half her problem and charged her for the rest.

---

4. When I Work

When I Work is a horizontal shift scheduling platform primarily serving retail, food service, and healthcare. Its distinguishing characteristic is per-user pricing ($2.50/user/month at entry, scaling to $6/user/month for advanced features), which creates the billing volatility described throughout the Monetization section. It is a

functionally capable product with a strong mobile app, shift swap features, and a notification layer, but its pricing structure makes it economically unpredictable for the high-turnover restaurant environment.

What it does well: Clean mobile UX, shift trading functionality, time clock integration, and a well-established brand in the shift-worker scheduling space. Its onboarding is faster than 7shifts because it does not require restaurant-specific configuration.

Where it breaks for the primary ICP: Per-user pricing at a restaurant with 18–22 active employees and 30–40% annual turnover creates a monthly invoice that fluctuates by $25–$60 depending on whether a new hire was added this week or whether the seasonal staff came back for the summer rush. Maria does not track headcount as a billing variable — she tracks it as a scheduling variable. The billing model asks her to think about software cost every time someone quits or is hired, which introduces the exact friction that flat-rate pricing eliminates. The product is also not restaurant-specific: it does not address callout coverage suggestions, does not understand overtime in the context of tipped employees, and does not position against the "my team ignores the group text" problem in restaurant-specific language.

---

5. Sling

Sling is a scheduling and communication platform that originated in the restaurant sector (launched by a restaurant owner in 2015) before expanding to retail and other hourly-work industries. It offers one of the most accessible free tiers in the category: basic scheduling and shift communication for unlimited employees at one location with no time limit. Its paid plans start at approximately $1.70/user/month (Premium) and $3.40/user/month (Business), which keeps it below When I Work in cost but reintroduces per-user variability.

What it does well: The free tier is genuinely functional for operators who need basic schedule building and internal team communication. The product's origins in the restaurant sector mean its UX reflects more restaurant-native thinking than Homebase. The communication layer (in-app messaging, task assignment, newsfeed) is stronger than most competitors at this price point.

Where it breaks for the primary ICP: The per-user paid tier creates the same turnover-billing friction as When I Work. The free tier lacks overtime alerts and the callout coverage suggestion layer. The communication features, while useful, add complexity that the primary ICP does not need — Maria does not want a team newsfeed, she wants confirmation that everyone saw the schedule. Sling's UX surface area is broader than the problem this product is solving, which makes the time-to-first-value longer than it should be for an operator whose primary pain is "building and communicating the schedule," not "managing all team communication."

---

6. Deputy

Deputy is a workforce management platform that competes primarily at the 20–200 employee SMB and mid-market level across multiple industries (restaurants, retail, healthcare, hospitality). Its pricing starts at

approximately $4.50/user/month for scheduling and scales with feature tiers. It offers strong compliance features (break tracking, award interpretation for international markets, labor law alerts) and integrations with major payroll systems.

What it does well: Deputy's compliance and payroll integration depth is best-in-class for SMB workforce management. For a restaurant group in a state with complex overtime rules (California, New York), Deputy's labor compliance layer is genuinely differentiated. Its integrations with Xero, QuickBooks, ADP, and Gusto make it attractive to operators who want scheduling connected to payroll.

Where it breaks for the primary ICP: Per-user pricing, feature complexity beyond the scheduling use case, and positioning as a "workforce management" platform rather than a scheduling tool all signal to Maria that this product was not built for her. The compliance depth that makes Deputy valuable for a 50-person restaurant group is invisible overhead for a 15-person independent operator in a state without complex award interpretations. Deputy is also geographically strongest in Australia and the UK, where its compliance features address those markets' labor award systems — its U.S. small restaurant penetration is lower than its global profile suggests. (directional)

Feature Comparison Matrix

The matrix below covers the eight features most directly relevant to the primary ICP's purchase decision. Features are assessed based on publicly documented product capabilities and known tier restrictions. Where a competitor offers a feature only on paid or higher-tier plans, this is noted.

Feature	7shifts	HotSchedules (Fourth)	Homebase	When I Work	Sling	This Product
Flat-rate per-account pricing	per location)	enterprise quote)	per location)	per user)	per user on paid)	
SMS notifications to staff on schedule publish	paid tiers)	enterprise)	Partial (paid only)	paid tiers)	Partial (paid only)	Operator tier, $49/month)
Overtime / hours-approaching alerts	paid tiers)		paid tiers)	paid tiers)	Partial (paid only)	Operator tier)
Callout coverage suggestions		enterprise config)		Partial (open shift posting)	Partial (open shift posting)	Operator tier)
Restaurant-specific design and language			horizontal)	horizontal)	Partial	
POS integration (Square / Toast)	strong)	enterprise POS)	Square)	Square)		Partial (roster import planned)
Multi-location scheduling view	paid tiers)		paid tiers)	paid tiers)	paid)	Manager tier, $89/month)
Functional free tier (notifications included)	free caps at 10 employees, no SMS)		free excludes SMS)		Partial (no SMS on free)	Partial (free tier up to 8 staff, no SMS — SMS unlocks at Operator)

Reading this matrix honestly: This product does not win across every row. On POS integration depth, 7shifts is materially ahead — its Toast and Square integrations are deeper and more established than what is planned for the MVP. On multi-unit enterprise features, HotSchedules and 7shifts at their higher tiers have genuine capability advantages. The competitive claim is not "this product has more features." It is "this product makes the features that matter to a 15-person independent restaurant accessible in a single session at a predictable monthly cost."

Pricing Comparison

The table below uses publicly available pricing information current at the time of writing. SaaS pricing changes frequently — verify against each competitor's current pricing page before using this table in any customer-facing comparison.

Competitor	Pricing Model	Entry Price	Notes
7shifts	Per location, tiered	Free (d10 staff, 1 location); ~$29.99/month for Entree	Free tier too restrictive for most independent restaurants (10-staff cap). Paid tiers scale to ~$135/month/location at Gourmet. Multi-location cost multiplies per-location price.
HotSchedules (Fourth)	Enterprise quote-based	Not publicly listed	Requires sales contact. Effectively inaccessible to independent operators. Minimum contract likely in the $200–$500/month range. (directional)
Homebase	Per location, tiered	Free (unlimited employees, 1 location); $20/month for Essentials	Free tier is the competitive threat, not the paid tiers. Essentials ($20/month) unlocks team communication and time-off management but not the full SMS notification layer.
When I Work	Per user	~$2.50/user/month (Attendant); ~$6/user/month (Standard)	At 20 staff, Attendant = $50/month — comparable to this product's Operator tier, but with unpredictable cost as roster fluctuates.
Sling	Per user (paid tier)	Free (basic scheduling, 1 location); ~$1.70/user/month Premium	Free tier is the most generous in the market for raw schedule-building functionality. Per-user paid tier reintroduces cost variability.
Deputy	Per user	~$4.50/user/month (Scheduling only)	At 20 staff, scheduling alone = $90/month. Compliance and payroll integration features add cost. Most cost-effective for operators who need the compliance layer; expensive for operators who only need scheduling.

This Product	Flat-rate per account	$0 (Starter, d8 staff); $49/month (Operator, d30 staff); $89/month (Manager, d3 locations, d60 staff)	Cost is fixed regardless of roster changes. The single most predictable billing structure in this competitive set.

The pricing story in one sentence: At 20 staff, this product's Operator tier ($49/month flat) costs the same as When I Work's per-user Attendant tier and less than Deputy's scheduling-only tier — with no invoice variability when the summer hire starts or the Saturday dishwasher quits.

Our Differentiation by ICP Segment

Maria (Owner-Operator, Primary ICP)

Maria's decision to switch from her Google Sheet is not a feature decision — it is a trust and time decision. She will not spend 45 minutes configuring a new tool to save 20 minutes per week. She will not pay a variable monthly fee that changes when someone new is hired. She will not use software that requires her to call support to figure out why her team is not getting notifications.

Why this product wins for Maria:

The single most important differentiator for Maria is not on the feature list — it is the time-to-first-published-schedule. Every competitor in this table, including Sling's generous free tier and Homebase's clean interface, requires configuration steps before a schedule can be built and sent: location setup, role definitions, employee invite flow, notification permission setup. This product's onboarding is designed to produce one outcome in the first session: Maria publishes a schedule and her team receives an SMS confirmation. That outcome — not a feature, not a dashboard — is what makes her open the app next Sunday instead of reopening her Google Sheet.

The flat-rate pricing is the secondary differentiator for Maria. She does not model software costs against headcount — she makes a monthly budget decision. $49/month is a line item she can approve in the same mental frame as her POS subscription. A per-user fee that changes each month requires a reconciliation decision she does not have bandwidth for.

No competitor at the $49/month price point offers flat-rate pricing, SMS notifications, overtime alerts, and callout coverage suggestions together without requiring a configuration investment that Maria will not make.

Devon (Floor Manager, Secondary ICP)

Devon's value story is different from Maria's. He is not trying to save time — he is trying to avoid the Monday morning conversation about overtime. His purchase decision is justified to his owner as cost prevention, not convenience.

Why this product wins for Devon:

The Manager tier's multi-location view and cross-location conflict detection address Devon's specific pain in a way no competitor does at comparable cost. 7shifts provides multi-location functionality, but its per-location pricing means Devon's owner sees the cost multiplied by location count — at two locations, 7shifts' The Works tier costs approximately $139.98/month. This product's Manager tier at $89/month for up to three locations is approximately 36% cheaper for the two-location use case and 56% cheaper for the three-location use case, at comparable scheduling functionality. (est.)

The labor cost reporting in the Manager tier — hours plus estimated wage cost per week, per location — gives Devon the specific artifact he needs for Monday morning: a printable or shareable summary that shows whether last week's labor percentage was above or below target. No competitor at this price point produces this report without a full payroll integration requirement. A configurable hourly rate field per employee produces the estimate Devon needs without the integration engineering cost that payroll connectivity requires.

Priya (Informal Scheduler, Tertiary ICP)

Priya is not a buyer, but she is the person who will advocate for the product inside the restaurant if the product solves her problem. Her problem is not schedule-building — it is being the unofficial communication hub for everything scheduling-related, on her own time and her own phone.

Why this product wins for Priya:

The Starter free tier's shareable schedule link — viewable on mobile without a login — is the single feature that matters most to Priya. She can share the schedule with her staff without requiring anyone to create an account, download an app, or accept an invitation. The time-off request inbox in the Operator tier moves availability requests off her personal WhatsApp and into a trackable queue. Neither of these features requires Priya to be the account owner — she benefits from the product even if Maria or Devon is the one paying for it.

Priya is also the referral vector most likely to expand the product's reach horizontally. When she leaves this restaurant and takes a shift lead role at another independent restaurant, she will recommend the product she used here because it is the one that made her informal scheduler role manageable.

Positioning Gaps

The competitive analysis reveals three positioning gaps that no current competitor addresses well for the independent restaurant segment.

Gap 1: The notification-included free tier

Homebase and Sling both offer functional free scheduling tiers. Neither includes SMS notifications to staff on schedule publish. This is not an accident — SMS delivery costs money, and free tier products cannot absorb unlimited SMS cost at scale. The result is a free tier gap: the most important feature for an independent restaurant operator (confirmation that everyone received the schedule) is universally paywalled at the point where it matters most.

This product's free tier also excludes SMS notifications. The gap is not in the free tier itself — it is in how the upgrade path is designed. If this product positions the first SMS notification moment as the "this is what the paid tier unlocks" event, it can use the notification gap as the most natural conversion trigger available: the operator publishes a schedule on the free tier, realizes no one got a text, and upgrades immediately because she needs to know the schedule reached her team. (est.) No competitor has designed their onboarding to make that specific friction moment a conversion event.

Gap 2: Callout coverage as a built-in, not a bolt-on

Open shift posting — where a manager posts an available shift and employees self-select to claim it — is a common feature in When I Work, Sling, and 7shifts. What no competitor offers at the independent restaurant price point is proactive callout coverage suggestion: when a shift goes uncovered (because of a callout, not a manager-created opening), the system surfaces the two or three staff members who are available that day, below their weekly hour threshold, and not already scheduled at another time. This distinction — reactive open-shift posting versus proactive coverage suggestion — maps directly to the operational reality of a Saturday morning callout, where Maria does not have time to post an open shift and wait for a response. She needs a list of people to call. No competitor under $100/month produces that list automatically.

Gap 3: Independent restaurant identity as a product signal, not a marketing claim

7shifts' homepage says "Built for restaurants." Homebase says "Works for restaurants." When I Work shows a restaurant worker in a stock photo. None of these are products that were designed exclusively for independent restaurants — they are products designed for general shift-worker industries that have added restaurant marketing language. The gap is not in features; it is in the product's behavior when Maria opens it for the first time. If the product's default role names are "Server," "Bartender," "Line Cook," and "Host" rather than "Employee Type A" or generic "Role 1," Maria knows immediately that someone who understands her operation built this. No competitor at the entry price point has made that default-state design investment. It costs nothing to implement and signals everything about who the product is for.

Indirect Competitors and Substitutes

The most important competitors are not software products.

Google Sheets / Excel (Primary Substitute)

The Google Sheet schedule is the actual incumbent for the primary ICP — not 7shifts, not Homebase. Maria built her scheduling spreadsheet herself, has customized it over two or more years, and knows how to use it. The switching cost from her spreadsheet to any paid tool is not zero: she must rebuild her roster in a new system, train her staff to check a new link instead of the one they have bookmarked, and trust that a $49/month product will not disappear or change its pricing in six months. The Google Sheet asks nothing of her. The competitive strategy against the spreadsheet is not a feature list — it is a first-session experience that produces a demonstrably better outcome (SMS confirmation that the whole team received the schedule) in less time than the spreadsheet takes, in the first session she uses it.

Group Text / WhatsApp (Communication Substitute)

The group text is not a scheduling tool — it is the fallback when the scheduling tool (spreadsheet or software) fails to confirm receipt. Every independent restaurant operator with a staff of 12 or more has a group text that includes scheduling-adjacent communication: "reminder, schedule is posted," "who can cover Saturday at 5?", "heads up, I'm changing the Tuesday close shift." The group text is deeply habituated, costs nothing, and requires zero onboarding. Any scheduling product that does not eliminate the need for the schedule-related group text will be used alongside the group text — not instead of it — which means the product adds complexity without removing the behavior it was supposed to replace.

Toast Scheduling (Bundled POS Feature)

Toast has been building scheduling features into its POS platform, available to existing Toast merchants at no additional software cost. For a restaurant already paying $100–$300/month for Toast processing and hardware, a "good enough" scheduling tool bundled at no incremental cost is a meaningful competitive threat — not because the feature is better than this product, but because the incremental cost is zero and the switching cost from existing Toast usage is lower than switching to a third-party tool. (directional) The Product Brief assesses this risk as medium likelihood and high impact. The response is the integration relationship described in the GTM Strategy section: being in the Toast partner ecosystem before Toast decides to deprecate third-party scheduling is a more defensible position than being outside it when that decision is made.

Square Team (Bundled POS Feature)

Square offers Square Team, a workforce management product that includes basic scheduling, time tracking, and team messaging. For Square-using restaurants on the free or Plus tier, some scheduling functionality is included in the existing POS subscription. The depth of Square Team's scheduling feature — specifically its

notification reliability, overtime visibility, and callout handling — is limited compared to a dedicated scheduling product, but the bundled cost structure makes it a default tool for operators who have not decided to pay specifically for scheduling. The risk mirrors the Toast scenario: not feature competition, but cost-structure competition.

Pen, Whiteboard, and Printed Paper (Physical Substitutes)

A non-trivial segment of independent restaurant operators — particularly in older, owner-operated establishments with low staff turnover and a consistent core team — schedules via a physical whiteboard or printed sheet posted in the back office. These operators are not the primary ICP for an SMS-notification-driven digital product. They are a distinct persona: resistant to digital tools by preference, not by inertia. Attempting to convert a whiteboard operator is a higher CAC investment than converting a spreadsheet operator with lower probability of success.

KEY TAKEAWAYS

- The real incumbent is the Google Sheet, not 7shifts. Most of the target market has never paid for scheduling software and is not making a competitive switch — they are making a first purchase. The competitive question is not "why is this better than 7shifts?" It is "why is this better than the spreadsheet she already knows how to use?" That is a faster and more winnable argument, and it requires a first-session experience designed around that specific comparison, not a feature parity table.
- Flat-rate pricing is the single most structurally differentiated position in this market. Every named direct competitor charges per user, per location, or both — which creates billing variability that is uniquely painful in the high-turnover, seasonally volatile independent restaurant context. No competitor has committed to a flat-rate-per-account model at the independent restaurant price point. If this product builds retention at the Operator tier, that pricing structure becomes the primary reason operators stay when a competitor offers a promotional rate — because predictability has more value than a $10/month discount in an operational context where every variable is already unpredictable.
- Toast and Square bundled scheduling is the highest-impact competitive risk, and the response is integration, not avoidance. The B2B SaaS market is projected to grow to USD 4,441.49 billion by 2034 [source: mordorintelligence.com/industry-reports/b2b-saas-market], a trajectory that structurally incentivizes POS platforms to bundle adjacent software categories to increase their own retention. The window to establish a presence in the Toast and Square partner ecosystems — before either platform decides native scheduling is sufficient for the independent restaurant segment — is finite and should be treated as a time-sensitive GTM priority, not a Phase 5 consideration.

04 Monetization

Monetization & Drops

Recommended Pricing Model

Flat-rate monthly subscription. No per-seat pricing.

The single most important pricing decision for this product is the billing unit. Per-seat pricing — used by enterprise tools like HotSchedules and, at higher tiers, by 7shifts — creates a structural mismatch with independent restaurant operations. Staff rosters turn over constantly. At a restaurant with 15–20 hourly employees, it is common to add two new hires and lose one in a single week. A per-seat model means the bill fluctuates monthly, creates invoice confusion for operators who are not tracking headcount as a metric, and introduces a churn trigger every time a high-turnover period hits. Maria and Devon are not managing seat licenses — they are managing a schedule. The billing unit should reflect that.

Flat-rate pricing removes the "do I add this new hire to the system or wait until she proves out?" hesitation that per-seat billing creates in the onboarding moment. It also makes the value conversation cleaner: the product costs $49/month, full stop, regardless of whether staffing is up or down this week.

This pricing model is benchmarked against the competitive landscape described in the Product Brief:

- *7shifts Entree*: ~$29.99/month for one location, up to 30 employees — the closest true competitor in the independent restaurant segment
- *Homebase*: Free tier plus paid plans from $20–$80/month per location — primarily focused on time tracking, scheduling is secondary
- *When I Work*: ~$2.50/user/month, making it unpredictably expensive at 20+ staff and subject to the per-seat problems described above
- *HotSchedules (Fourth)*: Enterprise positioning, effectively unavailable at the independent restaurant price point
- *Square/Toast native scheduling*: Bundled into existing POS contracts, no additional line item — the competitive threat that most directly undercuts willingness to pay

The pricing strategy positions this product above the free tools (spreadsheets, Homebase free) to establish perceived value, below or equal to 7shifts Entree to remove price as a switching barrier, and structurally simpler than per-seat tools to remove billing friction.

Pricing Tiers

Tier 1 — Starter · $0/month · Free forever

Designed for: Priya (the informal scheduler) and operators evaluating the product before committing. This tier is not a revenue vehicle — it is a trust vehicle and a referral surface.

Feature	Included
Locations	1
Staff roster	Up to 8 employees
Weekly schedule builder	
Shareable schedule link (staff view it on mobile)	
Manual shift assignment	
Time-off request inbox	
SMS/push notifications to staff	
Overtime visibility (weekly hours tracker)	
Callout coverage alerts	
Shift templates (save recurring week patterns)	
Multi-location view	
Manager accounts (additional logins)	
Labor cost reporting	

Reasoning: The 8-staff cap is a deliberate constraint. It is generous enough to let Priya use the product legitimately and let Maria evaluate it with her real roster. It is restrictive enough that a restaurant of 12–15 staff — the operational center of mass for this product — cannot run indefinitely on Starter without hitting the wall. The missing notification layer is the functional upgrade trigger: the moment a staff member says "I didn't know the schedule changed," Starter users feel exactly the pain that Pro solves.

Free-to-Pro conversion assumption: 15–25% of active Starter users convert to Pro within 90 days. (est.) Operators who publish three or more schedules on Starter are the highest-conversion cohort — they have established the habit and hit the staff-size or notification ceiling. Operators who build one schedule and go quiet are unlikely to convert without a re-engagement sequence.

Tier 2 — Operator · $49/month · Recommended tier

Designed for: Maria (owner-operator) and Devon managing a single location. This is the product's revenue center and the tier validated against Kill Condition K1.

Feature	Included
Locations	1
Staff roster	Up to 30 employees
Weekly schedule builder	
Shareable schedule link	
Manual shift assignment	
Time-off request inbox	
SMS notifications to staff on schedule publish	
Push notifications on schedule changes	
Overtime alerts (flags staff approaching 40 hrs mid-week)	
Callout coverage suggestions (who's available to fill this shift)	
Shift templates	
Multi-location view	
Manager accounts	1 additional login
Labor cost reporting	Basic (hours-to-date per employee)

Reasoning: $49/month is the price point validated through the Validation Roadmap's Phase 2 willingness-to-pay test and anchored in Kill Condition K1. The specific features unlocked at this tier — SMS notifications, overtime alerts, and callout coverage suggestions — are the three functional gaps most likely to surface in Phase 1 interviews as the difference between "I could manage with a spreadsheet" and "this solves a problem I can't solve any other way." Notifications are particularly important: they are the feature that transfers the cognitive burden of "did everyone see the schedule?" away from the operator.

At $49/month against 7shifts Entree at ~$29.99/month, this product is priced $19/month higher. That premium must be justified by faster time-to-schedule, a better mobile experience, and a simpler interface — not a longer feature list. If Phase 2 interviews reveal that $49 creates resistance and $29 does not, the Operator tier should be repriced to $29 and the Scale tier should absorb the revenue gap.

Tier 3 — Manager · $89/month

Designed for: Devon (floor manager at 2–3 location restaurant groups) and owner-operators who have grown beyond a single location.

Feature	Included
Locations	Up to 3
Staff roster	Up to 60 employees across all locations
All Operator features	
Multi-location schedule view (all locations in one dashboard)	
Float staff support (assign a staff member to either location in a given week)	
Cross-location conflict detection (flags staff double-booked across locations)	
Manager accounts	Up to 3 additional logins
Labor cost reporting	Full (hours + estimated wage cost per week, per location)
Priority support (same-day response)	

Reasoning: $89/month is positioned below 7shifts' Growth plan (~$69.99/month per location for multi-location features, meaning $139.98/month for two locations) and represents a meaningful discount for Devon's specific use case. The multi-location view and float staff support are the two features that make the Manager tier genuinely differentiated from Operator — not a padding of minor features, but a direct response to Devon's stated pain. At $89 for up to three locations, this is cheaper per-location than any named competitor at equivalent functionality.

The Manager tier also introduces the labor cost reporting that Devon needs to defend scheduling decisions to his owner on Monday mornings. This is not a payroll integration — it is a simple hours × configured hourly rate estimate. Building it as a true payroll connector is a Phase 5 or later consideration; at MVP, a configurable rate field per employee is sufficient to produce the report Devon actually needs.

Revenue Model Recommendation

Monthly flat-rate subscription. Annual prepay available as a discount lever, not a default.

The core billing structure is monthly. Annual prepay at a 15–20% discount (equivalent to ~2 months free) should be available but not promoted aggressively until Month 9 or later. Rationale: in the first 12 months, the primary goal is retention signal, not cash collection. Operators who prepay annually and then churn at month 4 obscure the true retention picture and create refund complexity. Monthly billing keeps churn visible and forces honest engagement with Kill Condition K4.

Annual prepay becomes the right promotional instrument after month-12 retention data confirms that average customer life exceeds 10 months. At that point, offering annual plans to the retained cohort converts a retention probability into cash certainty and lowers effective CAC by eliminating the monthly re-acquisition cost inherent in month-to-month churn. (est.)

Per-run pricing is not recommended. Per-schedule-published pricing would penalize the exact behavior the product needs to reinforce — publishing a schedule every week — and creates an incentive for operators to publish less frequently, which reduces habit formation and accelerates churn. Subscription pricing aligns revenue with the usage pattern that generates retention.

Hybrid consideration (Phase 5+): A usage-based layer — for example, charging per SMS notification sent above a threshold — could be introduced at scale if SMS costs become material. At the volumes projected in the first 12 months, this is not a financial concern. Do not introduce billing complexity before retention is proven.

Unit Economics

Metric	Value	Assumption
Customer Acquisition Cost (CAC)	$250–$350 (est.)	Blended across three tested channels: cold outbound email ($300–$450 CAC at 2–3% conversion), F&B association/peer referral ($100–$200 CAC), and POS marketplace referral ($150–$250 CAC). No verified CAC benchmark for restaurant scheduling SaaS was available in the research data; this estimate is model-derived from comparable SMB SaaS acquisition patterns and should be validated through Phase 4 channel testing before any capital is committed to scaling. (directional)
Average Revenue Per User (ARPU)	$55/month (est.)	Assumes ~80% of paying customers on Operator ($49/month) and ~20% on Manager ($89/month), producing a weighted average of ~$57/month. Rounded to $55 to account for annual discount uptake in later months. Starter free tier generates $0.
Lifetime Value (LTV)	$770 (est.)	At base-case 7% monthly churn, average customer life = ~14.3 months. LTV = $55 × 14.3 = $787, rounded to $770 at 70% gross margin applied. At kill-condition churn of 10% (average life = 10 months), LTV drops to $385 at margin — below 2:1 LTV:CAC, which is structurally unviable without CAC reduction or price increase.

LTV:CAC Ratio	2.5:1–3.1:1 (est.)	Base case: $770 LTV / $300 blended CAC = 2.6:1. This is marginal — the industry benchmark for healthy SaaS unit economics is 3:1 or higher. The business becomes viable with either churn reduction below 5% (LTV rises to ~$1,100, ratio = 3.7:1) or CAC reduction through a distribution partner (POS referral at $150 CAC produces a 5.1:1 ratio). This ratio is the primary reason Kill Condition K5 exists.
Payback Period	7–9 months (est.)	At $55 ARPU and 70% gross margin, monthly gross contribution = $38.50/customer. At $300 blended CAC: payback = $300 / $38.50 = ~7.8 months. This clears the 12-month payback threshold in Kill Condition K5, but only at the base-case CAC. If cold outbound proves to be the only working channel at $400+ CAC, payback extends to 10+ months and approaches the kill threshold.
Gross Margin	70–75% (est.)	Reflects SaaS infrastructure costs (hosting, database, SMS notification delivery via a provider such as Twilio), customer support at founder/early employee scale, and no sales commission at direct acquisition volumes. SMS notification costs are the primary variable cost item and are estimated at $0.01–$0.02 per message; at 20 staff receiving 2 messages per publish event weekly, SMS adds approximately $0.80–$1.60/customer/month in direct cost at Operator tier, keeping gross margin above 70%. Gross margin will compress to 65–68% if SMS usage scales faster than assumed.

A note on all unit economics figures: These are planning estimates derived from comparable SMB SaaS businesses and the cost structure assumptions above. No verified unit economics data for restaurant scheduling SaaS specifically was available in the research data. Every figure in this table must be updated with real observed data after Phase 3 pilots produce actual acquisition cost and retention signal. Do not use these numbers to make capital allocation decisions before that data exists.

12-Month Revenue Model

All figures are illustrative planning estimates. (est.) Revenue projections assume base-case assumptions hold. Validate each assumption through customer discovery before using for financial planning.

Month	New Customers	Churned	Cumulative Customers	MRR	ARR Run-Rate
1	3	0	3	$165	$1,980
2	4	0	7	$385	$4,620
3	5	0	12	$660	$7,920
4	7	1	18	$990	$11,880
5	8	1	25	$1,375	$16,500
6	10	2	33	$1,815	$21,780
7	12	2	43	$2,365	$28,380
8	15	3	55	$3,025	$36,300
9	18	4	69	$3,795	$45,540
10	20	5	84	$4,620	$55,440
11	22	6	100	$5,500	$66,000
12	25	7	118	$6,490	$77,880

Key assumptions underlying this model: (est.)

- *Monthly churn rate:* 7% (base case). Applied to cumulative customer base; rounded to nearest whole number for months 1–3 where the base is too small for fractional churn to be meaningful.
- *New customer growth rate:* Starts at 3 in month 1 (Phase 3 pilots converting to paid), grows to 25 by month 12 as at least one acquisition channel reaches repeatability. This is not a compounding growth rate — it is a manually stepped progression reflecting the channel-building reality of months 1–6 versus the channel-scaling reality of months 7–12.
- *ARPU:* $55/month blended, held constant. Does not model tier upgrades or annual discount uptake, both of which would increase ARPU modestly in months 9–12.

- *Free-to-paid conversion:* Not modeled as a separate funnel in this table. Starter users who convert are counted as "new customers" in the month they upgrade. Free tier growth and conversion rates are inputs to Phase 4 channel testing, not assumptions that can be made with confidence before pilot data exists.
- *Month 1 customers:* Represents the 3–5 paid pilots expected from Phase 3 of the Validation Roadmap converting to monthly subscriptions, not cold acquisition.

The month 12 ARR of ~$78K falls within the lower end of the Year 1 SOM range described in the Product Brief ($290K–$356K ARR at 100–300 customers). (est.) This model is deliberately conservative — it reflects the realistic acquisition pace of a pre-funded, founder-led product in a market that requires trust-based channels, not a paid acquisition flywheel. If cold outbound to Devon-type floor managers proves out in months 4–6 and produces consistent results, the new customer curve in months 8–12 could increase 50–80% above the base case without changing the structural model.

Sensitivity Analysis

Scenario	Key Assumption Changed	Month-12 ARR
Pessimistic	Monthly churn rises to 10% (hits Kill Condition K4); new customer acquisition runs at 60% of base case (cold outbound underperforms, no partner channel established); ARPU stays at $49 (no Manager tier adoption)	~$26,460 (est.)
Base	7% monthly churn; new customer growth as modeled above; $55 blended ARPU	~$77,880 (est.)
Optimistic	Monthly churn drops to 4% (product earns genuine habit formation); new customer acquisition runs at 150% of base case (one partner channel — POS referral or association co-marketing — activates in month 5); ARPU increases to $65 as Manager tier adoption reaches 30%	~$156,000 (est.)

Break-even month:

- Pessimistic scenario: Break-even is not reached within 12 months at a founder cost base of $6,000–$10,000/month. The business requires either a price increase, a pivot to higher-ARPU customers, or external capital to survive past month 9. This scenario maps directly to Kill Condition K4 and K5 triggers.

- Base scenario: Break-even on direct product costs (infrastructure, SMS, support tools) is reached around month 6–7. Break-even against a founder living cost of $8,000/month (a reasonable zero-salary threshold for a solo founder) is reached around month 10–11. (est.)
- Optimistic scenario: Break-even against founder living costs reached month 7–8; the business generates enough cash flow by month 12 to fund a first hire or initiate a small pre-seed raise from a position of demonstrated traction. (est.)

Top 3 Expansion Revenue Paths

These are the three upsell and add-on paths most likely to increase ARPU without requiring a new product category. All are Phase 5+ considerations — none should be built before retention at the core product is proven.

1. Annual Prepay Conversion (Tier Upgrade Path)

Once average observed customer lifetime crosses 10 months, promote annual billing to the active paid cohort at a 15–20% discount. This converts a probabilistic retention assumption into a cash certainty. For a 100-customer base on monthly plans at $55 ARPU, converting 40% to annual at 17% discount produces approximately $22,000 in upfront cash and reduces monthly churn measurement complexity. The expansion revenue is not new ARPU — it is LTV acceleration.

2. SMS Overage / Notification Packs (Usage-Based Add-On)

The Operator tier bundles SMS notifications for up to 30 staff. Restaurants with high-turnover seasons (summer, holidays) or larger rosters that push against the 30-staff cap could be offered a notification pack extension at $9–$15/month rather than forcing an immediate tier upgrade. This creates a lower-friction upsell surface for operators who are not ready for Manager pricing but are generating real SMS cost. Do not build this until SMS costs become a meaningful margin line item — at the projected Year 1 volumes, this is premature.

3. POS / Payroll Data Integration (Premium Feature Add-On)

The single most-requested feature from Devon-type floor managers, based on the persona analysis, is labor cost reporting tied to actual wage rates from the POS or payroll system. A basic version (manual wage entry) is included in the Manager tier. A premium version — a live integration with Square, Toast, or Gusto that pulls actual wage data and produces a real-time labor cost percentage against revenue — could command a $20–$30/month add-on at the Operator tier or be positioned as a Manager+ plan at $119/month. This is the feature most likely to convert a floor manager's owner from "this is Devon's scheduling convenience tool" to "this pays for itself in overtime prevention every month" — which is the phrase that eliminates churn driven by owner budget scrutiny. The integration engineering cost is non-trivial; do not build until the core scheduling habit is proven and at least 20 paying customers have explicitly named labor cost visibility as a retention driver in check-in calls.

KEY TAKEAWAYS

- Flat-rate pricing at $49/month is the right structure, but it is a hypothesis, not a decision. The $49/month Operator price must clear Kill Condition K1 through direct willingness-to-pay conversations in Validation Roadmap Phase 2 — if fewer than 3 of 10 interviewees signal genuine commitment at that price point, the tier structure in this section should be repriced before any build begins. The competitor benchmark (7shifts Entree at ~$29.99/month) sets the floor; the key Phase 2 question is whether the product's simpler, mobile-first positioning justifies a $19/month premium or whether $29 is the real ceiling for this segment.
- Unit economics are viable in the base case but fragile under churn. At 7% monthly churn, the LTV:CAC ratio of ~2.6:1 clears the minimum threshold but leaves no margin for acquisition cost overruns. The business does not become structurally healthy until churn drops below 5% (ratio rises to ~3.7:1) or a low-CAC distribution channel — POS referral, association partnership — reduces blended CAC below $200. The single most important lever in months 7–12 is not customer growth rate; it is churn reduction through the white-glove onboarding strategy described in the Validation Roadmap, because every percentage point of monthly churn reduction increases LTV more than adding five new customers does.
- The three expansion revenue paths are real but sequenced. Annual prepay conversion, SMS overage packs, and POS/payroll integration are all credible ARPU expansion mechanisms — but building any of them before 50 retained paying customers have demonstrated the core scheduling habit is a distraction that will slow the retention work that must happen first. The month-12 ARR target of ~$78K in the base case is achievable from the core two-tier paid structure alone; expansion revenue is a year-two story, contingent on the year-one retention story proving out.

05 Customer Personas

User Bias & Personas

Why This Section Exists

Personas in a pre-revenue product are not descriptions of customers you have — they are structured hypotheses about who will pay, who will use, and who will churn. Every claim below is falsifiable. The Validation Roadmap's Phase 1 discovery interviews are the primary mechanism for stress-testing these hypotheses against real operators.

This section also documents the biases most likely to corrupt your research signal before you have enough data to know it's corrupted. That documentation matters more than the personas themselves.

Persona 1 — Maria, The Owner-Operator

Demographics

- Age: 38–52
- Role: Sole owner and day-to-day operator of one casual dining or family-style restaurant
- Staff size: 8–18 hourly and tipped employees
- Tech profile: Uses smartphone fluently for personal tasks (banking, Instagram, group texts), but approaches new business software with deep skepticism — has been burned by tools that required more time to maintain than they saved

Her Week Maria opens the restaurant four days a week and covers the floor on two of them. Scheduling happens Sunday night after service, at the kitchen table, in whatever time is left after invoices and payroll prep. The schedule lives in a shared Google Sheet she built herself three years ago. She texts changes directly to staff. When someone calls out, she works the shift herself or texts everyone in the group chat and hopes someone picks it up.

She has looked at 7shifts twice. Both times she got to the pricing page, decided it was built for someone with an HR department, and closed the tab.

Pain Points

- The Sunday scheduling ritual takes 90 minutes she doesn't have

- Staff regularly claim they "didn't see" schedule changes posted to the Google Sheet
- She has accidentally scheduled two people as openers on the same day and not caught it until 6am
- Overtime is tracked manually; she has paid unexpected overtime on at least three occasions in the past year and only noticed at payroll
- High staff turnover means she is constantly re-explaining the schedule format to new hires

Goals

- Build next week's schedule in under 20 minutes
- Stop being the person who has to chase people down when a shift isn't covered
- Know — before she hits send — whether anyone is crossing 40 hours that week
- Not spend money on a tool that requires training or a learning curve she doesn't have time for

Jobs to Be Done

Job Type	Statement
Functional	When I'm building the weekly schedule, help me see who's available and who's already at risk of overtime before I finalize anything
Functional	When a staff member can't make a shift, help me find a replacement without making twenty phone calls
Emotional	When I publish the schedule, let me feel confident that everyone actually received it and has no excuse not to show up
Social	When I talk to my food cost consultant or my accountant, let me feel like I run a professional operation, not a barely-contained chaos

Defining Characteristic Maria is the buyer and the primary user. Winning her means winning both the sales decision and the activation moment in the same conversation. She has no procurement process, no vendor evaluation committee, and no budget approval cycle — which makes conversion fast if the value is immediately visible and slow if she has to imagine the value in the abstract.

Persona 2 — Devon, The Hired Floor Manager

Demographics

- Age: 26–38
- Role: General manager or floor manager at a 2–3 location independent restaurant group; does not own the business
- Staff size: 15–30 across locations they are responsible for
- Tech profile: Comfortable with SaaS tools — uses Slack, Square, Google Workspace daily — but has no authority to approve software spend above $100/month without owner sign-off

His Week Devon manages two locations for an owner who is mostly absent from day-to-day operations. Scheduling across both locations means cross-referencing two separate spreadsheets, managing staff who sometimes float between locations, and fielding texts from staff on both teams. The owner checks the weekly labor cost report every Monday morning and asks Devon to explain any overtime. Devon currently uses a color-coded Excel template he built over two years. He knows exactly where its limitations are.

He has heard of 7shifts and has a free trial account he never activated. The owner told him to find something that doesn't cost more than $50 a month. Devon looked for 30 minutes, found nothing that felt worth the effort of migrating, and went back to Excel.

Pain Points

- Managing float staff who work both locations creates scheduling conflicts neither spreadsheet can flag automatically
- He is accountable for overtime spend but has no real-time visibility into hours-to-date mid-week
- Staff shift-swap requests come in via text, Instagram DM, and occasionally a sticky note on the manager's board — there is no single place they live
- When someone calls out at Location 2, he has to manually check Location 1's schedule to know who might be free, then text them individually
- Turnover among younger staff means he is constantly adding new people to the schedule and explaining shift codes to them

Goals

- One place — not two spreadsheets — to see the full schedule for both locations simultaneously
- Real-time visibility into hours-to-date so overtime surprises don't show up on Monday morning
- A channel for shift swap requests that doesn't require him to monitor his personal text thread
- Something he can show the owner as evidence that he's running a professional operation

Jobs to Be Done

Job Type	Statement
Functional	When I'm scheduling across two locations, help me see which staff are available at which location and flag anyone I've double-booked
Functional	When a staff member requests a swap, give me a way to approve or deny it in one step without it disappearing into my text inbox
Emotional	When the owner asks about overtime on Monday, let me show him a report instead of recounting a conversation from memory
Social	When I'm interviewing for a better management job, let me describe a scheduling system I built and maintained — not a spreadsheet I inherited

Defining Characteristic Devon is often the primary daily user but not the final buyer. The sales conversation with Devon fails if it doesn't account for the fact that he will have to sell this upward to an owner who is not in the room. He needs a price point he can defend in a 2-minute conversation and a value story that sounds like cost reduction (labor cost visibility, overtime prevention) rather than convenience, because owners respond to dollars, not minutes.

Persona 3 — Priya, The Informal Scheduler

Demographics

- Age: 22–30
- Role: Senior server, FOH lead, or shift lead at a single-location restaurant — scheduling has been informally delegated to her because she's reliable, but it is not her job title and she is not paid for it
- Staff size: 6–15 people she coordinates
- Tech profile: Highly mobile-native; runs everything from her phone; has zero tolerance for web-based tools that don't work well on mobile

Her Week Priya was handed the schedule by the owner eight months ago because the previous manager quit and someone had to do it. She builds it in a WhatsApp group and a notes app. She has no visibility into who's requested time off except the texts she's received. She has no way to know if someone is close to overtime. She does this in addition to her server shifts, and she is starting to resent it. When a conflict arises, staff blame her personally — not "the schedule" — because the process has no institutional structure. She is the system.

She does not think of herself as a buyer of software. But she is the person most likely to ask the owner to buy something, because she is the one experiencing the pain daily.

Pain Points

- Scheduling is invisible work — she gets no recognition for doing it well and immediate blame when it goes wrong
- Staff text her at all hours because she is the only point of contact for schedule questions
- She has no formal process for time-off requests, so they arrive in multiple channels and some get lost
- She is exhausted by the cognitive load of tracking six to ten people's availability in her head
- She resents that this work has become permanent without any adjustment to her pay or role

Goals

- A simple way to post the schedule where staff can see it without texting her
- A time-off request system that is not her personal inbox
- A way to hand this responsibility back to the owner or a new manager if she needs to
- If she is going to do this job, to do it in 20 minutes instead of 90

Jobs to Be Done

Job Type	Statement
Functional	When I publish the schedule, make it so staff can see it on their phones without asking me to screenshot and send it to them individually
Functional	When someone requests time off, give me a single place where all requests live so nothing gets lost in my texts
Emotional	When I'm doing this work for no extra pay, let me at least feel like I'm doing it competently and efficiently
Social	When I explain the scheduling process to new hires, let me point to a system instead of giving them my phone number

Defining Characteristic Priya is a champion, not a buyer. She is not the decision-maker on spend, and she may not even know the owner is open to paying for a tool. Her value to the go-to-market strategy is as a referral source: she is the person most likely to tell her owner "I found something that would fix this," especially if she encounters the product organically (social media, peer recommendation) rather than through a cold outbound sequence. Designing for Priya means designing for mobile-first use and zero-configuration onboarding — because she will not set up an integration or configure a settings page.

Key Biases to Watch for in User Research

These are the distortions most likely to corrupt signal during Phases 1 and 2 of the Validation Roadmap. They are listed in order of likelihood and severity of impact on decision-making.

Researcher-Side Biases (What You Might Hear That Isn't True)

1. Courtesy Bias Restaurant operators — especially owner-operators like Maria — are hospitality professionals. Their instinct is to make the person across from them feel heard and valued. When asked "would you pay $49/month for this?" many will say yes, or indicate they would consider it, simply because declining feels rude in a face-to-face setting. This is the single most dangerous bias in Phase 2 of the Validation Roadmap.

Mitigation: Require behavioral evidence, not verbal commitments. A yes is an LOI, a pilot agreement, or a statement of specific current spend being displaced. "I'd consider it" is a no until proven otherwise. Follow up warm interviews with cold outreach to operators you have no relationship with — courtesy bias collapses in email.

2. Recency Bias in Pain Reporting An operator who had a no-show two days ago will dramatically overstate scheduling pain relative to an operator who had a smooth week. If you interview someone during a high-volatility period (summer weekend staffing, holiday schedules, post-layoff crunch), their reported pain level will not reflect their average week.

Mitigation: Ask operators to describe their last three scheduling cycles, not just the most memorable one. Listen for frequency of problems, not intensity. A problem that happens every six weeks is not the same as one that happens weekly, even if the operator describes both with equal frustration.

3. Confirmation Bias (Founder Side) When an operator says "yeah, we really struggle with scheduling," it is easy to hear validation of the product rather than validation of the problem. These are not the same thing. Struggling with scheduling could mean they want your tool, or it could mean they have accepted the struggle as the cost of running a restaurant and would never pay to solve it.

Mitigation: After every interview, write down exactly what the operator said, verbatim, before interpreting it. Flag any inference you make that is not directly supported by their words. Review the raw transcript against the success criteria in Phase 1 before counting it as a pass.

4. Selection Bias in Recruiting Operators who agree to a discovery interview about scheduling software are disproportionately likely to already be tool-aware, frustrated with their current method, and open to change. The population that will actually generate revenue includes a large segment of operators who would never agree to a 45-minute interview — they're too busy and too skeptical of anyone pitching them.

Mitigation: Do not recruit exclusively from restaurant Facebook groups or association member lists. These skew toward operators who are already seeking solutions. Include cold outreach to operators who have no reason to expect a conversation — their response rates and willingness-to-pay signals are more predictive of the actual market.

Product-Side Biases (What Behaviors Might Mislead You)

5. Novelty Effect in Pilot Retention As noted in the Validation Roadmap, operators often engage consistently in pilot weeks 1 and 2 because any new tool creates an initial engagement spike. This is not retention — it is curiosity. The diagnostic moment is week 3, when the first real operational pressure hits (a last-minute callout, a holiday weekend, a full house with two no-shows) and the operator must decide in real time whether to use the product or revert to the group text.

Watch for: Login activity in week 3 relative to week 1. Any drop of more than 40% in session frequency between week 2 and week 3 is a reversion signal, even if the operator tells you in their week 4 check-in that they're still using it. Self-reported usage is not a substitute for login data.

6. The "It's Fine" Attribution Error Operators who do not churn during the pilot may be staying because the product is genuinely valuable or because switching back to their old method requires re-explaining the change to their staff, which is its own friction. Retention driven by switching cost inertia looks identical to retention driven by value — until you try to raise the price or expand to a second location, at which point the distinction becomes catastrophic.

Mitigation: In every check-in call, ask directly: "If this tool disappeared tomorrow and you had to go back to your old method, what would you lose?" An operator who can name two or three specific things they would lose is retained by value. An operator who pauses and says "I don't know, I guess I'd just go back to the spreadsheet" is retained by inertia.

7. The Workaround Undercount Operators who have built sophisticated personal workarounds — a well-maintained Google Sheet with conditional formatting, a well-trained staff WhatsApp group, a laminated weekly template — systematically underreport their scheduling pain because their workaround has reduced the frequency of visible failures. The pain is still there; it has just been suppressed by effort they no longer consciously notice they're expending.

Mitigation: In Phase 1 interviews, ask specifically how long the current scheduling process takes, who maintains it, and what happens when the person who maintains it is unavailable. A system that depends on one person's undocumented knowledge is fragile in ways the operator may not have articulated as a problem.

User Journey: Discovery ’Signup ’First Value ’Retention

Each stage is analyzed by persona, because the same product feature serves different psychological needs at each stage depending on who is using it.

Stage 1 — Discovery

Maria (Owner-Operator) Most likely to discover through peer recommendation at a local restaurant association meeting, a post in a regional restaurateurs Facebook group, or a conversation with her POS sales rep. She will not find this product through Google search — she is not searching for "scheduling software for restaurants." She does not have the vocabulary yet. Discovery must reach her in a context she already trusts.

Devon (Floor Manager) More likely to discover through direct outreach — a well-written cold email that names his specific pain (multi-location visibility, overtime tracking) will get a reply. He is also reachable through the Square App Marketplace or through a referral from a peer manager at another restaurant group. He is more tool-aware than Maria and more likely to be actively looking.

Priya (Informal Scheduler) Most likely to discover through Instagram or TikTok — short-form content showing "how I schedule 12 staff in 15 minutes" resonates with her because she lives on her phone and the pain is personal. She may also discover through a peer recommendation from another server at a different restaurant who saw her complaining about the scheduling load.

Strategic implication: Peer referral and local community channels are the common thread across all three personas. Cold outbound optimized for Devon is the highest-ROI channel for early acquisition because he is reachable via email and has clearer, articulable pain. Maria and Priya require trust-based channels that are slower to build but produce more durable customer relationships.

Stage 2 — Signup

Maria Will not complete a signup that requires entering credit card details before she has seen the product work. She needs to build at least one schedule — with her actual staff names — before she is willing to pay. Onboarding must not assume any prior knowledge of scheduling software. If the first screen asks her to "configure your scheduling preferences," she will close the tab.

Critical moment: She needs to see a complete, shareable schedule in under 10 minutes on her first session, or the product fails her before she gives it a fair evaluation.

Devon Will complete a more involved signup if there is a clear structure to it. He is willing to invest 20–30 minutes in onboarding if he can see the multi-location view he needs. But he will stop if the pricing page shows a per-location charge that requires going back to his owner for approval — the first plan he encounters must stay within the $49–$99/month range his owner authorized without renegotiation.

Critical moment: He needs to see both locations and their respective staff in a single view before the end of session one.

Priya Will only complete signup on her phone. If the mobile experience requires scrolling through a web interface designed for desktop, she abandons. She needs to build and share a schedule entirely from the mobile interface within the first session. She is the most likely persona to send the schedule link to a staff member during onboarding — if that staff member's experience of receiving the schedule is seamless, she will complete signup.

Critical moment: Staff notification sent from within the product, confirmed as delivered, during the first session.

Stage 3 — First Value

Across all three personas, first value is the same functional outcome, arrived at via different emotional paths:

> *The first schedule is built, published, and confirmed received by staff — without a single text message from the operator.*

For Maria, this moment feels like relief: Sunday night is 45 minutes shorter and she knows everyone got the schedule.

For Devon, this moment feels like control: he can see both locations in one place and has a defensible record if the owner asks about overtime.

For Priya, this moment feels like legitimacy: she is no longer the informal person managing a chaotic group text — she has a system she can point to.

The product must engineer this moment to happen within the first session, not after a week of use. If first value requires two or three scheduling cycles to materialize, churn before that moment is structurally unavoidable.

Stage 4 — Retention

What keeps them:

Persona	Retention Driver	Retention Risk
Maria	Schedule build time genuinely drops below 20 minutes and stays there; no missed shifts attributable to communication failure	Reverts to group text under first high-pressure week; decides the product is "one more thing to manage"

Devon	Multi-location view eliminates the spreadsheet juggling; overtime visibility prevents Monday morning owner conversations	Owner decides $49/month isn't justified and asks Devon to go back to the spreadsheet; product doesn't support the complexity of float staff
Priya	Staff stop texting her directly because they can see the schedule themselves; time-off requests have a clear channel	The tool becomes something she has to maintain in addition to, not instead of, her current method

Churn signals by persona:

- *Maria:* Stops logging in on Sunday evenings (reverted to spreadsheet); contacts support asking how to export staff list (planning to leave)
- *Devon:* Stops updating both locations simultaneously (reverted to single-location use, complexity not solved); asks about cancellation policy in a support message
- *Priya:* Staff stop clicking schedule notification links (the habit of checking the tool did not transfer to staff); she stops updating the schedule after week 3

Persona Priority Ranking

Rank	Persona	Reasoning
1 — Primary	Maria (Owner-Operator)	She is the buyer and the primary user in the same person. Winning Maria means no two-stage sales process, no internal approval cycle, and no misalignment between who pays and who uses. Her pain is the one most directly tied to the $49/month willingness-to-pay assumption in Kill Condition K1. The product must work for her before it works for anyone else.

2 — Secondary	Devon (Floor Manager)	Devon is the highest-volume near-term acquisition target because he is reachable via direct outreach, has a clearer articulated need, and operates at the scale (2–3 locations, 15–30 staff) where scheduling complexity is high enough to justify the spend without prompting. The risk is the split between user and buyer — but that split is manageable if the price point stays within what Devon can approve informally.
3 — Tertiary	Priya (Informal Scheduler)	Priya represents a real and underserved pain, but she is structurally a champion rather than a buyer. Building for her mobile-first, zero-configuration use case is valuable because it also improves the experience for Maria and Devon — but she should not be the lead persona for product decisions or sales motions until the primary customer base is established. She is a referral vector and a product quality signal, not a revenue driver in Phase 3 or 4.

A note on ranking stability: This ranking reflects the current hypothesis, not a permanent architecture. If Phase 1 discovery reveals that floor managers like Devon are the primary pain-holders in 9 of 15 interviews while owner-operators like Maria describe scheduling as tolerable, the priority ranking should be updated before Phase 2 begins. The Validation Roadmap is designed to surface exactly this kind of signal — do not override it with persona assumptions made before the interviews happen.

06 Adversarial Review

Adversarial Review

Why This Might Not Work

1. The Pain Is Real But the Switching Cost Is Zero — in Both Directions

The product's core thesis is that independent restaurant operators are trapped in a painful manual process (spreadsheets, group texts) and will pay $49/month to escape it. But the same feature that makes the current behavior painful — its zero cost and zero setup — also makes it the default recovery path the moment anything goes wrong. Maria has been running her restaurant on a Google Sheet for three years. She knows exactly how it works under pressure. When the app has a bug on Sunday evening, or staff can't figure out the SMS link, or the onboarding takes 35 minutes instead of 20, the cost of switching back to the spreadsheet is precisely zero. There is no sunk cost, no migration lock-in, no data hostage. This product has to be reliably better every single week — not just at signup — or it loses to a tool that costs nothing and has never let anyone down.

The unit economics make this worse. At $49/month and an estimated LTV of $770 at 7% monthly churn (est.), the product cannot survive a churn rate that reflects even occasional operational failures. Churn at 10% — the Kill Condition K4 threshold — compresses LTV to $385 (est.), making the estimated $250–$350 CAC (est.) barely viable at a 1.1–1.5:1 LTV:CAC ratio. That ratio does not support a business; it supports a lifestyle product that breaks even if nothing goes wrong. A single bad Sunday during pilot onboarding — a missed SMS batch, a timeout, a confusing staff view — is not just a support ticket. It is a churn event that costs $770 in foregone revenue and restarts a $300 CAC cycle.

2. The Price Ceiling May Already Be Below Viable Unit Economics

The product targets independent operators with 3–9% net margins on $400K–$1.5M in revenue. (directional) At the lower end of that revenue range, a $400K restaurant generating 5% net margin earns $20,000/year in profit. A $49/month subscription represents 2.9% of annual net profit. That is not obviously unaffordable, but it exists in a purchasing environment where every recurring expense competes with food costs, labor, rent, and POS fees. The Monetization section acknowledges this framing and calls it a risk of "medium likelihood, high impact."

The structural problem is that $49/month may be both too high for the buyer and too low for the business. The estimated blended ARPU of $55/month (est.) with a $250–$350 CAC (est.) and 7% monthly churn produces a 2.6:1 LTV:CAC ratio (est.) that the section itself describes as "viable but fragile." Fragile unit economics at the projected case — before accounting for the likelihood that cold outbound CAC is higher than estimated, that free-to-paid conversion is lower than assumed, and that independent restaurant operators are more

price-sensitive under stress than they report in interviews — is not a margin of safety. It is a single bad assumption away from a business that cannot sustainably acquire customers.

3. The 12–18 Month Timing Window Assumes Incumbents Stay Still

The Product Brief identifies a "market timing advantage: 12–18 months before Toast/Square/7shifts push downmarket." This framing inverts the actual risk. Toast already has a scheduling feature in its POS ecosystem. Square has native team management. 7shifts has a free tier that covers up to 10 employees per location. The competitive moat being claimed — simplicity, flat-rate pricing, mobile-first design — is not a patent or a regulatory barrier. It is a UX preference and a pricing model choice, both of which can be replicated by a well-resourced competitor in a single product cycle.

The relevant question is not whether this product can be better than 7shifts for Maria today. It is whether 7shifts (which already has 7shifts Free covering the smallest operators) or Toast (which has Maria's payment processing relationship and daily active usage) decides that the independent operator segment is worth a focused product investment. If either does, they enter with an existing customer relationship, an existing support infrastructure, an existing brand, and a distribution channel that this product cannot match. The 12–18 month window is not a competitive moat — it is a countdown clock, and it may already be running.

4. Staff Turnover Undermines the Product's Own Core Workflow

The Competitive Landscape section notes that per-user pricing models (like When I Work at $2.50–$6/user/month) are "problematic in high-turnover environments" — which is precisely why this product uses flat-rate pricing. But high staff turnover does not just affect pricing models. It directly degrades the product's core value delivery. A scheduling tool's value depends on accurate, maintained staff data: phone numbers, availability preferences, position qualifications, overtime thresholds. At the independent restaurant level, annual staff turnover routinely exceeds 70–100%. (directional)

This means Maria is constantly onboarding new staff members into the system, collecting phone numbers from people who may distrust receiving texts from unknown business software, managing staff who have opted out of SMS, and dealing with the gap between who is in the system and who is actually working this week. Every new hire is a partial re-onboarding. Every staff departure is a potential opt-out or data deletion request. The administrative burden of maintaining clean staff data may, for some operators, exceed the burden the product is replacing — particularly in the first 60–90 days when the habit is not yet established. This is the most likely mechanism for the pattern noted as "High likelihood, High impact" in the Product Brief: habit reversion to group texts under operational stress.

5. Independent Restaurant Operators Are a Structurally Difficult Sales Motion

The GTM strategy depends on owner-operators who make unilateral purchasing decisions — Maria works the floor, manages her own schedule, and will subscribe without an approval process. This is real. But it coexists with a structural truth about independent restaurant operators as a customer segment: they are

geographically dispersed, have no professional association with purchasing authority, have very low tolerance for software that requires ongoing attention, and have historically been underserved by SaaS precisely because the economics of selling to them are difficult. The CAC estimate of $250–$350 (est.) via cold outbound and local community channels is plausible in theory. In practice, cold outbound to restaurant owners requires finding contact information, making contact when the owner is not managing a lunch rush, earning enough trust to schedule a follow-up, and converting a conversation into a trial. The conversion funnel for a non-technical buyer in a high-stress industry is not the same as the conversion funnel for a software buyer at a desk job. The actual CAC may be significantly higher than modeled — and the business does not work if it is.

Reference Class

No documented failure cases for restaurant staff scheduling SaaS targeting independent operators were found in the verified research data provided. The failure cases block is explicitly empty.

This is genuinely ambiguous and should not be read as validation:

Interpretation 1: The niche of independent-restaurant-first scheduling SaaS may be novel enough that prior attempts are not publicly documented. This is plausible — most failed B2B SaaS products targeting small, fragmented service businesses shut down without press coverage, post-mortems, or acquisition announcements.

Interpretation 2: Prior attempts exist but failed without public documentation. This is common in horizontal SMB SaaS targeting restaurants, retail, and service businesses, where the failure mode is typically not dramatic collapse but quiet revenue stagnation — the product reaches 50–100 customers and cannot grow further because CAC exceeds sustainable payback at the price point the market will bear. These businesses often persist for years below viability rather than shutting down cleanly.

What can be inferred from the competitive landscape data that is verified: the fact that 7shifts, Homebase, When I Work, Sling, and Deputy all exist and have achieved meaningful scale while not focusing exclusively on independent restaurants suggests either that (a) the independent restaurant segment genuinely cannot support a standalone scheduling SaaS at venture scale, and incumbents rationally chose to pursue larger customers, or (b) the segment is underserved and represents a real opportunity. Both interpretations are consistent with the observed market structure. Determine which applies by asking independent restaurant operators directly: "Has anyone pitched you scheduling software before? What happened?"

Steelman: The Case for Status Quo

The strongest argument for doing nothing is that Maria's current system already works — not elegantly, not professionally, but reliably enough that her restaurant is still open. She has been building schedules on a Google Sheet for three years. Her staff know to check their texts on Sunday night. When someone can't make a shift, they text the group and someone usually covers. The system has failure modes, but Maria has learned to manage them. She knows which staff members are reliable responders, which ones need a direct call, and which shifts are most likely to have a callout. That operational knowledge is not in any software product — it is in her head, and she applies it every week without a subscription fee.

At $49/month, the honest cost comparison is not "$49/month versus free." It is "$588/year versus a tool she already knows how to use under pressure, with no onboarding time, no staff training, no new phone numbers to collect, and no dependency on a startup that may not exist in 18 months." The time savings argument — 45 minutes per week recovered — is real if the product works flawlessly. But flawless is not the comparison class. The comparison class is: what happens the first time the SMS batch fails at 8pm on a Sunday, or a staff member can't open the link, or the overtime alert fires incorrectly? In those moments, Maria does not call support. She opens her Google Sheet and finishes the schedule. And the following week, she opens her Google Sheet first, because it didn't fail her last week.

The incumbent's perspective is rational: the spreadsheet has a 100% uptime record in Maria's experience. It requires no new behavior from staff. It costs nothing. Its failure modes are familiar and recoverable. The burden of proof is entirely on the new product to demonstrate, over multiple consecutive weeks without failure, that it is worth $49/month and the cognitive overhead of switching. That burden is high, the tolerance for failure is low, and the reversion path is always one tap away.

07 Kill Conditions

#	Condition	By When	Status
K1	Fewer than 3 of 10 discovery interviews with independent restaurant owners or managers express a clear willingness to pay at least $49/month specifically for scheduling software (not generic "interest")	Month 1	Pending
K2	Zero paid pilots (even at a steep discount) signed from a pool of at least 20 directly contacted restaurants	Month 3	Pending
K3	Fewer than 5 active paying restaurants retained after 60 days of use — indicating the product is not sticky enough to displace existing habits (spreadsheets, pen-and-paper, or incumbents like 7shifts or HotSchedules)	Month 6	Pending
K4	Average monthly churn across the paying base exceeds 10% for two consecutive months, suggesting the core scheduling workflow does not fit real restaurant operating patterns (e.g., shift volatility, last-minute callouts, tipped vs. hourly role complexity)	Month 9	Pending

K5	Customer Acquisition Cost cannot be brought below 12 months of subscription revenue per customer through any tested channel (outbound, local F&B associations, POS integrations) by the end of Q3, making unit economics structurally unviable at SMB price points (directional)	Month 9	Pending

These are the specific, observable outcomes that would indicate the core hypothesis — that independent restaurants will pay for and consistently use a dedicated scheduling tool — is wrong. Review these monthly against logged data, not memory.

08 Validation Roadmap

Validation Roadmap

Before a single line of code is written or a contract signed, these four phases answer the only question that matters at this stage: do independent restaurant operators have a real, recurring scheduling pain they will actually pay to solve — and is this team the right one to solve it?

Each phase is a gate. Failing a gate cleanly is as valuable as passing it.

Phase 1 — Customer Discovery: Is the Pain Real and Specific?

Hypothesis Independent restaurant owners and managers experience scheduling friction severe enough that they would describe it unprompted as a recurring operational cost — in time lost, shifts miscovered, or staff turnover — not merely a minor inconvenience they've already accepted.

Experiment Conduct 15 in-person or video discovery interviews with owners or managers of independent restaurants (1–3 locations, 5–30 staff). Recruit through local restaurant associations, Facebook groups for restaurant operators, Yelp/Google reviews of non-chain restaurants in your metro, and warm introductions. Do not recruit from SaaS review sites — those operators are already tool-aware and will skew responses.

The interview script must avoid leading questions. Ask only:

- "Walk me through how you build your schedule each week — start to finish."
- "What breaks down most often in that process?"
- "What did you do the last time someone called out 2 hours before a shift?"
- "What would need to be true for you to pay for something that fixed this?"

Record and tag responses by pain category: time cost, coverage failure, compliance/overtime exposure, staff communication. Do not pitch anything. Do not mention a price.

Cost Estimate $0–$200 (travel, coffee meetings, optional $10–15 Amazon gift card per participant to improve show rates). Primary cost is 30–40 hours of founder time.

Time Estimate 3 weeks

Success Criteria

- At least 10 of 15 interviewees describe a specific, recurring scheduling failure (not a vague inconvenience) without being prompted

- At least 8 of 15 can name an actual downstream consequence: a no-show that cost them labor, an overtime violation, a staff complaint
- At least 6 of 15 are currently using a workaround (spreadsheet, group text, whiteboard) that they describe with visible frustration

Failure Criteria Fewer than 6 of 15 can describe a concrete scheduling failure with a real downstream consequence. This signals that scheduling pain in this segment is diffuse and already tolerated — meaning the addressable market is far smaller than assumed. This does not directly trigger a Kill Condition yet, but it makes K1 unreachable in Phase 2 and should prompt a full pivot review before proceeding.

Don't Build Yet Scheduling UI, shift-swap logic, staff notification system, any mobile interface, integrations with POS systems (Toast, Square), anything in a database. No landing page with a waitlist form — it creates false signal.

Phase 2 — Willingness to Pay: Will They Actually Open Their Wallet?

Hypothesis At least 3 of 10 independent restaurant operators who confirmed real scheduling pain in Phase 1 will express clear, unprompted willingness to pay at least $49/month for a dedicated scheduling tool — not generic interest, but a statement of intent tied to a specific budget or current spend.

Experiment Return to the 10–12 interviewees from Phase 1 who showed the strongest pain signals. Present a concrete, specific value proposition (not a demo — a one-page visual mockup or even a printed PDF showing what the workflow would look like). Be explicit: "This would cost $49–$99/month. Would you pay that?" If they hesitate, probe: "What are you currently paying for scheduling tools, or how much is your current process costing you in manager time per week?"

Also run a parallel outbound sequence: directly contact 20 additional restaurant managers you have not spoken to yet via cold email or LinkedIn. Use a two-sentence pitch that names the specific pain and the price. Track reply rate and the quality of replies (question-asking vs. polite deflection).

Do not accept "I'd consider it" as a yes. A yes is: "Yes, I'd pay that," "I already pay X for Y and this is better," or agreement to sign a Letter of Intent for a paid pilot at any price.

Cost Estimate $100–$400 (printed materials, outreach tools like Apollo.io or Hunter.io for email finding, travel for in-person follow-ups). 20–30 hours of founder time.

Time Estimate 3 weeks

Success Criteria

- At least 3 of 10 Phase 1 returnees state clear willingness to pay e$49/month, unprompted by price anchoring

- At least 2 of 20 cold-outreach contacts reply with substantive engagement (asking about features, timing, or pricing — not auto-replies)
- At least 1 LOI or informal commitment to join a paid pilot at any price secured by end of phase

Failure Criteria Fewer than 3 of 10 Phase 1 contacts express willingness to pay at the stated price. This directly triggers Kill Condition K1 — and is a harder signal than it looks, because these are warm contacts who already confirmed pain. Unwillingness to pay from pre-validated pain holders means either the price is structurally wrong for this segment or the differentiation from free/current tools is insufficient. Either way, the unit economics behind K5 become impossible to solve.

Don't Build Yet Any functional prototype, MVP, or clickable demo. A static mockup (Figma or even PowerPoint) is sufficient and faster to change when feedback requires it. Do not build an onboarding flow, a billing integration, or a staff-facing mobile app. Do not register a domain with a live product page.

Phase 3 — Pilot Proof: Will They Use It Consistently Under Real Conditions?

Hypothesis Restaurant operators who commit to a paid pilot will actively use the scheduling tool across at least 4 consecutive weekly scheduling cycles without reverting to their prior method — proving that the product displaces an entrenched habit (spreadsheet, group text, or pen-and-paper) in the high-pressure, interruption-heavy environment of a working restaurant.

Experiment Sign 5–8 paid pilot agreements. Price at $29–$49/month (discounted from eventual target, but never free — payment is a commitment mechanism and free pilots produce misleading retention data). Build only the minimum workflow required for one scheduling cycle: create a weekly schedule, assign shifts to named staff, and notify staff via SMS or email. Nothing else.

Manually handle everything else. If a staff member wants to swap a shift, let them text the manager and you manually update the schedule in a spreadsheet you share with the pilot customer. You are not building a product yet — you are building evidence.

Check in with each pilot customer at the end of week 1, week 2, and week 4. Ask specifically: "Did you use this to build last week's schedule, or did you fall back to your old method? Why?" Log every instance of reversion and the reason.

Track one proxy metric obsessively: did the manager open and modify the schedule at least once in the 7 days before each shift week? That is the minimum signal of active use.

Cost Estimate $500–$2,000 (basic Glide, Notion, or Airtable build to simulate a scheduling interface; Twilio or similar for SMS notifications at ~$0.01/message; customer check-in time). 60–80 hours of founder time. Pilot revenue of $145–$392/month partially offsets cost.

Time Estimate 6 weeks (4 weeks of active pilot use plus 2 weeks of setup and onboarding)

Success Criteria

- At least 5 of 8 pilot restaurants actively use the tool to build their schedule in at least 3 of 4 pilot weeks (not reverting to prior method)
- Zero pilot customers request a refund or cancel within the 6-week window
- At least 3 of 5 active users can name one specific operational improvement (e.g., "I stopped getting texts about shifts," "I caught an overtime issue before it happened")

Failure Criteria Fewer than 5 active paying restaurants are retained after 60 days of use. This directly triggers Kill Condition K3. The specific failure mode to watch for: operators who use the tool for week 1 and week 2 (novelty effect) but revert by week 3 when the first real-world pressure hits — a Friday rush, a catering event, a sudden staff shortage. Reversion under pressure is more diagnostic than non-adoption from the start.

Don't Build Yet A production-grade multi-tenant SaaS infrastructure, a native mobile app, POS integrations (Toast, Square, Lightspeed), payroll export features, compliance/labor law alerting, or any enterprise tier feature. Do not implement SOC 2 controls or GDPR consent flows yet — these are required before any scale [verified research data], but premature at 8 pilot customers. Do not hire engineers.

Phase 4 — Retention and Channel Proof: Is the Business Structurally Viable?

Hypothesis Monthly churn can be held below 10% across the paying base for two consecutive months, and at least one repeatable, cost-efficient acquisition channel exists — meaning Customer Acquisition Cost can be projected toward payback within 12 months of subscription revenue, making the unit economics viable at SMB price points.

Experiment Expand from pilot cohort to 15–25 paying restaurants. Add one acquisition channel test per month — try at least three distinct channels in sequence, not simultaneously, so signal is clean:

- *Direct outbound* to restaurant owners via local F&B association member lists or cold email (measure reply-to-close rate and time-to-close)
- *POS partner referral* — approach one small regional POS reseller or the Square App Marketplace and ask if they will surface your tool to their restaurant clients (measure referral volume and quality)
- *Local restaurant association co-marketing* — sponsor a newsletter or speak at a meeting in exchange for a warm intro to members (measure cost-per-intro and conversion rate)

Simultaneously, track monthly churn weekly. If churn exceeds 10% in any single month, do a same-week exit interview — not a survey, a phone call — to identify whether it is a product fit failure, a price sensitivity issue, or a seasonal business pattern (many independent restaurants close or reduce hours in January or August, which creates artificial churn that is not a product signal).

Cost Estimate $1,000–$3,500 (outreach tooling, association fees, travel to local F&B events, one part-time contractor for customer success calls if founder bandwidth is exhausted). Revenue at 20 customers × $49/month = $980/month partially offsets. (est.)

Time Estimate 8 weeks

Success Criteria

- Monthly churn stays below 10% for two consecutive months across the paying base
- At least one acquisition channel produces a CAC with a projected payback period under 12 months of subscription revenue at the $49–$99/month price point (directional)
- At least 15 paying restaurants active at end of phase

Failure Criteria Monthly churn exceeds 10% for two consecutive months — triggering Kill Condition K4 — or no tested acquisition channel produces a CAC projecting to payback within 12 months by end of Q3 — triggering Kill Condition K5. Either outcome means the business cannot scale at SMB price points without a structural change: moving upmarket to multi-location operators, raising prices significantly, or finding a distribution partner (e.g., a POS company) who subsidizes CAC through a bundling arrangement.

Don't Build Yet Enterprise sales motion, annual contract infrastructure, or any feature requiring SOC 2 Type II certification (required before mid-market deals but premature here). Do not raise a seed round on pilot data alone — Phase 4 retention and channel data is the minimum credible evidence base for a fundraising narrative.

Phase Sequencing Logic

These phases are ordered to spend the least possible before reaching each kill condition. Phase 1 costs $200 and 3 weeks. If it fails, you have lost $200 and 3 weeks — not 6 months of engineering salary. Phase 3 is the first phase where real money (even at pilot pricing) and real build time are committed, and only after two phases of validated human signal. The B2B SaaS market is projected to grow significantly through 2034 [source: mordorintelligence.com], but market size is not a substitute for evidence that this customer segment, at this price point, with this product, generates durable retention. These phases are designed to answer that question before the market tailwinds become irrelevant to whether you can survive the first year.

09 GTM Strategy

SEO & GTM Strategy

Ideal Customer Profile

Primary ICP — The Accountable Owner-Operator

Who she is: An independent restaurant owner who manages her own schedule, works the floor regularly, and treats every dollar of operating expense as a personal decision. She runs one location with 10–25 hourly and tipped staff. She has been scheduling manually — spreadsheets, group texts, a whiteboard — for at least two years. She is not actively searching for scheduling software because she does not yet believe software can solve what she experiences as a people problem.

Firmographic profile:

- Business type: Independent casual dining, family-style, fast-casual, or neighborhood bar with food
- Location count: 1 (occasionally 2, with the second having opened in the last 18 months)
- Staff size: 10–25 employees
- Annual revenue: $400K–$1.5M (directional)
- POS system: Square or Toast (most common at this size)
- Tech stack otherwise: Google Workspace or nothing formal
- Geography: Mid-size U.S. cities, suburban markets, and neighborhoods underserved by enterprise software sales motions

Behavioral signals that indicate readiness: She has mentioned scheduling frustration in a local restaurant Facebook group in the last 90 days. She uses Google Sheets with conditional formatting she built herself, which signals she is capable of using a tool — she just has not found one worth switching to. She has visited the 7shifts or Homebase pricing page and left without signing up. She employs at least two staff members under 25, which means she has already experienced the pattern of schedule changes that "didn't go through" because her Google Sheet link was ignored.

What she will pay for: She will pay $49/month if she can see — within a single session — that the product builds a complete schedule faster than her spreadsheet and confirms staff received it without requiring her to chase anyone. She will not pay for a dashboard, a report, or a feature list. She will pay for a Sunday night that ends 45 minutes earlier than it does now.

Secondary ICP — The Multi-Location Floor Manager

Who he is: A hired general manager or floor manager responsible for day-to-day operations at a 2–3 location independent restaurant group. He is accountable to an owner who monitors labor costs but is not present daily. He has inherited a scheduling system he did not design and is maintaining it out of professional obligation rather than conviction.

Firmographic profile:

- Business type: Same as primary ICP, but at a small group level (2–3 locations, same ownership)
- Staff size: 15–35 employees across his locations
- Decision authority: Can approve spend up to ~$100/month informally; above that requires owner sign-off
- Tech profile: More tool-aware than the primary ICP; has existing SaaS in his stack (Square, Slack, Google Workspace)
- Geography: Same as primary ICP; slightly overrepresented in denser urban markets where multi-location independents are more common

Behavioral signals that indicate readiness: He manages float staff across locations in a spreadsheet with manual cross-referencing. He has been asked by his owner to explain an overtime charge in the past 60 days. He has a 7shifts or When I Work free trial he never activated. He uses his personal phone number as the primary channel for shift swap requests.

What he needs to hear: The value story must frame this as cost reduction — specifically, overtime prevention and labor cost visibility — rather than convenience or time savings. He cannot sell "this saves me 90 minutes a week" to his owner. He can sell "this flagged two overtime situations last week that would have cost us $180 in unplanned wages." That is the sentence that gets the owner to approve the spend without a meeting.

Positioning Statement

For independent restaurant operators who are still scheduling in spreadsheets and group texts, [Product Name] is a scheduling tool built for how restaurants actually run — where staff change weekly, shifts get covered at midnight, and the manager is also the one working the floor.

Unlike 7shifts and HotSchedules, which were designed for multi-unit chains with HR departments, [Product Name] is set up in one session, runs entirely from a phone, and gets your entire team on the same schedule before Sunday night is over.

The core promise: build next week's schedule in under 20 minutes, and know — before you hit send — whether anyone is crossing 40 hours.

Positioning Principles

- *Lead with the time problem, not the software category.* "Restaurant scheduling software" is a crowded category where incumbents have more brand equity. "Build next week's schedule in 20 minutes" is a concrete outcome that does not require the operator to already believe software is the answer.
- *Avoid enterprise language in all customer-facing copy.* Words like "workforce management," "labor optimization," "scheduling platform," and "team communication suite" are signals to Maria that this product was built for a company larger than hers. Use the words operators actually use: schedule, shift, callout, coverage, no-show, group text.
- *Position against the spreadsheet, not against 7shifts.* Most of the target market is not choosing between this product and 7shifts — they are choosing between this product and their current method (Google Sheet + group text). Competitor comparisons are relevant only after the product has established that it is meaningfully better than doing nothing, which is the actual incumbent.
- *Introduce the overtime/labor cost angle as the owner-facing value story* once the primary user (Maria or Devon) is already activated. Selling the ROI story before the operator has experienced the product creates a credibility gap that the product cannot yet fill.

Top 3 Acquisition Channels

Channel 1 — Peer Referral via Local Restaurant Communities

Why this is the right first channel: Trust is the primary purchase signal for the primary ICP. Maria does not buy software she found in a Google ad. She buys software that a peer at another restaurant told her worked. Independent restaurant operators are socially dense — they know each other through local business associations, supplier relationships, and shared neighborhood commercial interests. A single credible referral from one owner to three others outperforms 300 cold emails at this stage.

Tactics:

- Identify 5–8 regional restaurant Facebook groups, subreddits (r/restaurant, r/restaurantowners), and owner forums where Maria-type operators congregate. Join as a participant — not as a promoter — before any product exists. Understand what questions get asked repeatedly. The scheduling question will appear within two weeks.
- Draft a "problem post" template (not a product post) that opens a conversation about scheduling pain. Something like: "We're building a scheduling tool for independent restaurants and trying to understand what actually breaks — not what the software demos say breaks. Anyone willing to share what scheduling looked like last week?" This generates interviews and community credibility simultaneously.
- Design a referral mechanism into the product's free tier from the start. When Priya shares the schedule link with her staff, the staff view page should include a subtle, non-intrusive link: "Is your manager still

scheduling in a spreadsheet? They can do this instead." This is not an upsell — it is a peer signal traveling through the product's natural usage loop.

- Build a referral incentive for early pilots: one month free for every paying customer a pilot operator refers. At $49/month, this costs $49 and acquires a customer at significantly below the estimated $250–$350 blended CAC. (est.) The incentive is not the driver — the conversation opener ("I finally found something that doesn't suck") is. The incentive removes friction from the act of forwarding.
- Identify one or two local restaurant associations in target markets (state restaurant association chapters, city-level independent restaurant coalitions) and request a 5-minute presentation slot at a quarterly meeting. Do not pitch software. Present the findings from Phase 1 discovery interviews — "Here's what 15 restaurant operators told us about how they handle scheduling" — and make the tool available for free trials to attendees. Association credibility transfers to the product by proximity.
- Build a structured "case study ask" into the 90-day check-in call. By month 3, retained customers who answer positively to "would you recommend this to another operator?" should be asked for a 5-minute recorded conversation describing their before/after. These recordings become the most credible acquisition content available, deployed in community channels and in cold outreach as social proof rather than marketing copy.

Estimated CAC via this channel: $100–$200 (est.) at maturity. Zero variable cost beyond founder time in months 1–6.

Channel 2 — Targeted Cold Outbound to Floor Managers (Devon ICP)

Why this is the right second channel: Devon is email-reachable in a way Maria is not. He is more likely to have a public-facing email address (posted on the restaurant group's website, listed in Square or Toast merchant directories), more likely to read and respond to a well-targeted cold email, and more likely to be actively thinking about scheduling inefficiency as a professional problem worth solving. Cold outbound to Devon is the highest-volume early acquisition path for paid conversion.

Tactics:

- Build a prospecting list from three sources: (1) Google Maps search for "restaurant" in target cities filtered to independently owned, 2–3 location groups (distinguishable by shared branding, adjacent addresses, or the same ownership name appearing on multiple Yelp profiles); (2) Toast and Square merchant finder pages where multi-location operators are sometimes listed publicly; (3) LinkedIn searches for "general manager" + "restaurant" + city, filtered to profiles listing 2 or more employer locations.
- Draft three cold email variants, each under 90 words, each leading with a specific pain (not a feature):
  - æVariant A: "The multi-location scheduling problem" — opens with the double-booking cross-location scenario
  - æVariant B: "The Monday morning overtime conversation" — opens with the owner asking Devon to explain a labor cost spike

æVariant C: "The float staff problem" — opens with the challenge of scheduling staff who work two locations

- Send in batches of 20, measure reply rate and meeting conversion before scaling. A 3–5% reply rate on cold email to this persona is a success signal. (est.) Below 2% consistently across 60+ sends indicates either a targeting failure (wrong persona) or a messaging failure (wrong pain lead) — not a volume problem.
- Follow up every non-reply with a single one-line email after 5 days: "Not the right time?" This generates more replies than the original email in comparable SMB SaaS cold outreach patterns. (est.) Do not automate more than two touches total — restaurant operators who receive a third follow-up from an unsolicited sender report the email as spam at higher rates than other SMB segments. (directional)
- Once 5–10 Devon-type customers are active, personalize outbound with a specific reference: "I work with Devon-type floor managers at [Restaurant Group Name] in [City] — they were managing two locations in separate spreadsheets before switching." Named social proof in cold outbound to this persona is the single highest-converting personalization lever available before a formal case study exists.

Estimated CAC via this channel: $300–$450 (est.) at the volume levels projected in the first 6 months, declining as messaging sharpens and reply rates improve.

Channel 3 — POS Integration Marketplace Listings

Why this is the right third channel: Both Toast and Square operate app marketplaces where independent restaurant operators discover third-party software while already inside the tool they use daily. A listing in the Square App Marketplace or Toast Partner program places the product in a context of established trust — the operator is already paying Square or Toast, already knows their data lives there, and is in a mental frame of "software that works with my restaurant." The discovery moment is self-selected, the trust is borrowed from the POS platform, and the CAC is structurally lower than any cold outbound channel. (est.)

Tactics:

- Research both Square App Marketplace and Toast Partner Program application requirements before building any integration. Both have technical approval processes that take 4–8 weeks. (directional) Begin the application process in parallel with pilot onboarding, not after it. Waiting until the product is "ready" means losing 2–3 months of potential marketplace exposure.
- Design the POS integration to solve one specific pain before attempting to solve all integration possibilities: read the employee roster from the POS (Square or Toast) and pre-populate the staff list in the scheduling product on day one of onboarding. This eliminates the single highest-friction moment in the signup flow for POS-connected operators — re-entering 15–20 staff names manually — and creates an immediate "this is smarter than I expected" moment in the first session.
- Write the marketplace listing copy around the primary ICP's specific language: "Stop rebuilding your staff list every time someone quits. Pull your team directly from Square and have next week's schedule ready in 20 minutes." Do not write marketplace copy that sounds like it was written for an enterprise buyer — the operators reading it are Maria, not an IT procurement team.

- Request co-marketing consideration from the Toast or Square partnership team once 15–20 paying customers on the integrated plan exist. Both platforms have newsletter and in-app recommendation surfaces for highly-rated partner apps. Getting a mention in a Toast weekly newsletter to restaurant operators in one regional market can generate 50–100 new trial signups at effectively zero incremental cost. (est.) This is not achievable before the product has a review baseline and demonstrated retention — earn the co-marketing by building the product merit first.
- Evaluate a Gusto integration (payroll) as a secondary marketplace play once the Manager tier's labor cost reporting feature is built. Gusto's small business marketplace has significant independent restaurant representation, and a scheduling product that connects to both the POS (revenue side) and payroll (cost side) has a materially stronger value story for the Devon ICP's owner audience than a scheduling tool in isolation.

Estimated CAC via this channel: $150–$250 (est.) at steady state after marketplace listing is live and reviewed. Near zero before the listing is approved and visible to operators.

SEO: 10 Target Keyword Clusters

A preliminary note on SEO timing: organic search traffic compounds slowly. Ranking for most of the clusters below requires 6–12 months of consistent content output and domain authority development. SEO is not the right first-week acquisition channel — it is the right 6-month infrastructure investment that reduces paid CAC permanently once it starts producing. The keyword clusters below are ordered by a combination of conversion relevance and attainability, not by search volume alone.

(directional)

Cluster 1 — "restaurant employee scheduling" (Informational + Commercial)

Target keywords:

- restaurant employee scheduling
- how to schedule restaurant staff
- restaurant scheduling tips
- restaurant weekly schedule template

Search intent: Operators at the awareness stage — they are searching because they just had a bad scheduling week, not because they have decided to buy software. The content hook is educational (a free template, a how-to guide) that captures email and introduces the product as the logical next step.

Content form: A downloadable Google Sheets scheduling template (genuinely useful, free, no email gate initially) accompanied by a blog post: "How to Build a Restaurant Schedule in 20 Minutes." This is the

top-of-funnel piece that ranks for informational intent and converts visitors who have already decided manual tools are failing them.

Cluster 2 — "7shifts alternative" / "HotSchedules alternative" (High-Intent Comparison)

Target keywords:

- 7shifts alternative
- hotschedules alternative for small restaurants
- cheaper alternative to 7shifts
- restaurant scheduling software without monthly fees per location

Search intent: Operators who are already in the market for scheduling software, have evaluated an incumbent, and rejected it — usually on price or complexity. This is the highest conversion-intent cluster available and the one most worth competing for early.

Content form: A comparison page titled "7shifts vs. [Product Name]: Which one is actually built for independent restaurants?" structured around the three differences that matter to the primary ICP: price simplicity (flat-rate vs. tier complexity), mobile-first design, and time-to-first-schedule. Do not write this as a hit piece — write it as an honest comparison that acknowledges 7shifts is a better product for multi-unit chains and positions this product as the right choice specifically for operators running 1–3 locations who find 7shifts over-engineered.

Cluster 3 — "free restaurant schedule template" (High Volume, Low Intent, Email Capture)

Target keywords:

- free restaurant schedule template
- restaurant schedule template Google Sheets
- weekly restaurant staff schedule template Excel
- printable restaurant schedule template

Search intent: Pure informational — the operator wants a template, not a product. The value exchange is the template itself; the conversion goal is email capture and product awareness, not immediate signup.

Content form: A landing page with three genuinely useful downloadable templates (Google Sheets, Excel, printable PDF) and a soft introduction: "Or if you're tired of maintaining the template yourself, here's what using [Product Name] looks like instead." The conversion rate from this page to trial signup will be low, but the volume of operators finding it via organic search makes it worth building as a long-term organic channel.

Cluster 4 — "restaurant scheduling app" (Commercial Investigation)

Target keywords:

- restaurant scheduling app
- best restaurant scheduling app
- restaurant scheduling app for small restaurants
- restaurant staff scheduling app iPhone

Search intent: Operators who have decided a tool exists and are searching for it — mid-funnel, actively comparing. Mobile-qualified ("app," "iPhone") signals the Priya persona or a Maria who primarily works from her phone.

Content form: Product landing page optimized for this cluster, with mobile screenshots as the primary visual. The headline should be the promise, not the feature: "Your restaurant schedule, built and sent from your phone." App Store optimization (ASO) for any future mobile app should use these same keyword clusters.

Cluster 5 — "how to handle restaurant no call no show" (Problem-Aware, Solution-Adjacent)

Target keywords:

- restaurant no call no show
- how to handle employee no show restaurant
- restaurant callout policy
- what to do when restaurant employee doesn't show up

Search intent: An operator in active pain — this search happens on Saturday morning after someone didn't show up. The emotional state is frustration and urgency. The content should acknowledge the immediate situation and introduce shift coverage notification as a product feature that prevents this from being a manual scramble.

Content form: A blog post: "What to Do When a Restaurant Employee Doesn't Show Up (and How to Stop Spending Saturday Morning on Your Phone)." The post should be genuinely useful for the immediate situation (have a documented coverage protocol, keep an emergency availability list, etc.) and introduce the callout coverage suggestion feature naturally as a tool that pre-populates the available staff list so the manager isn't doing this from memory.

Cluster 6 — "restaurant overtime tracking" (Problem-Specific, High Conversion Relevance)

Target keywords:

- restaurant overtime tracking
- how to track employee overtime restaurant
- overtime alerts restaurant scheduling
- restaurant labor cost overtime

Search intent: Devon-type floor managers who have recently had an overtime conversation with their owner. This is a high-pain, specific search that signals someone who needs the overtime alert feature described in the Operator tier.

Content form: A short, data-dense blog post: "Why Restaurant Overtime Happens on Thursday (and How to Catch It on Tuesday)." The post explains the scheduling patterns that generate surprise overtime (understaffing earlier in the week leading to over-scheduling toward the weekend), uses real-operator language, and introduces the weekly hours tracker as the solution.

Cluster 7 — "restaurant staff scheduling software" (Category Entry, Commercial)

Target keywords:

- restaurant staff scheduling software
- restaurant workforce scheduling software
- scheduling software for small restaurants
- restaurant scheduling software free trial

Search intent: Active buyers searching for a product category. Higher competition from incumbents (7shifts, HotSchedules, Homebase will all rank here), but worth competing for once the product has landing page authority and genuine reviews.

Content form: Optimize the product homepage for this cluster. The homepage is the primary page competing for this intent — do not split this traffic to a blog post. The homepage headline, meta description, and first 200 words should contain the core cluster terms naturally while leading with the value promise.

Cluster 8 — "employee shift swap app" (Feature-Specific, Priya Persona)

Target keywords:

- employee shift swap app
- how to handle shift swaps restaurant
- shift trade app for restaurant employees
- employee shift swap request

Search intent: Either Priya (looking for a tool to manage a process she currently handles via text) or a Devon-type looking for a way to get shift swap requests out of his personal inbox. High specificity = high conversion relevance despite likely lower volume than broader clusters.

Content form: A landing page specifically for the shift swap feature, separate from the main product page. Title: "Stop Managing Shift Swaps in Your Text Messages." This page can be linked from the main product page as a feature detail and ranked independently for the specific cluster.

Cluster 9 — "how to make a restaurant schedule" (Instructional, Top of Funnel)

Target keywords:

- how to make a restaurant schedule
- how to write a restaurant schedule
- making a restaurant schedule for the first time
- what to include in a restaurant schedule

Search intent: New operators or newly-promoted informal schedulers (Priya persona) searching for basic guidance. Low commercial intent at the moment of search, but high lifetime value if converted — an operator who first encounters the product while learning how scheduling works is not switching from a competitor; she is building a habit from scratch.

Content form: A long-form guide: "How to Build a Restaurant Schedule (A Step-by-Step Guide for New Restaurant Managers)." This post covers the fundamentals — availability collection, shift structure, coverage ratios, communication — and demonstrates the product as the logical evolution from the manual process being described.

Cluster 10 — "[city] independent restaurant" + review/community terms (Local SEO)

Target keywords:

- [city] independent restaurant owners forum
- independent restaurant association [state]
- restaurant owner resources [city]
- small restaurant business owners [city]

Search intent: Not direct product search — this is a local community-finding intent. The SEO value is low directly, but the local presence signals (Google Business profile, local citations, city-specific content) contribute to the domain authority that improves ranking on the higher-volume clusters above.

Content form: City-specific landing pages ("Restaurant Scheduling Software for Independent Restaurants in [City]") become relevant at scale when local SEO investment can produce meaningful traffic volume. At pre-revenue stage, this cluster is worth tracking but not worth building content for.

Launch Sequence

Pre-Launch: 30 Days Before First Public Announcement

The goal of the pre-launch period is not buzz — it is a contained, controlled feedback loop with 5–8 operators who will become the social proof engine for everything that follows. A public launch without proof-of-concept customer stories is a louder version of silence.

Week 1 — Foundation

- Purchase the domain. Publish a single-page holding site with one sentence describing the product and an email capture form: "We're building scheduling software for independent restaurants. Want early access?" No feature list, no pricing, no screenshots. The email capture is the only conversion goal.
- Set up a simple analytics stack (Google Analytics 4 + Microsoft Clarity or Hotjar for session recording) on the holding page before any traffic arrives. Knowing how visitors behave on the holding page is useful; knowing it retroactively is not.
- Create a shared Google Document or Notion page to track every operator conversation, interview note, and pilot commitment in one place. No customer relationship management tool is required at this stage — a well-maintained document is sufficient and faster to maintain.
- Draft a 90-second product demo script that shows, in order: (1) adding three staff members, (2) building a one-week schedule by dragging shifts, (3) publishing the schedule, (4) showing the staff notification confirmation screen. This script is the foundation of every pilot onboarding conversation, every investor conversation, and every conference or association presentation. Write it before the product is built so it drives what gets built first.

Week 2 — Pilot Operator Outreach

- Contact the 3–5 operators from Phase 2 of the Validation Roadmap who expressed the clearest willingness to pay and offer them a free pilot with hands-on onboarding. Frame this explicitly as a co-build relationship: "You'll be helping shape what this becomes, and your first 90 days are free." This is not discounting — it is converting research participants into design partners.
- Draft the onboarding email sequence for pilot operators: (1) welcome and setup instructions, (2) a prompt to build their first schedule and share the link with their team, (3) a week-one check-in question ("What worked and what confused you?"), (4) a week-two active usage confirmation. Keep each email under 150 words. Restaurant operators do not read long emails.

- Set up a simple Slack channel or WhatsApp group for pilot operators. This is not a support channel — it is a community signal. When Maria messages the group at 11pm on Sunday saying "got through the whole schedule in 14 minutes," that message is more valuable than any feature request ticket.

Week 3 — Soft Presence Building

- Write and publish two pieces of content before any product launch:
  - æThe free restaurant schedule template (Google Sheets) with the companion blog post
  - ææA 7shifts alternative comparison page (using verified competitor pricing from publicly available pricing pages)

These pages should be live at least 3 weeks before any product launch so they have begun accumulating index time in search engines. Content published the day of a product launch ranks for nothing.

- Claim and complete the Google Business Profile for the company. For a B2B SaaS targeting local restaurant operators, local search presence is not irrelevant — operators searching for scheduling tools while at their restaurant often have location services active, and a verified business profile builds credibility signals for the domain.
- Submit the Square App Marketplace application and the Toast Partner Program application. Begin the process now, not after the product is stable. These approvals take time that cannot be compressed later.

Week 4 — Pre-Launch Content Stack

- Record two short-form videos for Instagram Reels and TikTok (under 60 seconds each):
  - æ"How I used to make the restaurant schedule (and what I do now)" — shot from the operator's perspective, not the founder's. If even one pilot operator is willing to record this, it is 10x more credible than a founder-produced demo.
  - æ"The thing that kills your Sunday night (and how to fix it in 20 minutes)" — a screen recording of the product solving the core use case, no voiceover required.

These videos do not need to go viral. They need to exist so that when an operator finds the product and searches for "real people using it," they find something.

- Create the product's first three landing pages: (1) homepage, (2) the 7shifts alternative comparison page, (3) a landing page specifically for the "build your first schedule in 20 minutes" promise. Each page should have a unique value proposition and a single call to action: start a free trial.

Launch Week

The launch event is not a ProductHunt post or a press release. It is the moment the product becomes publicly available for free trials, announced in the channels where the primary ICP actually spends time.

Day 1 — Community Announcement

- Post to 3–5 restaurant operator Facebook groups and subreddits where the founder has been a participant (not a promoter) for the prior 30 days. The post should not be a product announcement —

it should be a story: "I spent the last three months talking to 15 independent restaurant operators about scheduling. Here's what I learned, and here's what I built." Link to a landing page, not the product itself. The story earns clicks; the product earns signups.

- Email the full pre-launch list (everyone who submitted their email address on the holding page) with a personal note from the founder, not a marketing template. Three sentences: what was built, what it costs (free to try), and a direct link to sign up. Restaurant operators respond to human-sounding emails, not "We're thrilled to announce the launch of our revolutionary scheduling platform."

Day 2–3 — Pilot Operator Social Proof Activation

- Ask each active pilot operator to share their experience in one public venue of their choosing — a Facebook group post, a Google review, a text to a peer operator — with no script, just an honest description of their experience. Provide a direct link to share and a suggested framing: "I've been trying this for [X weeks] and here's what I noticed." Unsolicited-sounding peer endorsements in context convert at significantly higher rates than formal testimonials displayed on a website. (est.)
- Send the first cold outbound batch (20 emails, Devon ICP, Variant A) timed to go out on Wednesday morning — mid-week, mid-morning delivery increases open rates for this persona. (directional)

Day 4–5 — Monitoring and Response

- Monitor the community channels where the launch was announced. Respond to every comment, question, and criticism within 4 hours during launch week. Speed of response is itself a signal to operators who are evaluating whether this is a real company or a tool that will disappear. A founder who replies to a Facebook comment at 7pm on a Wednesday tells Maria more about this company's commitment to independent restaurants than any marketing copy can.
- Track signup source for every trial account created during launch week. Source tagging matters now because the channels that generate the first 20 signups are disproportionately predictive of the channels worth scaling. If 12 of the first 20 signups come from one Facebook group and 2 come from all cold outbound combined, that is a signal that should redirect the next 60 days of acquisition effort.

Day 7 — Reflection and Adjustment

- Conduct a brief internal review: how many trial signups, from what sources, with what behavior in the first session (did they build a schedule, did they add staff, did they publish anything)? Write down the answers before interpreting them. The numbers will be small. The patterns will be meaningful.

Post-Launch: 30 Days After Launch

Week 1 Post-Launch — First Retention Signal

- Personally call or text every trial user who signed up during launch week and has not published a schedule within their first 7 days. Not an automated email — a personal message: "Hey, I noticed you signed up last week and wanted to make sure it was clear how to [specific action]. Can I help you build your first schedule right now?" This is not scalable. It is not meant to be. It is the onboarding behavior that teaches the founder exactly where the product loses people before automation makes that learning invisible.

- Log every piece of qualitative feedback from trial users — support messages, Facebook comments, pilot check-in calls — into the shared tracking document. Look for the phrase that repeats three or more times in the first 30 days of live users. That phrase is the real positioning statement, written by customers, not by the founder.

Week 2 Post-Launch — Content Follow-Through

- Publish the second SEO content piece (the "restaurant no call no show" post or the "overtime tracking" post — whichever maps more directly to the pain that surfaced most frequently in launch week feedback). Don't publish on schedule; publish in response to what the first users are telling you matters.
- Submit the product to 3–5 software review sites where restaurant operators search for tools: Capterra, GetApp, G2, Software Advice, and Trustpilot. Create the vendor listing with complete information (screenshots, pricing, feature list) before requesting reviews. Ask each active pilot operator for a review with a direct link. Three verified reviews on Capterra with an average of 4 stars or above changes the competitive posture meaningfully for operators who cross-check products before buying. (est.)

Week 3–4 Post-Launch — Channel Scaling Decisions

- Review the first 30 days of acquisition data against the channel hypotheses. Ask three specific questions:
  - Which channel produced the highest absolute number of trial signups?
  - Which channel produced the highest percentage of signups who built at least one schedule?
  - Which channel produced the longest average session time in the first visit?

The right channel to scale is the one that answers all three positively — not just the channel with the most raw signups. A channel that produces signups who never build a schedule is producing CAC without producing customers.

- Send the second cold outbound batch (20 emails, Devon ICP, Variant B — the overtime morning variant) with a refinement based on any reply patterns from the first batch. If Variant A produced responses mentioning "I get in trouble every time someone works overtime," lead Variant B with that exact language, not the language the founder originally wrote.
- Evaluate paid acquisition (Google Search ads on Cluster 2 competitor alternative terms, Meta ads targeted to restaurant owner job titles in specific markets) as a channel to test once 10 paying customers are retained past 60 days and unit economics can be estimated from real data rather than model assumptions. Running paid ads before retention is proven produces a leaky bucket — every paying customer who churns at month 2 makes paid CAC unrecoverable.

Competitor Positioning Matrix

The matrix below maps the four most relevant competitors across the dimensions that matter to the primary ICP. It is descriptive, not adversarial — the goal is to identify the positioning space this product can occupy credibly, not to make claims about competitors that cannot be defended.

7shifts	Homebase	When I Work	Spreadsheet + Group Text	[This Product]
Primary Target	Multi-unit restaurant groups (5–50+ locations)	Small business (retail, food service, any hourly)	Small-to-mid service businesses (non-vertical)	Any operator who hasn't decided to pay for software
Pricing Structure	Per-location tiers ($29.99–$135/month per location)	Free tier + per-location paid plans ($20–$80/month)	Per-user ($2.50–$6/user/month)	$0
Pricing Predictability for Maria	Medium — predictable per location, but tier confusion at higher plans	Medium — free tier is limited; upsell path is complex	Low — cost scales with every new hire	High — always $0
Mobile Experience	Good (dedicated app, well-reviewed)	Good	Moderate	Poor (Google Sheets mobile is functional but not purpose-built)
Time to First Schedule	Moderate — setup requires location config, role setup, integration decisions	Fast for basic use; slower for notifications and approval flows	Moderate — per-user invite flow creates onboarding friction	Fast (the Google Sheet is already built)
Overtime Alerts	Yes (mid-tier and above)	Yes (paid plans)	Yes (paid plans)	No
Multi-location Support	Yes (core use case)	Yes (paid plans)	Yes	Operator-built manual process
Staff Notification Layer	Yes (push, email, SMS)	Yes (paid)	Yes	No — relies on group text
POS Integration	Toast, Square (strong integration)	Square, Gusto, Toast	Square, ADP	None
Built for independent restaurants specifically?	No — built for chains, adapted downmarket	No — horizontal small business tool	No — horizontal	N/A
Perceived complexity	High ("built for someone with an HR department" — direct quote from primary persona)	Medium	Medium	Low (Maria already knows how to use it)
Churn risk from competitor	Low in near-term — 7shifts customers tend to be larger and more embedded	Medium — Homebase free tier users are close to the primary ICP	Low — per-user pricing creates friction for the primary ICP at scale	High — the spreadsheet is always free and always available

Competitive space summary:

The clearest open positioning is the gap between the spreadsheet (free, familiar, functionally limited) and 7shifts (full-featured, complexity-high, priced per location). That gap is occupied by Homebase's free tier — which lacks the notification layer — and by no paid product that is explicitly designed for single-location or very small multi-location independent operators and priced as simply as a flat monthly rate.

The product does not need to beat 7shifts at multi-location enterprise scheduling. It needs to beat the spreadsheet at independent restaurant scheduling, in a way that Homebase's free tier does not because it

lacks SMS notifications and overtime alerts, and that 7shifts' paid tier does not because its pricing model creates hesitation at the point of commitment.

The competitive threat worth monitoring most closely is not 7shifts pushing downmarket with simplified pricing — it is Toast or Square building native scheduling into the POS dashboard as a bundled feature with no incremental cost. As noted in the Product Brief, this risk is assessed as medium likelihood and high impact. If either POS platform launches native scheduling in the 12–18 month window, the positioning above must shift immediately from "better than your spreadsheet" to "works with Toast/Square and does the specific things their native tool doesn't" — which requires the referral and integration relationships described in Channel 3 to already be in place before that moment arrives.

Key GTM Takeaways

- *The first 30 days are about trust, not reach.* The temptation at launch is to optimize for the widest possible distribution. The right optimization is for the deepest possible signal from the smallest possible audience. Five operators who used the product, told a peer, and renewed after 60 days are worth more than 500 trial signups from a Facebook ad that produced no retention data.
- *SEO and community channels compound; cold outbound converts now.* Build the content infrastructure for organic acquisition today because it takes 6–9 months to produce meaningful traffic. (est.) Run cold outbound to Devon-type floor managers in parallel because it produces pipeline this quarter. Do not mistake SEO effort for GTM progress — SEO is a 12-month investment with delayed returns, not a 30-day acquisition strategy.
- *The positioning matrix is a living document.* Competitors change their pricing and feature sets. Toast and Square bundling decisions are the highest-impact competitive variable. Update the matrix every quarter with verified pricing information, not memory. The B2B SaaS market is projected to grow significantly through 2034 [source: mordorintelligence.com/industry-reports/b2b-saas-market], which means more entrants and more feature bundling from incumbent platforms are structurally likely — the positioning space that exists today may compress faster than the launch timeline suggests.
- *GTM kill conditions mirror product kill conditions.* If Channel 1 (peer referral) produces zero organic referrals after 10 active customers, that is a signal that retention-driven advocacy is not working — either because customers are not retained long enough to refer, or because the product does not generate the kind of outcome that operators talk about unprompted. Channel failure at this stage is more often a product signal than a marketing signal. Check the churn data before changing the channel strategy.

10 Financial Projections

Financial Projections: Restaurant Staff Scheduling SaaS

A Note on the $323K ARR Figure

The Executive Summary references "100–300 paying customers H $323K ARR." That figure does not reconcile with the $55/month blended ARPU established in the Monetization section: 300 customers × $55 × 12 = $198K ARR, not $323K. Reaching $323K ARR at $55 ARPU requires approximately 490 paying customers. The projections below use the $55 ARPU assumption from the Monetization section as the authoritative input and flag where the ARR targets diverge. All forward-looking revenue figures are model-derived. (est.)

3-Year P&L Projection

Year	Revenue	COGS	Gross Margin	Operating Expenses	Net Profit / (Loss)
Year 1	$34,000	$6,200	81.8%	$57,000	($29,200)
Year 2	$214,000	$28,000	86.9%	$157,000	$29,000
Year 3	$534,000	$56,000	89.5%	$395,000	$83,000

All figures (est.).

Key Assumptions

Customer Growth Model

Revenue is heavily back-loaded in Year 1 because the validation roadmap does not begin paid pilots until Month 3, and the pilot cohort is intentionally small. The model assumes the following end-of-year (EoY) customer counts: (est.)

- EoY Year 1: 150 paying customers; average paying customers over the year H 50 (due to the ramp from zero)
- EoY Year 2: 500 paying customers; average over the year H 320
- EoY Year 3: 1,200 paying customers; average over the year H 800

These counts produce annual recognized revenue of $34K, $214K, and $534K respectively, not ARR run rates. The EoY Year 2 ARR run rate reaches $330K (est.) — the point at which the $323K figure from the Executive Summary becomes achievable, approximately 6 months later than initially projected.

Churn Rate

Base case assumes 6% monthly churn in Year 1 improving to 4.5% in Year 2 and 3.5% in Year 3 as the product matures and switching costs accumulate through POS integration and historical schedule data. If churn stays at 10% — the Kill Condition K4 threshold — Year 2 average customers drop to approximately 220 and revenue falls to ~$145K. (est.)

ARPU

$55/month blended (80% Operator at $49/month, 20% Manager at $89/month), as established in the Monetization section. Manager tier is not available until Phase 2 (Month 3–6), so Year 1 ARPU is closer to $50/month until the Devon persona is served. ARPU assumption carries forward unchanged in the base case; a shift toward 35% Manager tier adoption in Year 3 would lift blended ARPU to approximately $62/month. (est.)

COGS Composition

COGS covers four categories: infrastructure (Vercel Pro $20/month, Supabase Pro $25/month scaling to dedicated plan at 500+ accounts at ~$200/month, Trigger.dev escalating from $0 to $50/month at >50K runs, Upstash Redis, Anthropic API at ~$50–$100/month in Year 2+), Twilio SMS costs (approximately $1.00–$1.50 per account per month at 30 staff × 4 weekly SMS sends × $0.0079/message plus carrier fees), and Stripe payment processing at 2.9% + $0.30 per transaction. Payment processing alone represents $985 in Year 1, $6,200 in Year 2, and $15,500 in Year 3 at projected revenue levels. (est.)

Gross margin expands from 82% to 90% over three years as infrastructure costs scale sub-linearly relative to revenue — a standard SaaS infrastructure leverage pattern. The primary cost variable is Twilio SMS volume, which scales linearly with accounts and sets a floor on gross margin improvement.

Operating Expenses

The dominant OpEx line in all three years is customer acquisition. Using the Monetization section's blended CAC estimate of $250–$350 (est.), the model uses $300 in Year 1, $275 in Year 2, and $250 in Year 3 (reflecting referral channel maturation per the GTM strategy). This produces CAC spend of $45,000 in Year 1 (150 new customers), $110,000 in Year 2 (400 net new customers accounting for churn replacement), and $200,000 in Year 3 (800 net new customers). (est.)

The model assumes a single founder operating at zero salary through Year 1 (bootstrapped). Year 2 adds one part-time customer success hire at $40,000 annually to manage churn risk as the customer base scales beyond 150 accounts — the point at which one-on-one onboarding from the founder becomes unsustainable. Year 3 adds two full-time hires (one engineer, one customer success) at a combined $180,000. No venture-backed hiring plan is modeled here; the staffing level is calibrated to a capital-efficient path to profitability. (est.)

Startup Cost Breakdown

Category	One-Time Cost	Recurring Monthly
Infrastructure setup (Vercel Pro, Supabase Pro, Twilio A2P 10DLC brand + campaign registration)	$600	$95
Legal and compliance (privacy policy, Terms of Service drafting, DPA review across Supabase/Twilio/Vercel/PostHog/Anthropic)	$4,500	$0
Domain, design assets, product branding	$500	$0
Validation research (Phase 1–2 interview tools, travel, mockup hosting, cold outreach credits)	$600	$0
Marketing launch materials (landing page copywriting, first-channel setup for Facebook group outreach)	$1,000	$200
Accounting and business registration	$800	$50
Total	$8,000	$345

Pre-revenue monthly burn is $345/month in infrastructure and fixed services before any marketing or salary spend. This is intentionally lean — the four-service topology (Vercel, Supabase, Trigger.dev, Upstash) was selected in part because it keeps the pre-pilot monthly cost below $100 in pure infrastructure fees. (est.)

Break-Even Analysis

Inputs (base case):

- Monthly fixed costs (infrastructure + amortized legal/admin): $600
- Ongoing monthly marketing spend (CAC amortized at 15 new customers/month × $300): $4,500
- Variable cost per customer per month (SMS + Stripe): $2.00
- Average revenue per customer per month: $55

Contribution margin per customer: $55.00 $2.00 = $53.00 per month (est.)

Break-even customer count (covering fixed + marketing): ($600 + $4,500) / $53 = 96 customers (est.)

At the projected ramp rate, 96 paying customers is reached in approximately *Month 9–10. The model designates Month 10* as the break-even month. This aligns with Kill Condition K5 assessment at Month 9 — the empirical CAC audit will either validate this timeline or reveal that the acquisition cost assumption needs revision before crossing break-even. (est.)

Sensitivity to CAC: If actual blended CAC runs at $400 (above the estimated $250–$350 range) (est.), the break-even customer count rises to 126, pushing break-even to approximately Month 12–13. This is the single most sensitive variable in the break-even calculation — a 33% CAC increase extends break-even by two to three months.

Sensitivity to churn: At 10% monthly churn (Kill Condition K4), net customer growth requires acquiring roughly 1.1 customers to net one, increasing effective CAC per retained customer to ~$330+ and pushing break-even to approximately 110 customers (Month 11). (est.)

Funding Strategy

Recommended approach: Bootstrapped through Year 1, with a targeted pre-seed raise at Month 9–10 if and only if empirical validation supports it.

This product is not a venture-scale bet at launch. The $34K Year 1 revenue and $29K Year 2 net profit model (est.) do not support a Series A narrative. More importantly, the fundamental questions — will independent restaurant operators pay $49/month, will CAC be recoverable within 12 months, will churn stay below 10% — cannot be answered with capital. They can only be answered with customers. Raising money before those questions are resolved converts a validation exercise into a spend exercise, with venture expectations attached to an unvalidated business model.

The $8,000 startup cost and $345/month pre-revenue burn rate are designed to operate within a solo founder's personal runway. The validation roadmap through Month 6 costs approximately $2,700 in direct expenses (Phase 1: $200, Phase 2: $400, Phase 3: $2,000, plus $100/month infra). A founder with 6 months of personal runway and $10,000 in savings can complete the full validation without external capital.

If Kill Conditions K1–K3 pass and the product reaches 20+ retained paying customers by Month 6, a pre-seed raise of $150,000–$250,000 becomes rational. The use of proceeds is specific: fund the CAC required to reach 100+ customers (100 customers × $300 CAC = $30,000), hire a part-time customer success resource ($20,000), complete SOC 2 Type I gap analysis ($15,000), and extend founder runway for 12 months ($80,000–$160,000). (est.)

At that raise size, the appropriate sources are angel investors with direct restaurant or hospitality technology experience — specifically founders or operators who have navigated the SMB SaaS adoption problem before (companies like 7shifts, Toast, or Olo have alumni networks worth targeting). A $150K–$250K pre-seed from 2–4 angels does not require a board, does not impose a venture-return pressure that forces growth over profitability, and provides meaningful signal on the business model's credibility. (est.)

Avoid VC at pre-seed for this product. The restaurant SaaS market is well-understood by institutional investors, and the independent operator segment is explicitly the segment that 7shifts, Homebase, and Sling

have already captured at their lower tiers. A VC-funded company in this space needs to demonstrate a path to $10M+ ARR quickly, which requires either a much higher price point than $49/month, a volume of customers that requires significant sales infrastructure, or a land-and-expand model into multi-location groups — none of which are validated at this stage. Revenue-based financing is an option at Month 12+ once MRR exceeds $15,000/month, but the repayment multiples (typically 1.35–1.5×) make it more expensive than angel equity at this revenue level. (est.)

Sensitivity Analysis

Scenario	Key Assumption Changed	Year 1 Revenue	Year 2 Revenue	Break-Even Month
Pessimistic	Monthly churn 10%; CAC $400; free-to-paid conversion 20% below base	$19,000	$98,000	Month 14–15
Base	6% monthly churn; $300 CAC; ARPU $55	$34,000	$214,000	Month 10
Optimistic	4% monthly churn; $200 CAC via referral dominance; ARPU $60 (Manager tier uptake)	$51,000	$338,000	Month 7–8

All figures (est.).

Pessimistic scenario mechanics: At 10% monthly churn, the customer base requires a higher gross acquisition rate to achieve the same net customer count. Combined with a $400 CAC — plausible if cold outbound to non-technical, time-constrained restaurant owners converts at half the assumed rate — Year 1 revenue falls to approximately $19,000 and Year 2 to $98,000. This is the scenario in which Kill Condition K4 and K5 both trigger at Month 9, and the correct response is a controlled wind-down rather than continued investment.

Optimistic scenario mechanics: The GTM strategy identifies peer referral as the lowest-CAC channel at $100–$200 per customer (est.). If referrals account for 60% of acquisition volume by Month 6, the blended CAC falls to approximately $200. Combined with Manager tier adoption above the 20% baseline assumption (plausible if the Devon persona converts at pilot stage), ARPU rises to $60 and break-even arrives at Month 7–8. This is the scenario in which a pre-seed raise should be aggressive and timed before Month 9 to fund the growth inflection.

The variable that dominates all three scenarios is not revenue — it is churn. Moving from the pessimistic 10% monthly churn to the base case 6% more than doubles Year 2 revenue. Moving from base to optimistic 4% adds another 58%. No acquisition investment, pricing optimization, or feature expansion produces a comparable impact on the financial model. This is the precise mathematical reason the Adversarial Review describes reversion to spreadsheets as a revenue-threatening event, not merely a retention metric.

KEY TAKEAWAYS

- Monthly churn is the dominant financial variable, not customer acquisition. The difference between 6% and 10% monthly churn cuts Year 2 revenue by more than half (est.) — a gap no realistically achievable improvement in CAC or ARPU can offset. Every product decision that reduces the probability of a single bad Sunday sending Maria back to her Google Sheet is more valuable than any marketing optimization.
- The business reaches cash-flow positive at approximately Month 10 on a bootstrapped basis (est.), but the unit economics are fragile enough that a single bad assumption — CAC 33% above estimate, or churn at Kill Condition K4 threshold — extends break-even to Month 14–15 and materially changes the funding requirement. The Month 9 empirical CAC and churn audit is not a milestone; it is the decision point that determines whether to invest forward or apply Kill Condition K5.
- Year 1 recognized revenue of ~$34K (est.) understates the business's trajectory because revenue is ramp-constrained, not market-constrained. The EoY Year 1 ARR run rate of $99K (est.) and EoY Year 2 run rate of $330K (est.) are the investor-relevant metrics — but they are only credible if the pilot cohort retention data from Months 3–9 demonstrates that the product holds customers through the high-stress operational moments when the reversion path is always one tap away.

11 Product Design

UI / Interface Design

Overview

This section defines the structural and visual decisions that govern the product's interface. Two constraints shape every choice here: the primary user (Maria, scheduling on a phone on Sunday night, in a restaurant, under pressure) and the primary failure mode (a manager who opens the product, encounters friction, and returns to the Google Sheet she already knows).

The interface does not need to be impressive. It needs to complete one job in one session without requiring Maria to think. Every navigation decision, every component choice, and every micro-interaction is evaluated against that standard first. Visual sophistication is a Version 2 concern. Functional clarity at first contact is the condition for there being a Version 2.

The B2B SaaS market context is relevant here not for design inspiration, but for competitive framing: as the market grows toward USD 4,441.49 billion by 2034 [source: mordorintelligence.com/industry-reports/b2b-saas-market], the density of scheduling tools increases, and the switching cost from any of them to this product is never lower than at first contact. The interface must justify the switch in the first session, or it will not get a second one.

Key Screens

Each screen below is defined by its job — the specific outcome it must produce for a specific user — not by its feature set.

Screen 1: Onboarding — Roster Setup

Job: Get Maria from account creation to a populated roster in under 5 minutes, without requiring her to read instructions.

What it must do:

- Accept staff entry via three paths: manual name entry, CSV upload, and (Phase 4+) POS import. All three paths are visible on first load. No onboarding wizard that hides options behind a linear flow.
- Show a live count of staff added as each entry is confirmed ("3 of your team added — keep going or build your first schedule").
- Surface a "skip for now" option on every optional field (phone number, role, hourly rate) so Maria can start building before she has every detail.
- Fire the roster_uploaded event the moment the first staff member is saved.

What it must not do:

- Ask Maria to configure notification settings, timezone, or integrations before she has added a single person. Those are Phase 2 concerns; they do not belong at the entrance.

Screen 2: Schedule Builder

Job: Allow Maria to assign every staff member to every shift for the coming week in under 20 minutes, on a phone, with no prior training.

What it must do:

- Display a 7-column weekly grid with staff names as row headers. Each cell represents one shift slot for one day.
- Provide a tap-to-assign interaction: tapping a cell opens a bottom sheet with shift time presets (Breakfast, Lunch, Dinner, Custom) — not a time picker with spinning wheels.
- Show a running total of scheduled hours per staff member in the row header, updating live as shifts are added.
- Flag overtime risk inline: if a staff member crosses the configured threshold (default: 40 hours), the hours total changes to an amber warning state with a text label ("38 hrs — near overtime"), not just a color change.
- Surface an "Add open shift" option per day for shifts not yet assigned to a specific person.
- Include a one-tap "Copy last week" function accessible from the top of the builder. This is not a power feature — it is the first thing Maria will look for, because her schedule rarely changes more than 20% week to week.

What it must not do:

- Render the weekly grid as a fixed-width horizontal table that requires horizontal scroll on a 375px-wide phone. The grid must reflow or paginate to a day-by-day view on screens below 480px.
- Require a save action. Every shift assignment auto-saves and displays a persistent "Saving…" / "Saved" indicator in the header — silent until it fails, at which point an inline error appears with a retry action.

Screen 3: Publish & Confirm

Job: Give Maria a single, unambiguous action to send the schedule to her entire team, and confirm that the team received it.

What it must do:

- Present a pre-publish summary: number of staff scheduled, total hours, uncovered open shifts, and any overtime flags. This is not a warning screen — it is a confidence screen. Maria should feel ready to tap "Send."
- Offer two send options: SMS notification to all staff with a schedule link, and a copyable link for manual sharing (WhatsApp, group text). Both are visible on the same screen; SMS is the default.
- Show a real-time delivery status panel after sending: a list of staff names with a delivery indicator per person ("Sent," "Delivered," "Failed"). This panel is not a modal — it persists as part of the screen so Maria can see delivery happen in real time without navigating back.
- Surface a "Resend to undelivered" action that appears automatically when one or more delivery failures are detected.
- Fire schedulepublished on send and smsconfirmation_received on each confirmed delivery receipt.

What it must not do:

- Route Maria to a success screen that hides the delivery status behind a "View details" link. The delivery panel is the success state.

Screen 4: Staff Schedule View (No Login Required)

Job: Show a staff member their shifts for the week in a format they can read in 10 seconds on any phone, without creating an account.

What it must do:

- Display shifts in a vertical card list sorted by day and time, not a grid. A grid is a manager's view. A staff member needs to know: what days am I working, what time do I start, what time do I end.
- Show only the viewing staff member's shifts by default. Include a "View full schedule" toggle to see the team schedule — useful for Priya, who informally coordinates for others.
- Include an "I can't make this shift" action on each shift card that triggers a notification to the manager. This action does not require login and does not require the staff member to propose a replacement — that friction belongs to the manager screen, not the staff view.
- Display the restaurant name, week, and manager contact name prominently. Staff who receive a link with no context click away.

What it must not do:

- Prompt for account creation before displaying the schedule. The schedule is the product. The account creation prompt, if present at all, is a secondary action at the bottom of the page ("Want to set your availability? Create a free account").

Screen 5: Callout Coverage

Job: When a staff member can't make a shift, surface available replacements in under 30 seconds without requiring Maria to scroll through her contacts list.

What it must do:

- Trigger from two entry points: the manager marking a shift uncovered, or a staff member tapping "I can't make this shift" from the staff view.
- Display ranked available staff based on three factors: (1) not already scheduled that day, (2) worked this shift type previously, (3) under overtime threshold. Ranking logic is not exposed to the user — the list just appears in order.
- Allow one-tap notification to available staff ("Offer this shift"), with the message pre-filled ("Can you cover [Shift Time] on [Day]? Reply YES or NO").
- Show a response status view: a list of staff contacted, their response status (Pending, Yes, No), and a "Confirm [Name]" action that adds the shift to their schedule.
- Fire calloutcoveragesuggested on display and log timetocover_minutes from the moment the shift is marked uncovered to the moment it is confirmed filled.

Screen 6: Labor Cost Report

Job: Give Devon (and Maria's owner) a one-screen view of scheduled labor cost for the week, segmented by role, against a configurable budget target.

What it must do:

- Display total scheduled labor cost for the current week and the three prior weeks as a simple bar chart. No pie charts, no trend lines, no axis labels requiring interpretation.
- Show a role-level breakdown (FOH, BOH, Management) as a flat list with hours and estimated cost per role.
- Surface a "vs. budget" comparison if a weekly labor budget has been set. If no budget is set, replace the comparison with a prompt to set one ("Set a labor budget to track against it").
- Export as CSV and tagged PDF via a single button. The export button is always visible — not behind a "More" menu.

What it must not do:

- Display per-employee cost data on the default view. This is sensitive compensation information visible on a shared tablet or over-the-shoulder in a restaurant. Per-employee detail is behind a tap, not shown by default.

Screen 7: Settings

Job: Allow Maria or Devon to configure the product without requiring support.

Sections:

- *Roster management:* Add, edit, or archive staff. Set roles and hourly rates (optional). Manage availability templates.
- *Notification settings:* Configure SMS sender name, default notification timing (e.g., send schedule every Sunday by 8pm), overtime threshold (default 40 hours, adjustable).
- *Location management (Manager tier only):* Switch between locations, manage float staff who appear on multiple location rosters.
- *Billing:* View current plan, usage, next billing date, and upgrade or downgrade path. No support ticket required to change plan tier.
- *Integrations (Phase 4+):* POS connection status, payroll export settings.

Settings is a utility screen, not a discovery screen. It does not need onboarding tooltips or feature spotlights. It needs clear labels and a search field when the section list exceeds 8 items.

Navigation Structure

The navigation model must serve three distinct usage patterns without a single navigation structure that optimizes for the wrong one:

- *Maria (Sunday-night session):* Deep task focus. She opens the app to build and publish one schedule. She does not want a dashboard — she wants to be in the schedule builder immediately.
- *Devon (mid-week check-in):* Quick scan. He opens the app to check overtime alerts and confirm coverage. He wants the labor report and the week's status at a glance.
- *Priya (daily mobile check):* Single-purpose. She opens the app to see her shifts. She is on the staff view, not the manager app at all.

Given these patterns, the manager application uses a *bottom tab bar on mobile* with five items:

Tab	Icon	Primary User	Purpose
Schedule	Calendar grid	Maria	Opens directly to the current week's schedule builder or a published schedule view
Coverage	Person with checkmark	Maria / Devon	Active callout and open shift management
Labor	Bar chart	Devon	Labor cost report and overtime alerts
Team	Group of people	Maria / Devon	Roster management and staff availability
Settings	Gear	Maria / Devon	Account, notifications, billing

Tab order rationale: Schedule is first because it is the highest-frequency action. Settings is last because it is the lowest-frequency action. Coverage is second, not third, because the callout resolution flow has the highest urgency when it is active — a manager who opens the app at 6am on a Saturday because someone called out needs Coverage immediately visible.

On desktop (≥1024px): Bottom tab bar converts to a persistent left sidebar with the same five items plus a collapsible label. The sidebar does not auto-collapse. Restaurant managers using a desktop browser are typically at a back-office station and benefit from persistent context.

No hamburger menu. A hamburger menu on a product used in 90-second sessions by operators under operational stress is a navigation system that requires the user to remember where things are hidden. Every primary action is always one tap from the current screen.

Component Hierarchy: Schedule Builder

The schedule builder is the core product screen. Its component hierarchy governs rendering performance, accessibility structure, and the interaction model for the primary conversion flow.

Key structural decisions:

- The PublishButton lives in the ScheduleHeader, not at the bottom of the screen. It is always visible. Maria should never have to scroll to find the action that ends her session.
- The ShiftEntryBottomSheet renders as an overlay above the grid, not as a modal that removes the grid from view. Maria should be able to see which cell she tapped while entering shift details.

- The StaffNameCell is sticky on horizontal scroll. On a 7-day grid on a phone, days 5–7 require horizontal scroll. The staff member's name must remain visible during that scroll or the grid becomes disorienting.
- The LaborSummaryBar is sticky below the header and above the grid. It does not scroll with the grid. Devon's primary use of the builder is watching the labor total — he should see it without scrolling.

Design Principles

Five principles, each grounded in a specific product constraint or user behavior observed in the persona research.

---

1. The First Session Is the Product

Maria's decision to pay $49/month is made or lost in the first session. Every design decision that delays the moment she publishes her first schedule and receives a staff confirmation — extra onboarding steps, required fields, settings she doesn't understand yet — is a churn event disguised as a feature.

Applied: All optional fields (hourly rates, availability preferences, integration settings) are deferred from the first-session flow. The onboarding path is: add staff ’ add shifts ’ publish. Everything else is available, but nothing else is required. First-session completion rate is tracked as an activation metric, not a secondary concern.

---

2. Mobile-First Means Mobile-Actual

Mobile-first is often a design principle applied to desktop layouts that are then responsively compressed. This product's primary user is on a phone, in a restaurant, tapping with one hand, in a physically demanding environment. Mobile-first here means: design the phone layout first, test it on a real mid-range Android device under bright light, and then adapt it for larger screens — not the reverse.

Applied: The schedule grid does not render as a 7-column horizontal table on phones below 480px. It renders as a day-by-day vertical layout by default, with a "Week view" toggle for users who prefer the grid. Tap targets are a minimum of 44×44px on every interactive element. No feature requires a hover state to be functional.

---

3. Status Is Always Visible, Never Hunted

Restaurant operators make real-time decisions based on partial information. A manager who has to navigate to a separate screen to find out whether their schedule was delivered, whether someone is approaching overtime, or whether a callout has been covered will stop trusting the product as a real-time operational tool and return to the group text thread where status is immediately visible.

Applied: Delivery status, overtime flags, and open coverage gaps surface at the point of action — not in a notification center, not in a separate reporting view. An overtime flag appears in the staff row, next to the person's name, while the schedule is being built. A callout gap appears as a red indicator on the Coverage tab badge and on the specific shift cell, simultaneously.

---

4. Errors Are Instructions, Not Judgments

The operator audience for this product did not choose a scheduling software career. They chose restaurants. A product that responds to their mistakes with "Invalid input" or silent failures is a product that will be blamed the next time a shift goes uncovered. Every error state must tell the user what happened, why it happened, and what to do next — in plain language, in the context where the error occurred.

Applied: Every inline error message follows the format: "[What failed] — [Why it failed] — [How to fix it]." Example: "SMS not delivered to Jordan — phone number may be incorrect. Update Jordan's contact." This message appears inline in the delivery status panel, not in a toast that disappears after three seconds.

---

5. Simplicity Is Not Minimalism

There is a failure mode in "simple" SaaS design where features are hidden or removed in pursuit of visual cleanliness, and the product becomes non-functional for users with real complexity. Maria's restaurant has 18 staff members, some of whom have split availability, two of whom are minors with hour restrictions, and one of whom always requests Saturday off. The product must handle that complexity — it cannot pretend the complexity doesn't exist.

Applied: Complexity is accessible rather than absent. Advanced features (availability templates, minor labor law flags, role-based overtime rules) are present in the product but not surfaced in the default first-session experience. They are discoverable from within the relevant context — the staff profile, the shift cell, the settings panel — not promoted in the primary navigation or the onboarding flow.

Responsive Breakpoints and Mobile Approach

Breakpoint	Width	Layout Behavior
Mobile S	320px – 374px	Day-by-day schedule view only. Bottom tab bar. Sin-gle-column forms. Reduced font scale (14px base).

Mobile M	375px – 479px	Default mobile layout. Day-by-day schedule default with week view toggle. Bottom tab bar. 16px base font.
Mobile L / Phablet	480px – 767px	Week grid view becomes available as the default toggle. Bottom tab bar persists.
Tablet	768px – 1023px	Week grid view default. Bottom tab bar converts to a top tab bar or a narrow left sidebar (configurable). Two-column layout available for settings and roster screens.
Desktop	1024px+	Persistent left sidebar navigation. Full week grid default with labor summary bar to the right of the grid. Three-column layout on settings.

Mobile-first implementation rule: All components are built to the Mobile M (375px) spec first. Larger breakpoints are additive — they add layout columns, expand the schedule grid, and surface contextual panels — but they do not remove or reorganize the core interactions that mobile users depend on.

Device testing baseline: Functional testing is required on the following device targets before each release gate:

- iPhone SE (2nd gen, 375px, iOS)
- Samsung Galaxy A-series (360px, mid-range Android)
- iPad (768px, tablet layout)
- Chrome on Windows at 1440px

Performance target: First Contentful Paint under 2.5 seconds on a 4G connection for the Schedule Builder screen. A manager opening the app in a restaurant does not have guaranteed WiFi.

Dark / Light Mode Recommendation

Recommendation: Light mode only at launch. Dark mode in Phase 5+.

Reasoning:

The argument for dark mode is genuine: the product will be used in dimly lit restaurants, on phones with OLED screens, by users who have dark mode enabled system-wide. A product that ignores system dark

mode preference will render with a jarring white screen in low-light environments, which is a real UX failure for late-night shift publishing.

However, dark mode is not a single design decision — it is a parallel design system. Every color token, every component state (default, hover, focus, error, disabled), every data visualization (the labor bar chart, the overtime badge, the delivery status indicators) must be specified and tested in both light and dark contexts. For a product at the pilot stage with a small design and engineering team, maintaining two parallel design systems before product-market fit is confirmed introduces rework risk that is not justified by the benefit.

The mitigation at launch: Honor the prefers-color-scheme: dark media query with a single targeted intervention — reduce the background brightness of the Staff Schedule View (the no-login staff-facing page) to a near-white (#F8F8F8) rather than full white (#FFFFFF). This single change meaningfully reduces glare on OLED screens in low-light conditions without requiring a full dark mode token system. It is not dark mode. It is a reasonable accommodation that avoids the worst failure mode while deferring the full investment.

Dark mode trigger condition: When the product reaches 50 paying accounts and has confirmed retained engagement, allocate one design and one engineering sprint to a full dark mode token system. Do not ship a partial dark mode (some screens dark, some light) — the visual inconsistency is worse than a consistent light mode.

Key Micro-Interactions for the Publish and Confirm Flow

The Publish and Confirm flow — the moment a schedule is sent and delivery is confirmed — is the North Star activation event. It is also the moment of highest emotional stakes for Maria: she has just spent 15 minutes building the schedule, she is about to send it to 18 people, and she needs to trust that it worked. The micro-interactions in this flow are not decorative — they are trust signals.

Interaction 1: Publish Button State Transitions

The Publish button in the Schedule Header transitions through four states during the publish flow. Each state change is animated at 150ms ease-out — fast enough to feel responsive, slow enough to be readable.

State	Visual Treatment	Meaning
Inactive	Gray fill, muted label ("Publish Schedule"), disabled	No shifts added yet. Cannot be tapped.
Ready	Brand primary fill, active label ("Publish Schedule"), tappable	At least one shift exists. Ready to proceed.

Sending	Animated progress fill (left to right), label changes to "Sending…", non-tappable	Schedule is being transmitted to SMS provider.
Sent	Solid fill, checkmark icon replaces label text for 1.5 seconds, then label changes to "Sent — view delivery"	Transmission complete. Transitions to delivery status panel.

The checkmark moment (the 1.5-second pause on the checkmark state) is intentional. It is not a loading indicator. It is a moment of confirmation designed to give Maria a clear signal that the action completed before the UI transitions. A button that immediately moves to the next state without a perceptible "done" moment creates anxiety: did it actually send?

Interaction 2: Delivery Status List — Real-Time Population

After the schedule is sent, the Delivery Status panel renders as a list of all staff members with an animated status indicator per person. The list does not render all at once — each row populates as the delivery receipt is received from the SMS provider, creating a live-updating view rather than a static success screen.

Behavior:

- Rows that have not yet received a delivery confirmation show a pulsing gray dot ("Sent").
- Rows that receive a delivery confirmation animate from gray to

12 Technical Architecture

Technical Architecture

Overview

This section defines the technical structure of the product across three dimensions: what to build now, what to design for but not build yet, and what to deliberately defer until scale justifies the investment. The architecture decisions below are shaped by one constraint that overrides all others: the product must reach a paying pilot in 6 weeks, not 6 months. Every decision is evaluated against that constraint first.

The architecture is not neutral. It reflects the product's specific load profile — weekly scheduling bursts on Sunday evenings, real-time SMS delivery tracking in short windows, low concurrent user counts at launch — and the failure modes that matter most: an SMS confirmation that silently fails, a schedule that doesn't save before Maria closes the app, a callout coverage suggestion that never loads.

System Architecture Diagram

Key Technical Decisions with Justification

Framework and Hosting

- *Use Next.js as the full-stack framework deployed on Vercel.* Justification: API routes and the manager app share a deployment unit, eliminating a separate backend service to maintain during validation. Vercel's edge runtime handles the geographic distribution needed for SMS webhook latency without configuring infrastructure. The PWA capability of Next.js handles mobile-first requirements without a native app investment. The decision is revisable; the data model is not coupled to it.
- *Use Supabase as the primary database and auth provider.* Justification: Supabase provides PostgreSQL with row-level security (RLS) built in — the correct primitive for multi-tenant restaurant data isolation without building a tenant isolation layer from scratch. Auth, Realtime subscriptions, and the database ship as a single managed service, removing three infrastructure decisions from the critical path to first pilot. The tradeoff is vendor dependency; that dependency is acceptable before product-market fit and worth re-evaluating at 200+ paying accounts. (est.)

- *Implement row-level security policies before any paying customer is onboarded.* RLS is the enforcement mechanism that prevents one restaurant's data from being visible to another. It is not a post-launch audit item. Supabase RLS policies are written at the database level and enforced regardless of application logic errors — this is the correct security model for multi-tenant SaaS serving sensitive labor and compensation data.
- *Evaluate Supabase dedicated plan or migrate to self-hosted PostgreSQL on RDS when monthly active accounts exceed 500.* The Supabase free and pro tiers are appropriate for validation and early growth. At scale, connection pooling limits and shared infrastructure become constraints worth resolving. This decision is a BEFORE_SCALE concern, not a NOW concern.

Background Jobs

- *Use Trigger.dev for all background jobs from day one, not inline API route processing.* Justification: SMS dispatch, delivery receipt polling, and Claude API calls are all latency-variable operations that should not block the HTTP response cycle. An API route that inline-awaits an SMS send will timeout under load and, more importantly, will leave Maria staring at a spinner on Sunday night. Trigger.dev provides a job queue with retry logic, failure visibility, and execution logs accessible without setting up infrastructure. At pilot scale, the free tier is sufficient. The architecture decision (background jobs for all async work) is correct from the first deploy and avoids a painful refactor when moving from 5 pilots to 50 accounts.
- *Define retry policies for each job type before onboarding pilots.* SMS dispatch should retry 3 times with exponential backoff before marking delivery as failed. Claude API calls should retry once before falling back to a rule-based suggestion set (described below). Delivery receipt polling should retry on a 30-second interval for 10 minutes after send. These policies are not defaults — they must be explicitly configured and tested against Twilio sandbox before the first real schedule is sent.

AI Integration

- *Scope Claude API usage to three specific, bounded functions: coverage candidate ranking, shift conflict plain-language explanation, and schedule copy generation.* Claude is not the product. It is a utility called in the background by Trigger.dev jobs when a specific trigger condition is met. Coverage ranking is the most critical: when a shift is marked uncovered, the job passes structured staff availability data to Claude and requests a ranked list of candidates with a brief reason per candidate ("Jordan is available and worked this shift last week"). The response is parsed and stored, not streamed to the UI. Latency is acceptable because the response renders in a panel, not a blocking modal.
- *Build a rule-based fallback for every Claude-dependent feature.* If the Claude API is unavailable or the job fails after retries, coverage candidate ranking falls back to a deterministic algorithm: filter staff not scheduled that day, sort by hours worked that week (ascending), and surface the top 5. This fallback produces a less intelligent but fully functional result. The product must not degrade to an error state when the AI call fails — restaurant operators will not accept a product that tells them it cannot help because an external API is down.
- *Implement prompt versioning and output logging before scaling to 100+ accounts.* Every Claude API call should log the input parameters (not the raw prompt), the structured output, and whether the suggestion was accepted or dismissed. This data trains the rule-based fallback and provides the evidence base for

deciding whether the AI integration is delivering measurable value (time-to-cover improvement tracked via timetocover_minutes in the analytics schema).

- *Fine-tune or move to a smaller model for coverage ranking once training data is available.* Claude Haiku or a self-hosted smaller model will be meaningfully cheaper at scale for structured ranking tasks that do not require broad reasoning. This is not a NOW decision and should not be made before there is evidence the feature is used.

Data Flow: User Input ’Background Job ’Claude API ’DB ’Real-time UI

The following trace follows the callout coverage flow — the highest-stakes user interaction in the product — end to end.

Database Schema Overview

The schema is designed around three isolation boundaries: accounts (restaurants), schedules (weekly), and staff members. Every query that could leak cross-account data is blocked at the RLS policy level, not just at the application layer.

Core Entities and Relationships

Key schema decisions:

- *Store phone_e164 in E.164 format at write time, not at send time.* Validation happens on staff profile creation. A phone number that fails E.164 validation surfaces an inline error at input ("Please include the country code, e.g., +1 for US numbers") rather than silently failing when an SMS is dispatched at 8pm Sunday.
- *weekstartdate is enforced as a Sunday by a database CHECK constraint, not application logic.* Schedules with an ambiguous start date corrupt the weekly grid display and overtime calculations. One constraint prevents a class of bugs permanently.
- *analytics_events table is append-only with no updates or deletes permitted via RLS.* This table is the audit trail for kill condition evaluation. If a churn rate dispute arises at Month 9, the events table must be the authoritative record. Preventing application-level mutation is non-negotiable before the first paying customer.
- *Add a partitioning strategy to shifts and smsnotifications tables by accountid hash when accounts exceed 1,000.* At early scale, full table scans on these tables are acceptable. Partition strategy should be planned before migration complexity escalates.

API Design Principles

- *All API routes return consistent envelope shapes: { data, error, meta }.* Error objects include a code (machine-readable), a message (human-readable for developer logs), and a user_message (safe to surface in the UI). This consistency means the client error-handling layer is written once and applied everywhere.
- *No API route processes more than one primary side effect synchronously.* A route that sends an SMS, writes to the database, and fires an analytics event in a single async chain will fail in unpredictable ways when any one operation is slow. The pattern is: write to DB ’ enqueue job ’ return 202. The job handles everything else.
- *Implement idempotency keys on all POST routes that trigger SMS sends.* If Maria taps "Publish" twice because the button state transition is slow on a 3G connection, two schedule notifications should not be sent to 18 people. The client generates a UUID idempotency key per publish action; the server rejects duplicate submissions for the same key within a 60-second window.
- *Webhooks from Twilio must validate request signatures before processing.* Twilio signs all webhook requests with an HMAC-SHA1 signature. A webhook endpoint that processes unsigned requests can be triggered by any external party, producing phantom SMS events in the delivery status panel and corrupting the analytics record. Signature validation is a one-line implementation with the Twilio SDK; skipping it is not a reasonable shortcut.
- *Version the API at the route level (/api/v1/) before the first POS integration partner.* Phase 4 POS integrations (Toast, Square) require a stable API contract. Without versioning, breaking changes in internal routes force integration partners to update simultaneously or break silently.
- *Adopt OpenAPI spec generation when the API surface exceeds 20 routes.* At launch, a maintained README with request/response examples is sufficient. Generated documentation is a SOMEDAY concern.

Caching Strategy

The caching strategy at this stage is deliberately minimal. Premature caching introduces invalidation bugs in a product where data freshness is operationally critical — a cached schedule that shows Devon a shift as covered when it is actually uncovered is a product failure, not a performance optimization.

- *Cache session tokens in Redis (Upstash) with a 24-hour TTL.* Session validation happens on every API request. A database round-trip per request is unnecessary latency and Supabase connection overhead. Redis session cache reduces auth validation to a single Redis lookup.
- *Rate-limit SMS sends per account via Redis token bucket: 100 SMS per account per hour.* A misconfigured publish that sends duplicate notifications to 18 staff members 5 times is a Twilio bill and a user trust failure. Redis-based rate limiting with a clearly surfaced error ("Schedule already sent to this team — wait 1 hour to resend") prevents the failure mode without blocking legitimate use.

- *Cache the labor cost calculation for each published schedule for 5 minutes.* The labor report queries shifts × hourly rates and aggregates by role. At 30 staff members per schedule, this query is fast. At 500 accounts opening the labor report simultaneously on a Monday morning, it is not. A 5-minute stale-while-revalidate cache on the read path costs nothing and absorbs the concurrent read pattern.
- *Do not cache schedule draft state.* A draft schedule must always reflect the latest DB state. The auto-save pattern in the schedule builder (described in the Product Design section) writes every shift change to the DB immediately. The client renders from DB state, not a local cache. Caching draft schedules introduces merge conflict risk during concurrent editing (two browser tabs, a phone and a tablet) that is not worth solving at launch.
- *Evaluate CDN caching for the Staff Schedule View when 10,000+ weekly schedule views are plausible.* The no-login staff schedule view is server-rendered HTML with tokenized URLs. At early scale, no caching is needed. At significant staff-side traffic, a short-TTL CDN cache (60 seconds) reduces origin load substantially. This is a BEFORE_SCALE decision triggered by observed traffic, not a launch requirement.

Scalability Bottlenecks and Mitigations

The following are the five most likely technical bottlenecks, ordered by the likelihood of encountering them in the first 18 months.

1. Sunday Evening SMS Burst

Profile: All schedules for the week are published between 6pm and 10pm Sunday. At 200 accounts with 20 staff each, this is 4,000 SMS messages in a 4-hour window. (est.)

Bottleneck: Twilio has per-account messaging rate limits. A synchronous SMS dispatch in a single job will queue messages sequentially and produce 10–20 minute delivery delays for accounts that publish late.

Mitigation:

- Twilio A2P 10DLC registration (required for US business SMS). Unregistered numbers have significantly lower throughput caps. This is a compliance and throughput requirement before any paying pilots, not

13 Security & Compliance

Security & Compliance

Overview

This section defines the security and compliance posture of the product across authentication, data handling, API hardening, and regulatory obligations. Two constraints shape every decision here: the product handles real employee data — phone numbers, hourly rates, schedules — from the first pilot, and it does so in a multi-tenant environment where one restaurant's data must be structurally invisible to every other restaurant.

Security decisions are not distributed evenly across the product lifecycle. Some must be correct before the first conversation with a paying customer. Others only become blocking constraints when the account base reaches a size where a breach has reputational and legal consequence at scale. The time-horizon tags in this section reflect that reality. A founder who treats a SOC 2 Type II audit as a launch requirement will not reach the validation gates described in the Validation Roadmap. A founder who ignores data isolation until Month 9 will face a customer data incident at exactly the moment the product is trying to prove retention.

The organizing principle here is the same as in the Technical Architecture section: build the irreversible security decisions correctly from day one, defer the expensive certification processes until there is evidence of a product worth certifying.

Authentication Security

The product uses Supabase Auth as the primary authentication provider, as established in the Technical Architecture section. The following decisions govern how sessions are established, maintained, and terminated.

Session Architecture

- *Use Supabase Auth's JWT-based session model with short-lived access tokens (1-hour TTL) and longer-lived refresh tokens (7-day TTL by default, configurable to 30 days).* The access token is never stored in localStorage. It is stored in an httpOnly, Secure, SameSite=Strict cookie managed by Supabase's client library. An access token in localStorage is readable by any JavaScript running on the page — including third-party analytics scripts. The cookie model eliminates that attack surface.
- *Invalidate all active sessions on password change.* Supabase Auth supports global sign-out via the admin API. A manager who suspects their account is compromised must be able to end all sessions from

a single action in the Settings screen. The "Sign out of all devices" option is not a power feature — it is a basic security expectation for any product handling employee data.

- *Implement session activity logging.* Every successful authentication, failed login attempt, and session refresh is written to a sessionevents table with { accountid, userid, eventtype, ipaddresshash, useragenthash, occurred_at }. IP addresses and user agents are stored as one-way hashes, not raw values — sufficient for anomaly detection, insufficient to constitute surveillance data. This log is the evidence base for investigating a suspected account compromise in a pilot restaurant.
- *Enforce a maximum of 5 failed login attempts per email address within a 15-minute window before triggering a temporary lockout (15 minutes).* The lockout state sends an email to the registered address with an unlock link and a notice that repeated failures occurred. This is a brute force mitigation, not a CAPTCHA — CAPTCHAs create friction on mobile devices that are already the most common login surface for this product's users.
- *Add optional TOTP-based two-factor authentication for the Manager tier ($89/month).* Multi-location managers (Devon's persona) are controlling labor data and staff contact information for 2–3 locations and 15–35 employees. The risk surface justifies 2FA at that tier. Forcing 2FA on the Operator tier creates onboarding friction that will materially hurt activation rates for Maria, who is often setting up her account on a phone, in a restaurant, with a 5-minute attention window.

OAuth and Third-Party Identity

- *Restrict OAuth provider login (Google, Apple) to account creation only, not post-creation linking.* A manager who creates an account with email/password and then attempts to link a Google account introduces session state complexity that is not justified at this scale. OAuth at creation is simple — one identity provider, one account. Post-creation linking requires deduplication logic and creates an account recovery surface that is difficult to harden correctly.
- *Validate OAuth redirect_uri against a static allowlist registered with each OAuth provider before launch.* An open redirect in the OAuth callback flow is a known attack vector that allows session token theft via crafted URLs. The allowlist contains exactly: localhost:3000 for development, the Vercel preview domain, and the production domain. No wildcard patterns.

Staff Schedule View (No-Login Access)

The no-login staff schedule view — described in the Product Design section — creates a distinct authentication problem: it must be accessible without credentials while remaining inaccessible to anyone who does not have the specific link.

- *Generate tokenized schedule URLs using a cryptographically random 32-byte token (256 bits), not sequential IDs or UUIDs.* A URL like /schedule/view/xK7mP2... (random token) is functionally unguessable. A URL like /schedule/view/1042 or /schedule/view/[uuid] is not — UUIDs are not secret, and sequential IDs are trivially enumerable. The token is stored in the schedules table alongside the schedule_id and is regenerated each time the schedule is republished.
- *Set a 7-day expiry on staff schedule view tokens.* A schedule link that remains valid indefinitely allows access to historical staffing patterns, employee phone numbers (via the "View full schedule" toggle), and labor coverage data from an arbitrarily old date. Seven days covers the relevant work week plus two days

for late-access scenarios (a staff member checking their shifts on Tuesday after a Sunday publish). Expired tokens return a 410 Gone response with a plain-language message: "This schedule link has expired. Ask your manager to resend the current schedule."

- *Rate-limit staff schedule view requests to 60 requests per token per hour.* An unrestricted tokenized endpoint that accepts 60 requests per second is a data scraping surface. The rate limit is token-level, not IP-level — a shared restaurant WiFi network may serve multiple staff members simultaneously, and an IP-level limit would incorrectly block legitimate access.

Data Classification

The product handles three categories of data with meaningfully different sensitivity levels. Every storage and transmission decision should be evaluated against the classification of the data being handled.

Classification Tiers

Tier 1 — Personally Identifiable Information (PII): Staff member names, phone numbers in E.164 format, and (where collected) email addresses. This data is subject to GDPR, CCPA, and standard data protection expectations. It requires encryption at rest, limited retention, and a documented deletion path.

Tier 2 — Sensitive Business Data: Hourly rates, scheduled hours, overtime patterns, and labor cost totals. This is compensation data. In many states and jurisdictions, compensation data is subject to additional legal protections. It should not appear in logs, error messages, or over-the-shoulder visible default views — the Product Design section's decision to require a tap to reveal per-employee cost data on the labor report is a security decision, not just a UX one.

Tier 3 — Operational Data: Schedule structure, shift times, role assignments, and coverage status. Lower sensitivity, but still scoped to a specific restaurant account. Cross-account visibility of this data is a breach even if no PII is exposed — a restaurant operator can infer competitor staffing patterns from another restaurant's schedule.

Storage Decisions by Classification

- *All Tier 1 and Tier 2 data is encrypted at rest via Supabase's AES-256 encryption, which is applied at the volume level by default.* This is not sufficient on its own — encryption at rest protects against physical storage theft, not application-layer data access. Row-level security (described in the Technical Architecture section) is the enforcement mechanism for access control. Both are required.
- *Phone numbers (Tier 1 PII) are stored in E.164 format and never logged in plaintext in application logs, error messages, or analytics event properties.* The smsnotifications table stores the twiliomessage_sid as the reference to the outbound SMS — not the destination phone number. A log entry that contains a phone number is a compliance incident if the log is stored in a third-party service (Vercel log drains, PostHog, etc.) without appropriate data processing agreements.

- *Hourly rates (Tier 2) are stored as numeric(10,2) in the staff_members table and are excluded from the default SELECT return of any API route.** Every route that needs hourly rate data must explicitly name the field in its query. This is enforced by a code review standard, not a database constraint — but the standard must be documented before the first engineer other than the founder writes a database query.
- *Define a data retention policy before onboarding pilots.* Archived staff members' records are retained for 24 months from the archive date to support potential labor dispute or payroll audit use cases, then deleted. Deleted accounts' data is purged within 30 days of account termination, with the exception of any data subject to a legal hold. The retention policy is documented in the Terms of Service and referenced in the privacy policy before the first paying customer accepts those terms.
- *Implement a datadeletionrequests table and a manual processing workflow before onboarding pilots.* GDPR Article 17 and CCPA both provide individuals (including employees who received schedule notifications) with a right to deletion of their personal data. The workflow is manual at this stage — a staff member contacts the restaurant owner, who submits a deletion request through the Settings screen, which creates a record in datadeletionrequests and triggers a 30-day deletion job. The job is Trigger.dev-based and deletes the staff member's PII fields (name, phonee164) while retaining the anonymized scheduling record (staffmember_id with PII nulled) for labor history integrity.

API Security

Rate Limiting

The Technical Architecture section establishes Redis-based rate limiting for SMS sends. The following extends rate limiting to the full API surface.

- *Implement per-IP rate limiting on all unauthenticated API routes at 20 requests per minute.* Unauthenticated routes include /api/auth/login, /api/auth/signup, /api/webhooks/twilio/reply, and the staff schedule view endpoint. These routes are the highest-value targets for automated abuse. The Twilio webhook endpoint requires special handling — it must pass Twilio's signature validation before the rate limit check, not after, to avoid processing unsigned requests that consume rate limit tokens.
- *Implement per-account rate limiting on authenticated API routes at 300 requests per 15-minute window.* A legitimate restaurant manager building a schedule will not exceed 300 API calls in 15 minutes. A compromised account or a misconfigured integration will. This limit is surfaced to the client as an HTTP 429 Too Many Requests response with a Retry-After header, not a silent failure.
- *Set per-route rate limits for high-cost operations independent of the global authenticated limit:*
  - æSMS schedule publish: 3 sends per schedule per hour (idempotency keys enforce uniqueness; this is a secondary hard stop)
  - æCoverage offer SMS: 20 per account per hour
  - æClaude API-backed routes: 10 requests per account per 5 minutes (Claude API calls have both latency and cost implications)

Input Validation

- *Validate all input at the API route layer using a schema validation library (Zod is appropriate for a Next.js TypeScript codebase).* Every POST and PATCH request defines a Zod schema that specifies field types, lengths, formats, and allowed values. A request that fails schema validation returns a 400 Bad Request with a structured error object — not a 500 that leaks stack trace information. Phone numbers are validated against E.164 format at input; weekstartdate is validated as a Sunday (matching the database CHECK constraint) at input; schedule week ranges are validated against a ±52-week window from the current date to prevent UI corruption from malformed dates.
- *Parameterize all database queries.* Supabase's JavaScript client uses parameterized queries by default. The risk is not in the client library — it is in any raw SQL executed via supabase.rpc() for custom database functions. Any RPC function that accepts user-controlled input must use $1 parameter binding, not string interpolation. This is enforced by a code review standard before the first deploy to production.
- *Sanitize all text fields that render in the Staff Schedule View.* The restaurant name, shift notes, and manager contact name are user-supplied strings that render as HTML in the no-login staff schedule view. A restaurant that enters <script>alert(1)</script> as their restaurant name should not execute JavaScript in a staff member's browser. Output encoding is applied at the template layer (Next.js escapes JSX output by default; this standard explicitly prohibits dangerouslySetInnerHTML throughout the product).
- *Implement request body size limits on all API routes.* The default Next.js body limit is 1MB. A CSV roster upload is the largest legitimate payload in the product — a 30-person roster CSV is under 10KB. Set the body limit to 512KB for roster upload routes and 16KB for all other routes. An unexpectedly large request body is an attack surface and a billing concern if it triggers compute beyond the expected profile.

CSRF Protection

- *Rely on the SameSite=Strict cookie attribute as the primary CSRF mitigation, not a CSRF token.* A SameSite=Strict session cookie is not sent with cross-origin requests initiated by third-party pages. This eliminates the primary CSRF attack vector for the authentication flow without requiring a CSRF token implementation that must be maintained across all form submissions. This mitigation is effective for browser-based attacks; it does not apply to API routes called by server-side clients (POS integrations in Phase 4+), which authenticate via API keys rather than session cookies and require a separate access control model.
- *Add Origin header validation on all state-changing API routes (POST, PATCH, DELETE) before Phase 4 integrations.* The Origin header is checked against a static allowlist containing the production domain and (in development) localhost:3000. A request with a missing or non-allowlisted Origin is rejected with a 403 Forbidden. This is a defense-in-depth measure that complements SameSite=Strict rather than replacing it.

GDPR and CCPA Requirements

This product operates in a jurisdiction-ambiguous context: the operator (Maria) is in the United States, but CCPA applies if the restaurant has California customers, and GDPR applies if any staff member (the data subject whose personal data is processed) is in the EU. The practical answer for a product at this stage is to

build for both standards simultaneously, because the requirements overlap substantially and the engineering cost of a GDPR-compliant system is not meaningfully higher than a CCPA-compliant one.

What Makes This Product a Data Processor

When Maria adds a staff member's phone number to the roster, the product becomes a *data processor acting on behalf of Maria, who is the data controller. The staff member whose number is stored is the data subject*. This structure creates specific obligations.

- *Include a Data Processing Agreement (DPA) in the Terms of Service that is presented to every paying customer before their first schedule is published.* The DPA documents: what data is processed, for what purpose, on whose behalf, for how long, and under what security measures. A DPA is required under GDPR for any controller-processor relationship. For US operators, it signals data stewardship seriousness and is increasingly expected in B2B SaaS procurement. A lawyer-reviewed DPA template is available at reasonable cost from services such as Termly or via direct counsel — this is not a founder-drafted document.
- *Staff members who receive SMS schedule notifications are data subjects whose personal data (phone number) is processed.* They are not account users and cannot be expected to have accepted the product's terms. The product's privacy policy must be accessible at the foot of every staff schedule view page (the no-login link) and must include a contact mechanism for data subject access requests and deletion requests. The privacy policy is not a legal formality — it is the mechanism by which a staff member can discover what data the restaurant's scheduling tool holds about them. [Source: GDPR Article 13; CCPA Section 1798.100]
- *Build the datadeletionrequests workflow described in the Data Classification section before onboarding any pilots.* A pilot customer who receives a deletion request from a former employee and cannot fulfill it within 30 days (GDPR) or 45 days (CCPA) is in breach of their legal obligations, and the product is the tool that failed them. The workflow does not need to be automated at launch — a manual trigger in the Settings screen that creates a database record and alerts the product team is sufficient.
- *Write and publish a privacy policy that explicitly covers:* the categories of personal data collected (names, phone numbers), the legal basis for processing (legitimate interest on behalf of the controller, or contract), data retention periods, third-party processors (Supabase, Twilio, Vercel, PostHog), cross-border data transfer mechanisms (Standard Contractual Clauses with EU-based staff, if applicable), and the data subject rights (access, rectification, erasure, portability, objection). The policy must be reachable from the login screen, the staff schedule view, and the Settings billing page — not only from the footer of the marketing site.
- *Document every third-party data processor and ensure a DPA or equivalent data processing addendum is in place with each.* The relevant processors are:
  - æSupabase: DPA available in their enterprise terms; sign it explicitly.
  - æTwilio: Data Protection Addendum available in Twilio console; accept it.
  - æVercel: DPA available on request; request and sign it.
  - æPostHog: EU-hosted instance or DPA for US-hosted; document the choice.

æAnthropic (Claude API): Review Anthropic's data processing terms for business API use before sending staff-related data to the API. The coverage ranking job passes structured availability data to Claude — ensure Anthropic's terms permit processing of this type of data for this purpose and do not use it for model training without explicit consent.

- *Implement a self-service data export function (data portability, GDPR Article 20) from the Settings screen.* An account owner must be able to download all data held about their account in a machine-readable format (JSON or CSV). At launch, this is a manual request handled by the product team within 30 days. At 50+ accounts, the manual process becomes unscalable and a dedicated export function is required.

SOC 2 Readiness

SOC 2 Type II certification is not appropriate for a product that has not reached the Kill Condition K3 gate (5 retained paying customers at Month 6). The audit requires 6–12 months of documented compliance evidence and costs between $15,000 and $50,000 in auditor fees at minimum. (directional) Pursuing it before product-market fit is confirmed is a resource misallocation. However, the behaviors that lead to SOC 2 readiness are not expensive to begin — and starting them from day one eliminates the retroactive remediation work that makes SOC 2 audits painful for companies that deferred all compliance activity.

SOC 2 is built on five Trust Service Criteria: Security, Availability, Processing Integrity, Confidentiality, and Privacy. This product needs Security and Confidentiality from day one and should build toward Availability (uptime commitments) and Privacy (regulatory alignment) by the time the first enterprise or multi-location customer asks for it.

What to Do from Day One

- *Maintain a private architecture decision log (not public documentation).* Every significant technical decision — why Supabase was chosen, why the schema was designed as it was, why a specific rate limit value was set — is logged with a date, the decision, and the rationale. SOC 2 auditors verify that decisions are intentional and documented. A founder who cannot explain a 6-month-old decision from a log entry fails the process integrity criterion.
- *Define and document access control principles before writing a single line of production code.* The principle is: minimum necessary access. Engineers have read access to production logs and anonymized analytics. Write access to the production database requires explicit justification and a time-bound approval. No persistent root database access in production. These principles are documented in a private security policy document, not in the codebase.
- *Enable Supabase's point-in-time recovery (PITR) on the production database before the first real data is written.* PITR is the Availability backstop — it allows the database to be restored to any point in the last 7 days (or longer on higher Supabase tiers). A database that cannot be recovered to the state before a destructive operation is not a product that can make uptime commitments to paying customers.

- *Implement a formal incident response process before the first paying customer.* The process does not need to be elaborate. It needs to exist and be documented: (1) How is an incident detected? (PostHog alerts, Vercel error monitoring, user report.) (2) Who is notified? (Founder, then affected customer if data is involved.) (3) What is the containment response? (Disable affected account, revoke compromised tokens.) (4) What is the customer communication timeline? (Within 72 hours for data breaches affecting personal data — GDPR Article 33 requirement.) A two-page document that is actually followed is worth more than a 40-page policy that is not.
- *Enable audit logging for all admin-level database operations.* Supabase's pg_audit extension logs DDL and DML statements executed by admin roles. This log is the primary evidence artifact for SOC 2 Security criterion: it demonstrates that privileged access is monitored and attributable.
- *Conduct a dependency audit and remove unused packages before the first paying pilot.* Every npm package in the production dependency tree is a potential vulnerability surface. A monthly npm audit run with a documented triage process (critical vulnerabilities remediated within 7 days, high within 30 days) is the minimum viable dependency security practice. The triage process is documented and dated so it is auditable.
- *Create a vulnerability disclosure policy and publish it at /security.txt and a security page on the marketing site.* Penetration testers and security researchers who find a bug in the product need a documented way to report it responsibly. A disclosure policy that provides a contact email, a commitment to acknowledge within 48 hours, and a safe harbor statement for good-faith security research is the baseline expectation for any commercial SaaS.
- *Engage a SOC 2 readiness firm to conduct a gap assessment when the product has 50+ paying accounts and is actively in enterprise sales conversations.* The gap assessment (typically $5,000–$15,000) (directional) identifies what formal controls need to be implemented before the Type II audit observation period begins. Starting the assessment at 50 accounts gives 6–9 months to remediate gaps before a prospect formally requests a SOC 2 report.
- *Implement automated vulnerability scanning (Snyk or equivalent) as part of the CI/CD pipeline.* Manual dependency audits are insufficient at scale. Automated scanning catches newly disclosed vulnerabilities in existing dependencies without requiring a manual trigger.
- *Pursue SOC 2 Type II certification when a Fortune 500 restaurant group or a national chain procurement process formally requires it.* This is the correct trigger condition — not a revenue milestone, not a headcount milestone. SOC 2 is a sales enablement tool for a specific buyer profile. If the product's target customer (independent restaurant operators) never asks for it, the certification cost is never justified.

Stripe PCI Compliance

The product uses Stripe for subscription billing. PCI DSS compliance scope is entirely determined by how the payment form is implemented. The following decisions keep the product in the lowest-compliance tier (SAQ A) by ensuring that cardholder data never touches the product's servers.

- *Use Stripe Checkout or Stripe Elements (not a custom card form) for all payment collection.* Stripe Checkout redirects the user to a Stripe-hosted page. Stripe Elements renders Stripe-hosted iframes within the product UI. In both cases, the card number, CVC, and expiration date are entered directly into Stripe's infrastructure — they are never transmitted to or stored by the product's API. This decision is irreversible if custom card handling is later attempted; the PCI scope change is significant and expensive.
- *Never log, store, or transmit raw card data in any form.* Webhook payloads from Stripe contain a payment_method object with the last 4 digits of the card

14 Accessibility & CRO

Conversion & Accessibility

Overview

Two principles govern this section: first, accessibility is not a post-launch audit item — it is a structural requirement that, if deferred, becomes a rework cost that scales with codebase complexity. Second, conversion optimization at this stage is not about marginal A/B testing on button colors — it is about removing the specific friction points that cause the primary ICP to abandon the product before experiencing its core value. Both disciplines are addressed below with the specificity of a product that has a known user (Maria, scheduling on Sunday night from her phone) and a known first-session success condition (a complete schedule published, with SMS confirmation received by staff).

The B2B SaaS market context reinforces the urgency of both: as the market grows toward USD 4,441.49 billion by 2034 [source: mordorintelligence.com/industry-reports/b2b-saas-market], competitive density increases and tolerance for friction decreases. Products that convert and retain at launch have a structural compounding advantage over products that defer UX investment until after they have customers to lose.

WCAG 2.1 AA Requirements for Key Flows

WCAG 2.1 AA compliance is the minimum legal and ethical standard for a product serving a broad small-business operator audience. The independent restaurant segment includes operators across a wide age range (noted as 38–52 for the primary ICP), with varying levels of digital fluency, and staff (Priya persona) who access schedule view links on personal devices with no account setup. Accessibility failures in the staff-facing schedule view are particularly high-stakes: a staff member who cannot read the schedule due to low-contrast text or inaccessible mobile layout creates the exact outcome the product was designed to prevent — the manager receiving a "I couldn't see the schedule" text at 10pm.

The four flows below are prioritized by the frequency with which they will be encountered and the severity of failure if they are inaccessible.

Flow 1: Authentication (Sign-Up and Sign-In)

Why this flow matters for accessibility: Maria is creating an account for the first time, potentially on a phone, in a noisy environment (the restaurant), after a shift. Devon may be setting up the product on a tablet shared across manager functions. Priya is receiving a schedule link and may be creating a staff-view account for the first time with limited context.

WCAG 2.1 AA requirements:

Requirement	Criterion	Implementation
Color contrast	1.4.3 — Minimum contrast ratio 4.5:1 for all text (body, labels, placeholders, error messages)	Test all form label and error text against the background color. Placeholder text (typically gray-on-white) frequently fails at default values — do not rely on placeholder text to convey required field intent
Focus visibility	2.4.7 — Keyboard focus indicator visible on all interactive elements	All form fields, buttons, and links must display a visible focus ring. Do not remove outline in CSS without providing a custom focus indicator of equal or greater visibility
Error identification	3.3.1 — Error messages identify the specific field and describe the error in text (not color alone)	"This field is required" is not sufficient. "Email address is required" or "Enter a valid email address (e.g., name@restaurant.com)" meets the standard
Label association	1.3.1 — All form inputs programmatically associated with their visible label via <label for> or aria-labelledby	Do not use placeholder text as the sole label — it disappears on input and provides no persistent label for screen readers
Touch target size	2.5.5 — Interactive controls at least 44×44 CSS pixels	All sign-up form buttons, input fields, and the "show/hide password" toggle must meet minimum tap target size on mobile

Autocomplete support	1.3.5 — Autocomplete attributes on personal data fields	autocomplete="email", autocomplete="current-password", autocomplete="new-password" on all relevant fields. This is also a conversion requirement: autofill reduces form abandonment
Session timeout warning	2.2.1 — Users warned before session times out and given option to extend	If a sign-up form session can expire, provide a warning with a "stay signed in" option at least 20 seconds before expiry

Known failure mode to test: Social sign-in buttons ("Continue with Google") that render as images without accessible text. Every social login button must have an aria-label that matches the visible button label.

Flow 2: Scheduling Dashboard (Schedule Build and Edit)

Why this flow matters for accessibility: The schedule builder is the core product experience. If it is not navigable by keyboard, readable at standard zoom levels, or operable on a mid-range Android device, Maria's first session ends without her building a schedule — which means she does not convert from free to paid, and the product fails its primary conversion condition before any CRO intervention is possible.

WCAG 2.1 AA requirements:

Requirement	Criterion	Implementation
Drag-and-drop accessibility	2.1.1 — All functionality available via keyboard	Shift assignment via drag-and-drop must have an equivalent keyboard interaction (e.g., select shift with Enter, navigate to slot with arrow keys, confirm assignment with Enter). Drag-only interfaces are an accessibility blocker for keyboard-only and motor-impaired users
Color not sole differentiator	1.4.1 — Color is not the only means of conveying status	If shifts for FOH are shown in blue and BOH in green, add a text label or icon pattern in addition to color. A color-blind user (approximately 8% of male users) must be able to differentiate shift types without relying on hue alone

Zoom and reflow	1.4.10 — Content reflowing at 400% zoom without horizontal scrolling	The schedule grid must reflow to a single-column view when zoomed to 400% on a 1280px-wide viewport. A fixed-width schedule table that requires horizontal scrolling at 400% zoom fails this criterion
Status messages	4.1.3 — Status updates (e.g., "Schedule saved," "Staff member added") conveyed via role="status" or aria-live	Auto-save confirmation messages and inline validation feedback must be announced to screen readers without requiring focus to move to the message
Sufficient text size	1.4.4 — Text resizable to 200% without loss of content or functionality	Test the dashboard at 200% browser text size. Shift labels, staff names, and hour totals must remain readable and not overflow their containers
Data table structure	1.3.1 — Schedule grid conveyed as a data table if rendered as one	If the weekly schedule is rendered as an HTML table, use <th scope="col"> for day headers and <th scope="row"> for staff name headers. Screen readers announce table headers per cell, which is the only way the schedule is navigable non-visually
Motion and animation	2.3.3 — Animations can be disabled via prefers-reduced-motion	Any drag-and-drop animation, notification toast animation, or page transition must respect the OS-level reduced motion preference

Known failure mode to test: Shift time entry fields that accept only mouse-based time picker interactions. All time pickers must accept keyboard input (typed hours and minutes) as an alternative to the picker widget.

Flow 3: Staff Schedule View (Read-Only Link, No Login Required)

Why this flow matters for accessibility: This is the most frequently accessed page in the entire product — every staff member at every pilot restaurant accesses it each week. It is viewed on personal phones, often on low-end Android devices, sometimes in low-light conditions (a dark restaurant, a dim apartment). It requires no account, no login, and no product knowledge. Accessibility failures here are brand failures: if a staff member cannot read their schedule, the manager receives a text asking for the schedule, which is the behavior the product was designed to eliminate.

WCAG 2.1 AA requirements:

Requirement	Criterion	Implementation
Contrast on mobile	1.4.3 — 4.5:1 contrast ratio in all lighting conditions	Test on a real device in outdoor/bright light. Many mobile designs that pass contrast checks on desktop fail perceptually on phones with low brightness or in bright environments. A minimum ratio of 5.5:1 provides margin
Readable without CSS	1.3.1 — Content structure communicated via semantic HTML, not visual layout alone	The schedule view must display the correct shift information in a logical order when CSS is disabled. Use <time> elements for shift hours, and order the DOM to match the visual reading order
Link purpose	2.4.4 — All links describe their destination or purpose	If the schedule view includes a link "Download PDF" or "View previous week," the link text must describe the action. Avoid "Click here" or "See more" without context
Tap targets	2.5.5 — All interactive elements (week navigation arrows, download buttons) at minimum 44×44 px	Week navigation controls on mobile are particularly failure-prone — left/right arrows rendered as small icons frequently fail tap target requirements
No time-sensitive content	2.2.1 — If any content auto-refreshes or expires, user is warned	If the schedule view URL expires after a period, display a clear message with instructions to request a new link from their manager — not an unmarked 404 error

Flow 4: Export (Schedule PDF, Labor Report)

Why this flow matters for accessibility: Devon uses labor cost reports to justify the product's cost to his owner. Maria prints the schedule for the back-office board. Exported documents that are not accessible are non-compliant when used in any workplace context — and a PDF that screen reader software cannot parse is a WCAG failure that extends beyond the web product itself.

WCAG 2.1 AA requirements:

Requirement	Criterion	Implementation
Tagged PDFs	WCAG 2.1 maps to PDF/UA for exported documents — all PDFs must be tagged with document structure (headings, table headers, reading order)	Do not generate PDFs as image exports or untagged print-to-PDF outputs. Use a PDF library that produces tagged output (e.g., PDFKit with tag support, WeasyPrint with proper semantic HTML input)
Download trigger accessibility	2.1.1 — Export button is keyboard-operable	The export button must be reachable and activatable via keyboard. If the export is triggered via a custom styled , replace with <button>
Download confirmation	4.1.3 — Confirmation that download initiated conveyed to screen readers	"Your schedule has been downloaded" should be announced via aria-live="polite" when the download begins, not only shown visually
Print stylesheet	1.4.10 — Print view does not lose content	If schedule printing is available via browser print rather than PDF export, implement a @media print stylesheet that removes navigation, adjusts layout to a single column, and ensures all shift data is visible within page margins
CSV export for labor data	No WCAG criterion — operational requirement	Offer CSV export alongside PDF for labor cost data. CSV is screen-reader-navigable via Excel or Numbers, and is the format Devon's owner is most likely to request for budget review

Top 5 CRO Opportunities

Each opportunity below is paired with the specific persona affected, the current failure mode, and the intervention. These are ordered by estimated conversion impact, not implementation complexity.

CRO-

15 Analytics & Tracking

Analytics & Tracking

Overview

Analytics at this stage has one job: tell you whether the product is working before the money runs out. The kill conditions established in the validation roadmap are only enforceable if the underlying data exists when each gate is reached. This section defines what to measure, how to measure it, and what to watch so that decisions at Month 1, 3, 6, and 9 are made against logged events — not reconstructed from memory.

Every metric below maps to a specific business question. If a metric does not answer a question you will act on, it is not in this document.

North Star Metric

Schedules Published with e1 Staff SMS Confirmation Received — Weekly Active Accounts

Call this *SPSC-WAA* internally, or simply "confirmed schedules per week."

This is the number of paying accounts that published at least one schedule in the past 7 days and had at least one staff member receive an SMS confirmation of that schedule.

Why this metric and not another

Candidate Metric	Why It Fails as North Star
MRR	Lags behavior by billing cycle. A churning customer still shows MRR until they cancel.
Accounts Created	Measures top-of-funnel noise. Free-tier signups who never build a schedule inflate the number.
Schedules Published	Does not confirm that the product solved the communication problem. A schedule built but not received is a spreadsheet with extra steps.

DAU / MAU	Restaurant scheduling is weekly, not daily. A DAU ratio optimized for daily apps will produce misleading signals for a weekly-cadence product.
Staff Logins	Staff accessing the schedule is downstream of the manager publishing it. Optimizing for staff logins before optimizing for manager behavior inverts the dependency.

SPSC-WAA captures the moment of value delivery — not account creation, not feature usage, not payment. Maria published a schedule. Her team got a text. That is the product working. Every other metric in this document is either a leading indicator pointing toward that outcome or a lagging indicator confirming it compounded into revenue and retention.

North Star target by phase

Phase	Target	Rationale
End of Phase 3 (Month 3)	5 accounts posting confirmed schedules in at least 2 consecutive weeks	Proves the core loop works for paying pilots, not just one-time demo sessions
End of Phase 4 (Month 6)	20 accounts	Minimum signal for product-market fit at the independent restaurant scale
Month 12	100+ accounts	Aligns with the 100–300 paying customer Year 1 target of ~$323K ARR established in the Product Brief (est.)

Supporting Metrics

Supporting metrics are organized in four groups: acquisition health, activation quality, retention signals, and revenue integrity. Each maps to a kill condition or a decision gate.

Acquisition Health

Metric	Definition	Review Cadence	Kill Condition Link
Signups per week (by channel)	New accounts created, segmented by source (referral, cold outreach, POS partner, organic)	Weekly	K2: if zero paid conversions from 20+ contacts
Free-to-paid conversion rate	% of Starter accounts that upgrade to Operator within 30 days of first schedule publish	Monthly	K3: if < 5 retained paying customers at Month 6
Channel CAC	Total channel spend ÷ paying customers acquired from that channel	Monthly	K5: if CAC exceeds 12 months of subscription revenue
Signup-to-first-session rate	% of new accounts that return to the product within 7 days of signup	Weekly	Diagnostic: high signups + low return rate = onboarding problem, not demand problem

Activation Quality

Metric	Definition	Review Cadence	Kill Condition Link
Time-to-first-published-schedule	Minutes from account creation to first schedule published, per session	Per cohort	Core product bet: if median > 20 minutes, the "faster than a spreadsheet" claim fails
Time-to-first-SMS-confirmed	Minutes from first schedule publish to first staff SMS confirmation received	Per cohort	Activation event. If this does not happen in first session, Maria re-opens her spreadsheet
Onboarding completion rate	% of new accounts that complete all three onboarding steps: roster upload, first schedule built, first schedule published	Weekly	If < 60%, onboarding is leaking before activation
Staff invite acceptance rate	% of invited staff members who accept their invite and access the schedule link	Per cohort	If < 50%, the staff communication loop is broken — the product is not replacing the group text

Retention Signals

Metric	Definition	Review Cadence	Kill Condition Link
Weekly schedule publish rate	% of paying accounts that published e1 schedule in the past 7 days	Weekly	Primary retention health check. A restaurant not publishing is a restaurant about to churn
30-day retention	% of new paying accounts still active (published e1 schedule) 30 days after first payment	Monthly	K3: retention failure at 30 days predicts churn before the 60-day K3 gate
60-day retention	% of new paying accounts still active 60 days after first payment	Monthly	K3 direct: < 5 retained paying customers after 60 days triggers kill evaluation
Monthly churn rate	Paying accounts cancelled ÷ total paying accounts at start of month	Monthly	K4: > 10% for two consecutive months triggers kill evaluation

Feature engagement depth	% of Operator accounts using e3 features (schedule publish, overtime alert, callout suggestion, time-off request) in a given month	Monthly	Accounts using only one feature are at highest churn risk. Engagement breadth predicts retention.

Revenue Integrity

Metric	Definition	Review Cadence	Kill Condition Link
MRR	Sum of all active subscription revenue in the current month	Monthly	Baseline health signal
ARPU	MRR ÷ total paying accounts	Monthly	Target: $55/month blended. Drift below $49 indicates Starter accounts not converting to Operator (est.)
LTV (rolling)	ARPU ÷ monthly churn rate	Monthly	At 7% churn: $786. At 10% churn: $550. Below $385 at current churn rate: unit economics non-viable (est.)
LTV:CAC ratio	Rolling LTV ÷ blended CAC by channel	Monthly	Target e 2.6:1 base case. Below 1.5:1 requires immediate CAC reduction or pricing intervention (est.)
MRR expansion rate	Net new MRR from upgrades (Operator ' Manager) as % of total MRR	Monthly	If zero Manager tier conversions by Month 6, multi-location positioning is not landing with Devon persona

Event Tracking Schema

The following 18 events form the minimum viable tracking schema for the first 90 days. Every event includes a name in snake_case (compatible with Mixpanel, PostHog, and Segment), the properties to capture, and the trigger condition.

Authentication Events

Event Name	Properties	Trigger
account_created	`{ source: string, plan_tier: "starter"\	"operator"\
session_started	{ accountid: string, plantier: string, dayssincecreated: number, device_type: string }	User authenticates and lands on the dashboard
plan_upgraded	`{ fromtier: string, totier: string, triggersurface: "on-boardingprompt"\	"paywall_hit"\
plan_cancelled	`{ tieratcancellation: string, monthsactive: number, can-cellationreason: string\	null, lastschedulepublished-daysago: number }`

Onboarding Events

Event Name	Properties	Trigger
roster_uploaded	`{ staffcount: number, up-loadmethod: "csv"\	"manual_entry"\
onboardingstepcompleted	`{ step: "roster"\	"firstshiftadded"\
onboarding_abandoned	{ lastcompletedstep: string, timespentminutes: number, abandonment_surface: string }	User exits the product without completing onboarding. Fired on session end if onboarding is incomplete

Core Product Events

Event Name	Properties	Trigger
schedule_published	`{ accountid: string, weekstartdate: string, staffshiftscount: number, to-talscheduledhours: number, timetobuildminutes: number, notification_method: "sms"\	"link_only"\
smsconfirmationreceived	{ accountid: string, sched-uleid: string, staffmemberid: string, minutessincepublish: number }	Delivery receipt confirmed from SMS provider. This is the North Star activation event — it is a backend event, not a user action

overtimealerttriggered	{ accountid: string, staffmemberid: string, hoursattrigger: number, alertacknowledged: boolean }	System detects a staff member approaching or exceeding the configured overtime threshold
calloutcoveragesuggested	`{ accountid: string, shiftid: string, suggestionscount: number, suggestionaccepted: boolean, timetocover_minutes: number\	null }`
timeoffrequest_submitted	`{ accountid: string, staffmemberid: string, requestdate: string, requesttype: "dayoff"\	"availabilitychange", devicetype: string }`
timeoffrequest_actioned	`{ accountid: string, requestid: string, action: "approved"\	"denied", timetoaction_hours: number }`

Conversion & Engagement Events

Event Name	Properties	Trigger
paywall_hit	`{ accountid: string, featureattempted: string, plan_tier: "starter",	

16 DevOps & Hosting

DevOps & Hosting

Overview

This section defines the infrastructure, deployment pipeline, monitoring approach, and operational runbook for the restaurant staff scheduling SaaS. Every decision reflects one governing constraint: the product must reach a paying pilot in 6 weeks, not 6 months, and the infrastructure must not become the reason it doesn't.

One important note on architecture: the Technical Architecture section specifies Supabase as the primary database and auth provider. This DevOps section aligns with that decision — references to "Neon" in the section brief are noted, but this document uses *Supabase (PostgreSQL)* as established in the Technical Architecture section. Where Neon would be a viable alternative, it is called out explicitly. The rest of the stack (Vercel, Trigger.dev) maps directly.

The infrastructure decisions below are not symmetrical in timing. A CI/CD pipeline that deploys broken code to production before the first pilot destroys trust with the exact operators who are taking a chance on an unknown product. A Kubernetes cluster for 10 paying restaurant managers is waste that burns runway. Every decision is anchored to which failure mode it prevents and when that failure mode becomes relevant.

Infrastructure Architecture

The production infrastructure at launch is a four-service topology. No service is self-hosted. No service requires infrastructure management by the founding team before product-market fit is established.

Vercel

- *Use Vercel as the primary hosting platform for the Next.js application.* Vercel handles the manager PWA, all Next.js API routes, and the server-rendered staff schedule view from a single deployment unit. No separate server process, no separate container. The Vercel Hobby plan is sufficient for development; the Pro plan ($20/month) is required before the first pilot for custom domains, team access, and deployment protection.
- *Deploy to a single Vercel region (us-east-1 / IAD1) at launch.* Restaurant operators and staff are US-based for the initial ICP. Multi-region deployment adds complexity without a latency benefit for a US-only user base. Edge runtime is used only for auth middleware (JWT validation before each request) — not for

database-bound API routes, which require a persistent connection to Supabase and should run in the Node.js runtime.

- *Configure Vercel's serverless function timeout to 30 seconds maximum for all API routes, with a 10-second target for user-facing routes.* The Twilio webhook route, which must respond within 15 seconds to avoid Twilio retry storms, has an explicit 12-second timeout with a fallback response if the database write has not completed. A function that silently times out produces a duplicate SMS send on the next Twilio retry — a trust-destroying failure for every staff member who receives the same message twice at 8pm Sunday.
- *Enable Vercel deployment protection on the production environment.* Deployment protection requires Vercel authentication to access non-production deployment URLs. Preview deployments of branches containing unreleased features or database migrations should not be reachable from a public URL with customer data.
- *Evaluate Vercel's concurrency limits and upgrade to an Enterprise plan if Sunday evening SMS burst traffic produces function queue delays exceeding 2 seconds.* At 200 accounts publishing simultaneously, Vercel Pro's concurrent execution limits may introduce queueing. The correct response is to push more work into Trigger.dev background jobs (where the queue is explicit and observable) rather than to hold work inside serverless functions.

Supabase

- *Use Supabase Pro plan ($25/month) from the first day of pilot onboarding, not the free tier.* The free tier pauses databases after 1 week of inactivity. A restaurant manager who opens the app on a Tuesday and finds it returning a 503 because the database is paused has an immediate and permanent trust loss. The Pro plan costs $25/month and removes the pause behavior, adds 8GB database storage, and includes daily backups. This is a non-negotiable pilot requirement.
- *Enable Point-in-Time Recovery (PITR) on the Supabase Pro plan.* PITR is included on Pro at a 7-day recovery window. This means any destructive operation — an accidental DELETE without a WHERE clause, a bad migration — can be reversed to a specific timestamp within the last 7 days. Enable it the moment the production project is created, before any real data is written.
- *Create separate Supabase projects for each environment: dev, staging, and production.* A single shared Supabase project for all environments creates migration risk — a schema change tested in development that is applied to production via the wrong connection string is a data incident. Three projects mean three distinct connection strings, three distinct RLS policy sets, and no path from development code to production data without an explicit deployment step.
- *Evaluate Supabase dedicated plan when monthly active accounts exceed 500.* The Pro plan's shared infrastructure (connection pooling limits, shared compute) becomes a bottleneck at significant concurrent write volumes. The Sunday evening scheduling burst is the stress test — if Supabase Realtime subscription events begin arriving more than 3 seconds after the triggering DB write, the dedicated plan's dedicated connection pooler (PgBouncer) resolves the bottleneck. This evaluation should be triggered by observed latency, not by an arbitrary account count.

Trigger.dev

- *Use Trigger.dev Cloud (Hobby plan, free) for all background jobs from the first pilot.* The Hobby plan provides 50,000 job runs per month, which is sufficient for 100+ accounts with weekly scheduling cycles. Each schedule publish triggers approximately 20–25 job runs (one per staff member SMS plus delivery polling). At 100 accounts, a full Sunday publish cycle generates approximately 2,000–2,500 job runs — well within the free tier limit. (est.)
- *Define explicit retry policies for each job type in Trigger.dev configuration before the first pilot, not after the first failure:*
  - sms.schedulePublish: 3 retries, exponential backoff starting at 30 seconds, dead-letter queue alert after final failure
  - sms.coverageOffer: 3 retries, 15-second backoff, dead-letter queue alert
  - coverage.rankCandidates: 1 retry (Claude API), fallback to rule-based algorithm on second failure, no dead-letter queue required (degraded result is acceptable)
  - analytics.churnAlert: 1 retry, no dead-letter queue (non-critical)
- *Connect Trigger.dev to a Slack channel for dead-letter queue alerts before onboarding the first paying customer.* A failed SMS dispatch that silently lands in a dead-letter queue without alerting the founder means Maria's staff never received their schedule. The Slack alert for a dead-letter queue event must contain: job type, account ID, affected shift or schedule ID, failure reason, and a direct link to the Trigger.dev run log for that job. Fifteen minutes of Trigger.dev webhook configuration prevents the worst possible pilot failure mode.
- *Upgrade to Trigger.dev Pro ($50/month) before exceeding 50,000 monthly job runs.* Pro provides 500,000 monthly runs, concurrency controls, and priority queues. Priority queues allow Sunday evening SMS dispatch jobs to be elevated above lower-priority background work (churn detection, analytics aggregation) during peak load windows.

Upstash Redis

- *Use Upstash Redis (free tier: 10,000 requests/day) for session caching and SMS rate limiting from day one.* Upstash is serverless Redis with per-request pricing — there is no idle infrastructure cost. The free tier comfortably handles early pilot usage. A single Redis instance handles both session token validation (reducing Supabase auth round-trips) and the token bucket rate limiter for SMS sends.
- *Configure Redis key TTLs explicitly for every key type:*
  - Session tokens: 24-hour TTL (aligned with Supabase JWT refresh window)
  - SMS rate limit buckets: 1-hour TTL (rolling window resets)
  - Delivery status buffers: 15-minute TTL (Twilio delivery receipts arrive within 10 minutes in normal conditions)
  - Idempotency keys for schedule publish: 60-second TTL
- *Upgrade to Upstash Pay-as-you-go and set a monthly spend cap before exceeding 500 accounts.* At 10,000 accounts, Redis request volume from session validation alone could exceed free tier limits.

> Upstash's per-request pricing remains cost-effective at moderate scale; the spend cap prevents an unexpected billing event from a traffic spike.

CI/CD Pipeline

The CI/CD pipeline has one job: make it impossible to accidentally deploy broken code to production, and make it fast enough that a founder who needs to push a hotfix at 9pm on a Sunday (when Maria is publishing her schedule) can do so in under 5 minutes.

```
```

17 Open Issues & Tracker

Open Issues & Tracker

How to Use This Document

Each issue includes a severity rating, a description of the gap, why it creates risk, what would close it, and a suggested owner. Severity is assigned against one criterion: *Critical means the product cannot safely onboard a paying customer without resolving it. Important means it can launch but with meaningful quality or retention risk. Nice-to-have* means it is worth doing when runway and capacity allow.

Issues are drawn from conflicts, gaps, and deferred decisions across the full set of prior sections. Where an issue is directly traceable to a specific prior section, that section is named.

Critical — Blocks Launch / First Paying Pilot

C-01 · Row-Level Security Policies Are Not Yet Written or Tested

What it is: The Technical Architecture and Security & Compliance sections both identify RLS as mandatory before revenue, but neither documents which specific policies exist, what they enforce, or how they are tested. The instruction is "RLS must exist" — not "here are the policies."

Why it matters: Multi-tenant isolation is the single highest-impact security failure mode. If a policy is missing or misconfigured on any table that holds staff PII (phone numbers, names, hourly rates), one restaurant operator can read another restaurant's data. This is not a theoretical risk — it is the default Supabase behavior when RLS is enabled but a policy is missing: the table returns no rows, or with a permissive default, all rows. Either failure mode is catastrophic before a pilot.

What would resolve it: Write and explicitly test RLS policies for every table that holds customer or staff data (accounts, staffmembers, schedules, shifts, smsnotifications, session_events). Tests must include: (a) a query authenticated as Restaurant A cannot return any rows from Restaurant B's tables, and (b) a query with no auth context returns zero rows. These tests run in CI against the staging Supabase project before any pilot onboarding.

Suggested owner: Technical lead / founding engineer.

C-02 · Twilio A2P 10DLC Registration Status Unknown

What it is: The DevOps & Hosting section references Twilio A2P 10DLC as part of the SMS infrastructure, but no section documents whether registration has been initiated, what the approval timeline is, or what fallback exists during the registration window.

Why it matters: A2P 10DLC registration is required by US carriers for application-to-person SMS. Unregistered numbers are subject to carrier filtering — delivery rates drop significantly and unpredictably. The product's North Star metric (Schedules Published with Staff SMS Confirmation Received) cannot be validated if SMS messages are being silently filtered. A2P registration approval takes 2–6 weeks. (directional) If this is not started before the pilot, the pilot may launch on an unregistered number and produce misleading delivery data.

What would resolve it: Initiate Twilio A2P 10DLC brand and campaign registration immediately. Document the registration status and expected approval date. Define a contingency: if registration is not approved before the pilot start date, pilot onboarding is delayed or restricted to a small number of test numbers using a Twilio trial account. Do not onboard paying pilots on an unregistered number.

Suggested owner: Founder / operations.

C-03 · Data Processing Agreements with Third-Party Processors Are Not Signed

What it is: The Security & Compliance section identifies that DPAs must be signed with Supabase, Twilio, Vercel, PostHog, and Anthropic before onboarding paying customers. None of these are documented as complete.

Why it matters: The product processes staff PII (phone numbers, names) from the first pilot. Under GDPR, a data processor operating without a signed DPA from its sub-processors is in breach of Article 28. Under CCPA, the equivalent is a missing service provider agreement. Beyond legal risk, this is a practical risk: if a pilot customer's staff member submits a deletion request and the product cannot demonstrate a documented data flow with signed processor agreements, the operator (the data controller) is exposed.

What would resolve it: Sign or formally accept DPAs with all five processors before the first pilot contract is executed. Supabase and Twilio both have self-service DPA acceptance in their consoles. Vercel requires a request. PostHog offers EU Cloud hosting as an alternative to a US-hosted DPA. Anthropic's API terms require specific review for business data processing — this one requires a legal read, not just a checkbox. Document the completion date of each agreement in a private compliance log.

Suggested owner: Founder, with legal counsel review for Anthropic terms.

C-04 · Privacy Policy and Terms of Service Do Not Yet Exist

What it is: The Security & Compliance section specifies that a privacy policy covering GDPR and CCPA obligations must be published before onboarding pilots, accessible from the login screen, the staff schedule view, and the Settings billing page. No section indicates this document has been drafted or reviewed.

Why it matters: The staff schedule view is accessed by people who have not created accounts and have not consented to any terms — they simply received a text message with a link. The privacy policy at the foot of that view is their only mechanism for understanding what data the product holds about them and how to request its deletion. Without it, the product is processing personal data (the act of rendering a page with a staff member's name and shift) without a documented legal basis or notice mechanism. A pilot restaurant whose employee complains to a data protection authority about receiving an SMS from an unknown service with no privacy disclosure creates regulatory exposure for the operator and reputational exposure for the product.

What would resolve it: Draft and publish a privacy policy (using a lawyer-reviewed template service such as Termly or direct counsel) that explicitly covers: data categories collected, legal basis, retention periods, third-party processors, cross-border transfers, and data subject rights. Draft Terms of Service that include the Data Processing Agreement provisions described in the Security & Compliance section. Both documents must be live URLs before the first pilot agreement is signed. Legal review is strongly recommended given the employee data context.

Suggested owner: Founder, with legal counsel.

C-05 · Incident Response Process Is Not Documented

What it is: The Security & Compliance section requires a formal incident response process before the first paying customer. The process is described in outline but does not exist as a document, and no monitoring alerts are confirmed as configured.

Why it matters: GDPR Article 33 requires notification to the relevant supervisory authority within 72 hours of discovering a personal data breach. CCPA has a parallel notification obligation. Without a documented process, the 72-hour clock starts running from when a breach is discovered — and if there are no alerts configured, the discovery itself may be delayed. A pilot with 5 restaurants and 50 staff members' phone numbers in the database is already a notifiable data set if compromised.

What would resolve it: Write a two-page incident response document covering: detection sources (PostHog error alerts, Vercel error monitoring, Trigger.dev dead-letter queue, user report), initial triage steps, containment actions (disable affected account, revoke tokens), customer notification timeline, and regulatory notification trigger criteria. Configure at minimum one automated alert: an error rate spike in Vercel monitoring or PostHog.

The document does not need to be elaborate — it needs to exist and be findable in under 30 seconds at 11pm on a Sunday.

Suggested owner: Founder.

C-06 · Claude API Data Processing Terms Not Verified for Staff Data

What it is: The Security & Compliance section flags that Anthropic's API terms must be reviewed before staff-related data is sent to the Claude API for coverage ranking. This review is listed as pending and unresolved.

Why it matters: The coverage ranking job passes structured staff availability data to the Claude API. If Anthropic's business API terms permit using customer-submitted data for model training, the product is transferring employee scheduling data (which can include availability patterns, shift preferences, and indirectly compensatory information) to a third party without the data subject's knowledge or consent. This is a GDPR Article 6 legal basis problem and a reputational risk. The product cannot go live with this job active until the terms are confirmed safe.

What would resolve it: Review Anthropic's current API Terms of Service and any applicable Data Processing Addendum for business customers. Specifically confirm: (a) whether Anthropic uses API-submitted data for model training by default, (b) whether there is an opt-out mechanism, and (c) whether a DPA is available that limits data use to inference only. If terms are unfavorable and no DPA is available, the coverage ranking job must use rule-based logic exclusively until terms are resolved. Document the outcome in the compliance log.

Suggested owner: Founder, with legal counsel if terms are ambiguous.

C-07 · No Defined Rollback Procedure for Database Migrations

What it is: The DevOps & Hosting section describes a CI/CD pipeline but the document is truncated before the migration and rollback procedures are specified. No section documents how a failed database migration is reversed in production.

Why it matters: The first schema migration applied to a production database with real pilot customer data is irreversible without a tested rollback procedure. A migration that drops a column, renames a table, or applies a CHECK constraint incorrectly can corrupt the scheduling data for all active restaurants simultaneously. Unlike application code, a broken database schema cannot be fixed by reverting a Vercel deployment. The risk is highest at launch because migrations are frequent in early product iteration.

What would resolve it: Document a migration runbook before the first production migration is applied. The runbook must cover: (a) all migrations are applied to the staging environment and validated against a production data snapshot before applying to production, (b) every migration has a corresponding rollback script tested

in staging, (c) PITR timestamp is recorded before every production migration so a recovery point is known, and (d) migrations that cannot be safely rolled back (irreversible column drops) require a 24-hour staging soak before production application. Store the runbook in the architecture decision log referenced in the Security & Compliance section.

Suggested owner: Technical lead / founding engineer.

C-08 · Stripe Subscription Webhook Handler Is Not Confirmed Complete

What it is: The Security & Compliance section specifies Stripe Checkout or Elements for PCI compliance, but no section documents that the Stripe webhook handler — which handles subscription created, subscription cancelled, payment failed, and invoice paid events — has been implemented or tested.

Why it matters: If the webhook handler is missing or incomplete, a restaurant that cancels its subscription or whose payment fails will continue to have access to the product. Conversely, a restaurant whose payment succeeds may not have their account correctly upgraded from the free tier. Both failure modes are trust-destroying for a product still proving unit economics in the pilot phase. Payment failure handling also affects the churn metric tracked as Kill Condition K4 — involuntary churn from failed payments is a different problem than voluntary churn, and conflating the two produces misleading retention data.

What would resolve it: Implement and test Stripe webhook handlers for: customer.subscription.created, customer.subscription.deleted, customer.subscription.updated, invoice.paymentsucceeded, invoice.payment-failed. Each event must update the accounts table subscription_status field and trigger the appropriate access control change. Webhook signature validation using Stripe's signing secret must be implemented before the first live payment. Test with Stripe's webhook CLI in the staging environment against all event types before pilot launch.

Suggested owner: Technical lead / founding engineer.

Important — Affects Quality or Retention

I-01 · Onboarding Time-to-Value for Maria Is Not Tested Against the <20-Minute Target

What it is: The Product Brief, Customer Personas, and GTM Strategy sections all cite a <20-minute time-to-first-published-schedule as a critical activation criterion. The Product Design section describes the onboarding flow but does not document that this target has been validated against a real user completing the flow on a mobile device.

Why it matters: The 20-minute target is the product's most important activation claim. If the actual median time is 35 minutes — entirely plausible given roster setup, staff phone number collection, and first schedule creation — the activation rate will be materially lower than projected. The Analytics section targets >60% onboarding completion, but completion at 35 minutes is a different product than completion at 18 minutes. Reversion to Google Sheets is the identified primary failure mode, and it is most likely to occur in the first session if the process takes longer than expected.

What would resolve it: Conduct moderated usability testing of the onboarding flow with 3–5 people matching the Maria persona (non-technical, phone-first, under mild time pressure) before pilot launch. Measure actual time from account creation to first SMS confirmation received. If median exceeds 20 minutes, identify the specific steps causing delay and redesign before onboarding paying pilots. The 20-minute claim should not appear in marketing materials until it is validated by observed behavior, not assumed.

Suggested owner: Product / founder.

I-02 · Staff SMS Opt-Out and STOP Handling Flow Is Not Defined

What it is: Twilio automatically processes STOP replies from SMS recipients and removes those numbers from future sends. No section defines what happens in the product when a staff member opts out: whether the manager is notified, whether the staff member's record is updated, and how the manager is expected to re-establish contact.

Why it matters: Staff members who reply STOP to a schedule notification create a silent communication failure. The manager publishes a schedule, the product reports successful delivery to all staff, but one staff member (who replied STOP two weeks ago) never received it. That staff member doesn't show up for their shift. The manager blames the product, not the opt-out behavior. This is a churn-triggering event, and it is most likely to happen during the first month of a pilot — exactly when the product is trying to prove its core value. STOP handling is also a carrier compliance requirement, not optional.

What would resolve it: Define the STOP handling flow: when Twilio receives a STOP reply from a staff member's number, the staffmembers record for that number is flagged smsopted_out: true and the manager receives an in-app notification on their next login (not an SMS, which they can't receive) stating "[Name] has opted out of SMS notifications. You'll need to contact them directly for schedule updates." The staff member's schedule view link remains accessible. Document this in the Product Design section and implement before pilot launch.

Suggested owner: Product / technical lead.

I-03 · Multi-Location Account Model Is Architecturally Underspecified

What it is: The Monetization section defines a Manager tier at $89/month supporting up to 3 locations. The Customer Personas section defines Devon as a 2–3 location general manager. The Technical Architecture section does not specify the data model for multi-location accounts — specifically whether a single account row covers multiple locations, or whether each location is a separate account linked to a parent, and how RLS policies apply across this structure.

Why it matters: The data model decision for multi-tenancy with multiple locations is irreversible once pilots are onboarded. If locations are modeled as separate accounts, the Manager tier's cross-location visibility feature requires cross-account queries — which directly conflict with the RLS policy model designed to prevent cross-account data access. If locations are modeled as sub-entities under a single account, the schema must be designed before any data is written. Getting this wrong in Month 1 requires a migration on production data in Month 3, during the phase when retention is being measured for Kill Condition K3.

What would resolve it: Define the multi-location data model explicitly before any schema is written: accounts ’ locations ’ staffmembers (and schedules, shifts) with RLS policies scoped to accountid (not locationid). The Manager tier user has access to all locationid rows where account_id matches their account. Document this in the Technical Architecture section as a confirmed decision, not an assumption. Validate that Devon's cross-location visibility use case is satisfiable within this model before building.

Suggested owner: Technical lead / founding engineer.

I-04 · Callout Coverage Ranking Algorithm Behavior Under Edge Cases Is Undefined

What it is: The Product Design section describes the Callout Coverage screen as showing "ranked available staff (not scheduled, prior experience, under OT)" with one-tap notify. The Technical Architecture section specifies Claude API with rule-based fallback. Neither section defines behavior when: there are zero available staff, all available staff are at OT threshold, or the Claude API returns an unexpected response format.

Why it matters: Callout coverage is the highest-stress moment in the product's usage cycle — a manager dealing with a no-show at 6:45am needs the coverage screen to return something useful, not a blank list or an error. If the screen fails silently (returns an empty ranked list because all candidates are filtered out) and the manager has no fallback path visible, they immediately switch to their group text thread. This is the identified reversion trigger from the Product Brief, and it occurs at the exact moment the product is supposed to prove its value.

What would resolve it: Define explicit fallback states for the Coverage screen: (a) if zero staff are available after all filters, show all staff with their current schedule status visible and a note explaining why they appear below the recommended list, (b) if all candidates are at or above OT threshold, show them anyway with an OT warning label and let the manager decide, (c) if Claude returns an error or unexpected format, fall back

to rule-based ranking without surfacing the AI failure to the user. Specify these states in the Product Design section and test them with mock data before pilot launch.

Suggested owner: Product / technical lead.

I-05 · Free Tier Conversion Path Is Defined in Monetization But Not in the Product

What it is: The Monetization section defines a Starter free tier (up to 8 staff, 1 location) as a "conversion lever for evaluation." No section specifies what the in-product upgrade path looks like, what triggers the upgrade prompt, or what happens to a free-tier account that exceeds 8 staff members.

Why it matters: A free tier that doesn't convert is a cost center — it consumes Supabase storage, Twilio message credits (if SMS is available on free), and support attention. The conversion mechanism is critical to the unit economics: the Monetization section's $55 ARPU target assumes a specific free-to-paid conversion rate, but that rate is entirely dependent on whether the upgrade prompt appears at the right moment, with the right framing. A hard block at 9 staff members with no context is a churn trigger. A well-timed prompt when Maria first tries to add a staff member beyond the limit, positioned as "You've built your team — here's what you unlock at $49/month" is a conversion event.

What would resolve it: Define the upgrade prompt trigger conditions, copy, and screen placement before the free tier is live. Minimum required states: (a) user attempts to add the 9th staff member — inline upgrade prompt with feature comparison, (b) user has published 3+ consecutive schedules — proactive upgrade prompt in the Schedule screen based on demonstrated value, (c) settings billing page — explicit tier comparison table. Specify which features are visibly gated vs. hidden on the free tier.

Suggested owner: Product / founder.

I-06 · Vercel Function Timeout Configuration for Twilio Webhooks Has a Race Condition

What it is: The DevOps & Hosting section specifies a 12-second timeout for the Twilio webhook route to avoid retry storms, with "a fallback response if the database write has not completed." The fallback behavior is not defined — specifically what HTTP response code is returned and whether Twilio interprets it as a success or a failure triggering a retry.

Why it matters: Twilio expects a 200 OK response within 15 seconds or it retries the webhook. If the product returns a 500 or 503 because the database write timed out at 12 seconds, Twilio retries the same event — which could produce a duplicate inbound reply processing and, in a worst case, duplicate staff member records or double-counted delivery confirmations. The Analytics section's North Star metric (SPSC-WAA) is only reliable if delivery confirmations are idempotent.

What would resolve it: Define the Twilio webhook timeout handling explicitly: if the database write does not complete within 10 seconds, return 200 OK with a body of <Response/> (Twilio expects TwiML or an empty 200 for no-reply handling), log the incomplete write to the dead-letter queue for async processing, and reconcile the delivery status via the next Twilio delivery receipt polling job. The 200 response prevents Twilio retries; the dead-letter queue ensures the event is not lost. Add an idempotency key on inbound webhook processing keyed to the Twilio MessageSid.

Suggested owner: Technical lead.

I-07 · Analytics Event Schema Has No Documented PII Exclusion Enforcement

What it is: The Analytics section defines 18 tracked events and specifies PostHog as the analytics platform. The Security & Compliance section states that phone numbers must never appear in analytics event properties. No section documents how this exclusion is enforced in code — whether it is a code review standard, a PostHog property filter, a middleware layer, or a linting rule.

Why it matters: Analytics event schemas in early products drift over time. An engineer adds staffphone to a shiftassigned event for debugging purposes. That field is now in PostHog's data store (a third-party processor), potentially without a DPA that covers PII. At audit time — or at a data subject access request — the product cannot demonstrate that PII is excluded from analytics without reviewing every event's payload individually. This is not a theoretical risk; it is the most common source of accidental PII leakage in SaaS products.

What would resolve it: Implement a server-side analytics event wrapper that explicitly strips a defined list of disallowed fields (phone_e164, `