Restaurant Menus That Learn What You Like — Product Specification

Generated March 18, 2026

Restaurant Menus That Learn What You Like — Product Specification

Version: 1.0 | **Date:** March 17, 2026 | **Status:** Draft for Engineering Review

Problem Statement

The restaurant industry completed a forced digital transformation during COVID-19 and then stopped. Seventy-five percent of restaurants globally now deploy QR code menus (TableQR, 2025), yet 81% of U.S. diners still prefer a physical menu over a QR code, and only 1% rate QR codes as their favorite menu format in a survey of 850 U.S. diners (TableQR, 2025). The industry digitized the delivery mechanism without improving the experience. The result is a static PDF behind a scannable square — the lowest-fidelity expression of what a connected, data-rich digital menu could be.

The operator-side consequences are measurable and costly. Thirty percent of a typical restaurant's sales come from just three menu items, yet most operators price and position every item by gut feel rather than margin data (LinkedIn/Restroworks, 2025). Average operators waste 8–12% of food inventory; world-class operators waste 3–5% — the gap attributable entirely to measurement systems (LinkedIn/Restroworks, 2025). When a kitchen runs out of an item, the 86'd status propagates through a verbal chain from cook to server to diner, introducing a lag that produces the most common negative dining experience: ordering something that isn't available. There is no automated mechanism in the current QR menu ecosystem to remove an 86'd item from the diner's view in real time.

The diner-side gap is equally documented. Eighty-three percent of consumers are receptive to personalized restaurant experiences, yet only 44% report receiving offers or recommendations they find relevant (QSR Magazine, citing PYMNTS research). The average Gen Z adult belongs to 4.4 restaurant loyalty programs (National Restaurant Association, 2024), demonstrating demonstrated willingness to share data in exchange for personalized value — but current QR menus collect no preference signal whatsoever. A diner who visits the same fast-casual chain weekly receives an identical, undifferentiated menu on every scan.

The PDF complaint is not merely a UX irritation. PDF menus are not indexed by search engines, meaning dish-level content does not appear in queries. PDF menus embedded as images cannot be read by screen readers, creating accessibility barriers for disabled diners (Embark Studio, 2025). A Hacker News commenter in 2021 described the canonical failure: "My big issue is that every menu QR code I've scanned results in a PDF I have to download... sized for the desktop, which means you have to pinch and scroll to see anything." A Reddit user in 2025 described scanning a QR menu and receiving "just a plain list of food names and prices. No images. No context. Nothing that answered basic questions."

The market has digitized the menu without making it intelligent. No competitor identified in systematic competitive research offers the complete combination of cross-restaurant diner preference learning, real-time 86'd-item management, dynamic pricing for slow movers, and frictionless QR access without an app download. The infrastructure exists. The adoption behavior is established. The intelligence layer is unbuilt.

Target Audience

Persona 1: The Multi-Unit Fast-Casual Operator (B2B, Primary)

Who they are: Operators of 3–20 fast-casual restaurant locations — think regional taco chains, build-your-own bowl concepts, or local burger franchises — who are already running at least one SaaS tool (scheduling, inventory, or POS) and have a designated person responsible for technology decisions. This is typically an owner-operator, a VP of Operations, or a Director of Marketing at the chain level. Annual revenue per location ranges from $800K to $2.5M. Net margins are 3–9%.

Why not single-location independents in Phase 1: Independent single-location operators face three compounding barriers that make them a poor initial beachhead. Only approximately 31% of smaller hospitality properties have adopted comprehensive SaaS modules (Hospitalitynet.org, 2025). Their organizational fragmentation and low IT literacy create high support cost per dollar of revenue. Most critically, the personalization flywheel requires repeat visit data to generate value — a single-location independent with low repeat visit rates cannot demonstrate ROI within a 60-day pilot window. They are a Phase 2 audience once the product has proven ROI in the multi-unit context and has reduced onboarding friction to a self-service motion.

Behavioral profile: These operators have experienced the pain of manual data reconciliation across locations. They make pricing decisions based on intuition rather than dish-level margin data. They have likely tried and churned at least one analytics SaaS tool because it required too much manual data entry. They respond to outcome-framed pitches ("this pays for itself when three additional dessert orders are added per table turn") and require a 30–60 day pilot with pre-agreed metrics before committing to an annual contract.

Goals: Reduce food waste, increase check average, eliminate the 86'd-item embarrassment, and understand which menu items are actually driving margin — not just revenue.

Constraints: Cannot absorb tools that require POS replacement, server retraining, or more than two hours of onboarding time. Technology spend decisions require demonstrable ROI within one billing cycle.

Persona 2: The Returning Millennial/Gen Z Diner (B2C, End User)

Who they are: Adults aged 18–43 who dine out 2–4 times per week, belong to at least one restaurant loyalty program, and use their phone as the primary interface for restaurant interactions. They are accustomed to algorithmic recommendation quality from Netflix and Spotify and bring that expectation to food ordering. Fifty-eight percent of Gen Z loyalty members say they are less likely to try a new restaurant because they prefer establishments where they have membership (QSR Magazine, 2025).

Behavioral profile: They scan QR codes immediately upon sitting — 78% of first-time visitors and 65% of return customers do so (Menu.Miami, 2025). They experience anxiety around ordering from an unfamiliar menu (Dynamic Yield, 2025). They have dietary restrictions or preferences — vegan, gluten-free, halal, high-protein — that they must manually communicate on every visit. They do not want to download an app to get a better menu experience, but they will provide an email address or phone number in exchange for a meaningfully personalized result.

Goals: Order confidently, avoid dishes that conflict with dietary needs without having to ask the server, and discover high-probability-of-enjoyment dishes at restaurants they haven't visited before.

Constraints: Will abandon a consent or profile creation flow that takes more than 60 seconds. Will not download a native app as a prerequisite for a better menu. Will not tolerate seeing a menu item they ordered that turns out to be unavailable.

Persona 3: The Chain Manager (B2B, Secondary)

Who they are: A director or VP at a multi-unit operator overseeing 5–20 locations. They do not manage individual restaurant operations day-to-day but are accountable for cross-location performance, menu consistency, and marketing ROI. They currently synthesize location-level data manually from multiple exports.

Goals: Identify which menu items are underperforming across the chain, understand location-level variation in diner preferences, and make menu engineering decisions based on actual scan and order data rather than sales rep anecdotes.

Goals & Success Metrics

Operator-Side Metrics (measured per pilot restaurant)

Metric	Baseline	Target at 60 Days	Measurement Method
Check average lift	Operator's prior 90-day average	+8% vs. baseline	POS order data comparison
Scan-to-order conversion rate	Estimated 40–55% for static QR menus	≥65%	Platform scan events vs. order completions
86-item lag time	Verbal chain, unmeasured (estimated 5–15 min)	<30 seconds from operator action to menu removal	EightySixEvent timestamp vs. menu propagation timestamp
Monthly operator churn rate	N/A (new product)	<5% after pilot conversion	Active subscription count month-over-month
Dish view-to-add-to-order rate for promoted items	Baseline from first 2 weeks of pilot	+15% for algorithmically surfaced items vs. non-surfaced	Platform analytics

Diner-Side Metrics (measured per diner cohort)

Metric	Target at 60 Days	Measurement Method
Return scan rate (same diner, same restaurant, ≥2 scans)	≥35% of profiled diners	DinerProfile scan event count
Profile completion rate (dietary preferences set)	≥50% of diners who create a profile	DinerProfile.dietary_restrictions populated
Recommendation acceptance rate (surfaced item ordered)	≥25% of surfaced recommendations result in order	Recommendation event vs. order event join
Consent acceptance rate	≥60% of first-time scanners complete consent flow	ConsentRecord creation vs. QR scan event

Network Effect Metric

Metric	Target at 90 Days	Rationale
Diner profiles per participating restaurant	≥150 active profiles per location	Minimum density to generate statistically meaningful preference signals for recommendation ranking

Metric	Target at 90 Days	Rationale
Cross-restaurant profile reuse rate	≥15% of profiles used at ≥2 restaurants	Leading indicator of network effect moat forming

Business Metrics

Metric	Target	Window
Monthly Recurring Revenue	$25,000–$50,000	At end of 90-day Phase 1
Pilot-to-paid conversion rate	≥60% of pilot operators convert to paid	Within 30 days of pilot completion

---

Non-Goals / Out of Scope

The following are explicitly out of scope and must not be built, implied in sales materials, or scoped into engineering estimates without a formal spec revision.

No POS replacement or payment processing. This platform reads from and writes to existing POS systems via API and webhook. It does not process payments, manage tabs, or replace any POS function. Scope creep into payment flows creates PCI-DSS obligations and competitive conflict with Toast and Square.

No third-party delivery platform integration. DoorDash, Uber Eats, and Grubhub integrations are Juicer's domain. This platform serves in-restaurant diners only. Delivery menu management is a separate product category.

No voice or phone channel. Phone-based ordering and reservation handling is Slang AI's domain. This platform is browser-based QR only.

No staff-facing CRM or server tablet experience. Staff-facing guest intelligence is Loyalist's domain. This platform puts intelligence in the diner's hand, not the server's. Building a server-facing layer would require hospitality workflow design and server training that is outside the current scope.

No dynamic pricing in Tennessee at any phase. Tennessee SB 1807, effective July 1, 2026, makes personalized algorithmic pricing an unfair or deceptive act under the Tennessee Consumer Protection Act. The dynamic pricing module must be hard-blocked for any restaurant location with a Tennessee address at the jurisdiction level, with no operator override.

No dynamic pricing without mandatory disclosure in New York. New York's Algorithmic Pricing Disclosure Act requires explicit consumer notification when prices are set by algorithms using personal data. The dynamic pricing module may not be activated for New York locations without a compliant disclosure flow built into the diner-facing UI. This is a launch requirement for Phase 3, not a post-launch compliance item.

No native iOS or Android app in Phase 1. The diner experience is a mobile web browser experience only. Native app development introduces app store review cycles, push notification infrastructure, and a mandatory download barrier that contradicts the core no-app-required value proposition.

No single-location independent restaurant sales motion in Phase 1. The sales team must not pursue, onboard, or support single-location independent restaurants during Phase 1. This is a go-to-market constraint, not a technical one. Independents are Phase 2 after self-service onboarding is built.

No cross-restaurant pricing signal sharing. The dynamic pricing module must operate on each restaurant's own historical data only. No aggregated pricing signals, no benchmarking against other restaurants on the

platform, and no shared pricing recommendations across competing operators. This is a hard architectural constraint driven by DOJ hub-and-spoke antitrust risk.

No browser fingerprinting as the diner identity mechanism. Diner identity must be anchored to phone number or email address with explicit consent. Browser fingerprinting is legally indefensible under GDPR and CCPA for the cross-restaurant profile use case and must not be used at any phase.

No gamified rewards or loyalty points system in Phase 1. Gamification mechanics require loyalty program infrastructure, points ledger accounting, and redemption flows that are out of scope. Diners receive personalization value, not points.

No family or group ordering with shared preferences in Phase 1. Group preference reconciliation requires a separate UX design pattern and data model that is deferred to Phase 2.

No embeddable menu widget for restaurant websites in Phase 1. The widget use case requires a separate embedding contract, cross-origin security policy, and SEO architecture that is deferred.

No white-label product offering before the enterprise tier is defined. White-labeling requires brand configuration infrastructure, separate deployment pipelines, and a legal framework for reseller agreements that must not be improvised.

User Stories / Jobs to Be Done

Actor: Restaurant Operator (Admin)

Story OP-1: First-Time Menu Setup When I am onboarding a new location, I want to upload or sync my existing menu and configure dietary tags, so I can go live with an intelligent QR menu without rebuilding my menu from scratch.

Acceptance Criteria:

- Operator can import a menu via CSV template or Toast/Square sync within 30 minutes of account creation
- Each menu item can be tagged with ≥1 dietary attribute (vegan, vegetarian, gluten-free, halal, kosher, contains nuts, contains dairy, contains shellfish)
- Menu goes live (QR code scannable by diners) within 60 minutes of completing setup
- Operator receives a confirmation with a downloadable QR code asset and a table card template

Story OP-2: Marking an Item as 86'd When my kitchen runs out of an item mid-service, I want to mark it as 86'd from my phone in under 10 seconds, so diners stop ordering it and I stop having to apologize.

Acceptance Criteria:

- 86 action is accessible from the operator dashboard in ≤3 taps/clicks
- Item is visually removed from all active diner menu sessions within 30 seconds of operator action
- 86 event is timestamped and logged in EightySixEvent table with operator_user_id
- Item is automatically restored to available status at the start of the next business day unless the operator sets a manual restore date
- Operator receives confirmation that propagation completed successfully

Story OP-3: Enabling Dynamic Pricing on a Slow Mover When I notice a menu item has low order velocity in the afternoon, I want to enable a time-limited discount on that item, so I can move inventory and increase cover revenue during slow periods.

Acceptance Criteria:

- Operator can view items ranked by order velocity in the last 7 days from the dashboard
- Operator can set a discount percentage (5–40%) and a time window (start time, end time, days of week) for any item
- System checks restaurant location's jurisdiction before activating; blocks activation with explanatory error message if jurisdiction is Tennessee or if NY disclosure flow is not completed
- Discounted price is displayed to diner with original price struck through and a "Today's Special" label
- Dynamic pricing rule is logged in PricingRule table with all parameters and operator_user_id

Actor: Diner (First-Time)

Story DN-1: First QR Scan with No Prior Profile When I scan a QR code at a restaurant I've never visited, I want to quickly tell the system my dietary restrictions, so I see a menu that's already filtered to what I can eat.

Acceptance Criteria:

- Menu loads in browser (no app download prompt) within 2 seconds on a 4G connection
- Consent and profile creation flow is completable in ≤60 seconds
- Diner can select dietary restrictions from a predefined list and add a free-text allergy note
- Menu is immediately filtered to show only compliant items after preference selection
- Diner can decline consent and still access an unfiltered, anonymous menu with no data stored

Story DN-2: Diner Setting Dietary Restrictions When I have a new dietary restriction I didn't previously set, I want to update my profile from any restaurant's QR menu, so my preferences are reflected everywhere I go.

Acceptance Criteria:

- Profile edit is accessible from the menu UI in ≤2 taps
- Changes to dietary restrictions propagate to the current session's menu display immediately (no page reload required)
- Updated preferences are persisted to DinerProfile and reflected on next scan at any participating restaurant

Actor: Diner (Returning)

Story DN-3: Returning Diner Profile Retrieval When I scan a QR code at a restaurant I've visited before, I want my preferences and past order history to be recognized automatically, so I get a personalized menu without re-entering my information.

Acceptance Criteria:

- Returning diner on same device within 30-day device recognition window sees personalized menu without OTP re-entry (device recognition token, distinct from session token, is valid for 30 days)
- Returning diner on a new device or after 30-day expiry is prompted for phone/email OTP, which completes in ≤3 steps
- Menu displays "Based on your past orders" label on recommended items
- Profile retrieval failure (identity not found) falls back gracefully to first-time diner flow with option to re-link profile

Actor: Chain Manager (Multi-Location)

Story CM-1: Cross-Location Analytics View When I am reviewing weekly performance, I want to see dish-level analytics aggregated across all my locations, so I can identify which items to promote, retire, or reprice chain-wide.

Acceptance Criteria:

- Chain manager dashboard shows scan rate, add-to-order rate, and 86-frequency for each menu item, filterable by location and date range
- Data is available at the Organization level (all locations) and drillable to individual Location level
- Export to CSV is available for any date range
- Dashboard loads within 3 seconds for chains with up to 20 locations
- Chain manager cannot view raw diner profile data — only aggregated, anonymized dish performance metrics

---

Functional Requirements

Module 1: QR Menu Display Engine

FR-1.1 The system must render a complete menu page in a mobile browser without requiring any app download, app store redirect, or browser plugin installation.

FR-1.2 The menu page must load at P95 < 2 seconds on a simulated 4G connection (10 Mbps download, 50ms latency).

FR-1.3 The system must support dietary/allergen filtering across the following attributes: vegan, vegetarian, gluten-free, halal, kosher, contains-nuts, contains-dairy, contains-shellfish, contains-eggs, contains-soy. Each attribute must be independently toggleable.

FR-1.4 The system must remove any item with an active EightySixEvent from the diner-facing menu display within 30 seconds of the operator action that created the event, for all active browser sessions at that location.

FR-1.5 The menu must display item name, description, price, dietary tags, and at least one image per item (image is optional for operator but the layout must not break if no image is provided).

FR-1.6 The menu must meet WCAG 2.1 AA compliance: minimum 4.5:1 color contrast ratio for all text, all interactive elements must be keyboard-navigable, all images must have alt text populated from item description.

FR-1.7 The menu must display an empty-state message ("We're refreshing this section — check back soon!") when all items in a category are simultaneously 86'd, rather than displaying an empty category header.

FR-1.8 The menu must function in offline or degraded connectivity mode by serving a cached version of the last-loaded menu state, with a visible banner indicating "Menu may not reflect latest availability."

Module 2: Diner Preference Engine

FR-2.1 The system must create a DinerProfile anchored to a verified phone number or email address. Browser cookies and device fingerprinting must not be used as the primary identity anchor.

FR-2.2 The system must present a consent flow on first scan that completes in ≤2 user interactions (one interaction to view the consent summary, one interaction to accept or decline).

FR-2.3 If a diner declines consent, the system must display the full unfiltered menu with no personalization and must not create any DinerProfile record or store any behavioral data for that session.

FR-2.4 The DinerProfile must persist across all participating restaurants. A diner's dietary restrictions set at Restaurant A must be applied automatically when they scan at Restaurant B.

FR-2.5 The recommendation ranking algorithm must surface items by combining: (a) past order history at this restaurant, (b) ratings given by the diner, (c) dietary restriction compliance, (d) items ordered at other participating restaurants in the same cuisine category. The algorithm must not use cross-restaurant pricing data as a ranking signal.

FR-2.6 The system must display a maximum of 5 "Recommended for You" items at the top of the menu. Recommended items must also appear in their original category position.

FR-2.7 The system must support a diner-initiated profile deletion request that removes all DinerProfile data within 30 days, in compliance with GDPR Article 17 and CCPA deletion rights.

FR-2.8 A diner must be able to update dietary restrictions from within the menu UI without leaving the menu page.

Module 3: Operator Dashboard

FR-3.1 The operator dashboard must display, for each menu item over a selectable date range: total scan impressions, add-to-order count, add-to-order rate (add-to-order / impressions), and 86-frequency (number of times marked 86'd).

FR-3.2 The dashboard must support menu CRUD operations: create item, edit item (name, description, price, image, dietary tags), archive item (soft delete, not visible to diners but retained in analytics history), and restore archived item.

FR-3.3 The 86 management interface must allow an operator to mark any item as 86'd in ≤3 taps from the dashboard home screen on a mobile browser.

FR-3.4 The dashboard must show a real-time indicator of how many active diner sessions are currently viewing the menu at each location.

FR-3.5 The dashboard must support role-based access: Organization Admin (full access across all locations), Location Manager (access to single assigned location only), Read-Only Analyst (view analytics, no edit permissions).

FR-3.6 The dashboard must be functional on a mobile browser (iOS Safari, Android Chrome) without requiring a desktop browser.

Module 4: Dynamic Pricing Module

FR-4.1 The system must detect "slow mover" items defined as: items with an add-to-order rate in the bottom 20th percentile for the current day-part (breakfast, lunch, dinner) compared to the trailing 14-day average for the same day-part.

FR-4.2 The system must surface slow-mover items to the operator with a suggested discount range (5–40%) calculated to bring the item's projected margin to within 10% of its full-price margin.

FR-4.3 The system must perform a jurisdiction check before activating any pricing rule. If the restaurant location's state is Tennessee, the activation must be blocked with the error: "Dynamic pricing is not available in Tennessee (SB 1807). Contact support for details."

FR-4.4 For New York locations, the system must require the operator to complete a disclosure acknowledgment workflow (see FR-4.9) before a pricing rule can be activated. Activation without completed acknowledgment must return a 422 error with code `disclosure_acknowledgment_required`.

FR-4.5 The disclosure acknowledgment for New York must include a diner-facing disclosure statement displayed on the menu: "Prices on this menu may be personalized based on your dining history." This disclosure must be rendered in a minimum 12px font, not hidden behind a tooltip or collapsed element.

FR-4.6 The operator must record their disclosure acknowledgment via a dedicated endpoint (`POST /v1/operator/locations/{location_id}/pricing-rules/disclosure-acknowledgment`) that accepts a boolean `acknowledged: true` and returns a `disclosure_acknowledgment_id` stored on the Location record. A pricing rule activation for a NY location must include this `disclosure_acknowledgment_id` in the request body.

FR-4.7 The dynamic pricing module must use only that restaurant's own historical order data as input. Cross-restaurant pricing data must not be used as a signal under any circumstances.

FR-4.8 All pricing rule activations, modifications, and deactivations must be logged in the PricingRule audit trail with timestamp, operator_user_id, original price, discounted price, and jurisdiction.

FR-4.9 The system must display discounted prices to diners with the original price struck through, the discounted price in a visually distinct color, and a "Today's Special" label. The diner must not be shown that the discount is personalized to them unless the NY disclosure requirement applies.

Module 5: Multi-Location Management

FR-5.1 An Organization Admin must be able to view cross-location analytics for all locations within their Organization from a single dashboard view.

FR-5.2 The location hierarchy must support three levels: Organization → Chain (optional grouping of locations) → Location. A Chain entity is a named grouping within an Organization used for regional or brand segmentation.

FR-5.3 Menu templates must be creatable at the Organization level and pushable to selected locations, with location-level overrides permitted for price and item availability.

FR-5.4 Cross-location analytics must aggregate dish performance data without exposing individual diner profile data to the chain manager. All cross-location analytics must be anonymized and aggregated at the item level.

FR-5.5 The system must enforce tenant isolation: a Location Manager at Restaurant A must not be able to query, view, or infer any data belonging to Restaurant B, even if both are on the same platform.

Non-Functional Requirements

NFR-1: Menu Page Load Performance The diner-facing menu page must load to first contentful paint at P95 < 2 seconds and to interactive at P95 < 3 seconds, measured on a simulated 4G connection (10 Mbps down, 50ms RTT). This must be verified in CI for each deployment using Lighthouse CI with a minimum score of 80 on Performance.

NFR-2: 86-Item Propagation Latency From the moment an operator submits an 86 action to the moment the item is removed from all active diner browser sessions at that location, the elapsed time must be < 30 seconds at P99. This must be measured via synthetic test: a test session subscribed to the SSE channel must receive the removal event within 30 seconds of the EightySixEvent being created in the database.

NFR-3: System Availability The platform must maintain 99.9% uptime during restaurant operating hours, defined as 6:00 AM to midnight local time for each restaurant location. This translates to a maximum of 43.8 minutes of downtime per month during operating hours. Scheduled maintenance must occur between midnight and 5:00 AM local time. Availability must be monitored per-location, not globally, because a regional outage affecting a subset of locations still violates the SLA for those locations.

NFR-4: Concurrent Session Support The system must support up to 200 concurrent active diner browser sessions per restaurant location without degradation in menu load time (P95 < 2s maintained). Load testing must validate this threshold before Phase 1 launch.

NFR-5: WCAG 2.1 AA Compliance The diner-facing menu display must pass WCAG 2.1 AA automated checks (axe-core or equivalent) with zero critical violations. Manual testing must verify keyboard navigation and screen reader compatibility (VoiceOver on iOS, TalkBack on Android) before launch.

NFR-6: Consent Flow Completion The GDPR/CCPA consent flow must be completable by a diner in ≤2 user interactions. The consent record must be created server-side within 1 second of the diner's acceptance action. The consent flow must not use pre-checked opt-in boxes.

NFR-7: Data Retention and Deletion SLA DinerProfile data must be deletable within 30 days of a verified deletion request (GDPR Article 17 / CCPA). Anonymized, aggregated analytics data (AnalyticsSnapshot) is not subject to deletion and may be retained indefinitely. Operator data must be retained for 7 years for tax/audit purposes after account termination. Raw session logs must be purged after 90 days.

NFR-8: Authentication Security Operator JWT tokens must expire after 8 hours of inactivity. Diner session tokens must expire after 4 hours. Device recognition tokens (for returning-diner skip-OTP flow) must be distinct from session tokens, stored in localStorage, and expire after 30 days. A diner arriving at a new restaurant location may use their device recognition token to initiate profile retrieval without a new OTP, provided the token was issued for the same diner identity (phone/email) — the token is diner-scoped, not location-scoped. This resolves the security boundary: the device recognition token authenticates the diner's identity; a new location-scoped session token is issued upon successful profile retrieval at the new location.

NFR-9: Input Validation All API endpoints must validate and sanitize inputs against OWASP Top 10. SQL injection and XSS vectors must be blocked at the API layer. File uploads (menu item images) must be validated for MIME type and scanned for malware before storage.

NFR-10: Consent Record Integrity The `ip_address` field on a ConsentRecord must be captured server-side from the HTTP request context (e.g., `X-Forwarded-For` header after proxy trust validation) and must never be accepted from the client request body. The client-submitted consent payload must not include an `ip_address` field; if present, it must be ignored. This is a consent record integrity requirement.

NFR-11: Observability Every API endpoint must emit structured logs (JSON) with: request_id, endpoint, response_time_ms, status_code, tenant_id, and error_code (if applicable). P95 latency, error rate, and availability must be dashboarded in real time. Alerting must trigger if: error rate exceeds 1% over a 5-minute window, P95 latency exceeds 3 seconds, or 86-propagation latency exceeds 30 seconds.

NFR-12: Multi-Tenant Data Isolation Each restaurant's data must be isolated at the database query layer using row-level security or equivalent tenant-scoping. A query issued in the context of Tenant A must be provably incapable of returning rows belonging to Tenant B. This must be verified by a dedicated security test suite that attempts cross-tenant data access.

User Flows

Flow 1: First-Time Diner QR Scan → Consent → Profile Creation → Personalized Menu

Happy Path:

1. Diner scans QR code at table. Browser opens menu URL with `location_id` and `table_id` query parameters.
2. System detects no device recognition token in localStorage. Renders consent screen (not the menu).
3. Consent screen displays: what data is collected, how it is used across restaurants, and a link to the privacy policy. Two buttons: "Personalize My Menu" and "Browse Without Saving."
4. Diner taps "Personalize My Menu." System prompts for phone number or email address.
5. Diner enters phone number. System sends OTP via SMS. OTP entry screen displayed.
6. Diner enters OTP. System verifies. DinerProfile created. ConsentRecord created with server-captured IP, timestamp, consent_version, and identity_anchor (phone/email hash).
7. System prompts: "Any dietary restrictions?" Diner selects from predefined list. Profile updated.
8. Personalized menu rendered. "Recommended for You" section populated (empty on first visit — shows most popular items at this location as fallback). Device recognition token written to localStorage (30-day TTL).

Error States:

- OTP not received: "Didn't get a code? Resend" link available after 30 seconds. Maximum 3 OTP attempts before 15-minute lockout with user-visible message.
- Invalid phone number format: Inline validation error before submission. "Please enter a valid 10-digit phone number."
- OTP expired (>10 minutes): "This code has expired. Request a new one." Diner returns to phone entry step.

Empty State: First-time diner with no order history sees "Most Popular Here" section instead of "Recommended for You." Label reads "Discover This Menu" to avoid implying personalization that hasn't yet occurred.

Consent Declined Path: Diner taps "Browse Without Saving." Menu renders immediately with no dietary filter pre-applied. No DinerProfile created. No ConsentRecord created. No session data stored beyond the current browser session. A persistent but non-intrusive banner reads: "Want personalized recommendations? Tap here to set up your profile." Tapping the banner restarts the consent flow.

Flow 2: Returning Diner QR Scan → Identity Resolution → Profile Retrieval → Ranked Menu

Happy Path (same device, within 30-day window):

1. Diner scans QR code. Browser opens menu URL.
2. System detects valid device recognition token in localStorage. Extracts diner_id from token.
3. System validates token server-side (not expired, not revoked). Issues a new 4-hour location-scoped session token for this location.
4. DinerProfile retrieved. Dietary restrictions applied. Recommendation ranking executed.
5. Personalized menu rendered with "Recommended for You" section populated from order history and ratings. No OTP required.

Happy Path (new device or expired token):

1. Diner scans QR code. No device recognition token found.
2. Consent screen displayed. Diner taps "Personalize My Menu."
3. Diner enters phone/email. OTP sent and verified.
4. System finds existing DinerProfile for this identity. Profile retrieved (not created new).
5. New device recognition token written to localStorage. Personalized menu rendered.

Error States:

- Identity not found (phone/email not in system): System treats as new diner and proceeds through Flow 1 profile creation. Message: "We don't have a profile for this number yet — let's create one."
- OTP failure after 3 attempts: 15-minute lockout. Message: "Too many attempts. Please try again in 15 minutes or browse the menu without personalization." Anonymous menu offered immediately.
- Device recognition token tampered or invalid (signature mismatch): Treated as expired. OTP flow initiated.
- Profile retrieval timeout (>3 seconds): Anonymous menu rendered immediately with message: "Personalization is loading — your preferences will appear shortly." Async retry in background.

Returning Diner at a New Participating Restaurant: Device recognition token is diner-scoped (not location-scoped). System validates the token, retrieves the DinerProfile, and issues a new location-scoped session token for the new restaurant. The diner's cross-restaurant preferences are applied. No new OTP is required within the 30-day device recognition window. The new restaurant does not receive any data about the diner's history at other restaurants — only the diner's dietary preferences and aggregate taste profile signals are used for recommendation ranking.

Flow 3: Operator Marks Item 86'd → Propagation to Live Menus → Diner Session Update

Happy Path:

1. Operator opens dashboard on mobile browser. Taps "86 an Item" (accessible from home screen).
2. Operator searches or scrolls to item. Taps item. Taps "Mark as 86'd." Confirmation modal: "Remove [Item Name] from the menu? Diners currently viewing the menu will see it disappear." Operator confirms.
3. EightySixEvent record created with: item_id, location_id, operator_user_id, timestamp, restore_date (null = next business day auto-restore).
4. Event published to SSE channel for this location_id.
5. All active diner browser sessions subscribed to this location's SSE channel receive the event. JavaScript removes the item from the DOM without page reload.
6. Operator receives confirmation: "Item removed from [X] active sessions."

Error States:

- SSE channel unavailable: System falls back to a 15-second polling interval for active sessions. Operator is not notified of the fallback — it is transparent. The 30-second propagation SLA still applies.
- Operator loses connectivity mid-action: EightySixEvent is not created. Operator sees a network error toast. No partial state is written.
- Item already 86'd: System returns a 409 Conflict with message: "This item is already marked as unavailable."

Diner Session Update Edge Case: A diner has the item in their "cart" (add-to-order state) when the 86 event arrives. The item is removed from the menu display and a toast notification appears: "Sorry, [Item Name] just became unavailable. It's been removed from your order." The diner's order total is updated.

Flow 4: Operator Enables Dynamic Pricing on Item → Compliance Check → Price Display to Diner

Happy Path:

1. Operator navigates to "Pricing" tab. Views slow-mover list for current day-part.
2. Operator selects item. Sets discount percentage and time window. Taps "Activate."
3. System performs jurisdiction check on location address. Location is not in Tennessee. Location is not in New York. Pricing rule created. PricingRule record written with all parameters.
4. Diner menu displays item with struck-through original price, discounted price in green, and "Today's Special" label.
5. At the end of the time window, price automatically reverts to original. PricingRule status updated to `expired`.

Tennessee Block Path:

- Operator taps "Activate." System checks jurisdiction. Location is Tennessee.
- Activation blocked. Error displayed: "Dynamic pricing is not available at this location (Tennessee SB 1807, effective July 1, 2026). This feature is disabled for all Tennessee locations. Contact support@[platform].com for more information."
- No PricingRule record is created. No partial state written.

New York Disclosure Path:

- Operator taps "Activate." System checks jurisdiction. Location is New York. System checks `Location.disclosure_acknowledgment_id`. Field is null.

- Activation blocked with error code `disclosure_acknowledgment_required`. Operator is directed to complete the NY disclosure acknowledgment flow.
- Operator navigates to `POST /v1/operator/locations/{location_id}/pricing-rules/disclosure-acknowledgment`. Reviews disclosure text. Submits `{"acknowledged": true}`. System creates a `DisclosureAcknowledgment` record and returns `disclosure_acknowledgment_id`. This ID is stored on the Location record.
- Operator returns to pricing rule creation. Submits activation request including `disclosure_acknowledgment_id`. Rule created. Diner menu displays item with NY-compliant disclosure statement: "Prices on this menu may be personalized based on your dining history."

POS Integration Unavailable:

- Operator activates pricing rule. POS sync for price update fails (Toast/Square API timeout).
- Platform displays the discounted price to diners via its own menu display. A warning banner appears in the operator dashboard: "Price update could not be synced to your POS. Orders placed at this price may require manual adjustment at checkout."
- The pricing rule remains active on the platform. The operator is responsible for reconciling the POS price discrepancy.

Data Model

Core Entities

Organization `id` UUID PK | `name` VARCHAR(255) NOT NULL | `billing_email` VARCHAR(255) NOT NULL | `stripe_customer_id` VARCHAR(255) | `created_at` TIMESTAMPTZ | `deleted_at` TIMESTAMPTZ (soft delete) *Classification: Internal business data. Not PII.*

Chain (optional grouping within Organization) `id` UUID PK | `organization_id` UUID FK → Organization NOT NULL | `name` VARCHAR(255) NOT NULL | `created_at` TIMESTAMPTZ *Classification: Internal business data.*

Restaurant (Location) `id` UUID PK | `organization_id` UUID FK → Organization NOT NULL | `chain_id` UUID FK → Chain NULLABLE | `name` VARCHAR(255) NOT NULL | `address_line1` VARCHAR(255) | `city` VARCHAR(100) | `state` VARCHAR(2) | `postal_code` VARCHAR(20) | `country` VARCHAR(2) DEFAULT 'US' | `timezone` VARCHAR(50) NOT NULL | `jurisdiction_flags` JSONB (computed: `{tn_sb1807_blocked: bool, ny_disclosure_required: bool}`) | `disclosure_acknowledgment_id` UUID NULLABLE FK → DisclosureAcknowledgment | `active` BOOLEAN DEFAULT true | `created_at` TIMESTAMPTZ *Index: organization_id, state Classification: Business data. Address is not PII in B2B context.*

OperatorUser `id` UUID PK | `organization_id` UUID FK → Organization NOT NULL | `email` VARCHAR(255) UNIQUE NOT NULL | `name` VARCHAR(255) | `role` ENUM('org_admin', 'location_manager', 'read_only') NOT NULL | `location_id` UUID FK → Restaurant NULLABLE (null = org-wide access) | `password_hash` VARCHAR(255) | `last_login_at` TIMESTAMPTZ | `created_at` TIMESTAMPTZ | `deleted_at` TIMESTAMPTZ *Index: organization_id, email Classification: PII (email, name). Encrypted at rest.*

Menu `id` UUID PK | `restaurant_id` UUID FK → Restaurant NOT NULL | `name` VARCHAR(255) NOT NULL | `is_active` BOOLEAN DEFAULT true | `template_source_id` UUID NULLABLE (FK → Menu, for chain-pushed templates) | `created_at` TIMESTAMPTZ | `updated_at` TIMESTAMPTZ

MenuItem `id` UUID PK | `menu_id` UUID FK → Menu NOT NULL | `name` VARCHAR(255) NOT NULL | `description` TEXT | `price` NUMERIC(10,2) NOT NULL | `image_url` VARCHAR(500) | `category` VARCHAR(100) | `is_archived` BOOLEAN DEFAULT false | `dietary_tags` TEXT[] | `allergen_flags` TEXT[] | `display_order` INTEGER | `created_at` TIMESTAMPTZ | `updated_at` TIMESTAMPTZ *Index: menu_id,*

is_archived Classification: `dietary_tags` *and* `allergen_flags` *on a MenuItem are business data describing the item, not the diner. Not special-category data. However, the combination of a DinerProfile's* `dietary_restrictions` *with their order history may constitute health-adjacent inference data — see DinerProfile classification note.*

DinerProfile `id` UUID PK | `identity_type` ENUM('phone', 'email') NOT NULL | `identity_hash` VARCHAR(255) UNIQUE NOT NULL (SHA-256 of normalized phone/email, salted) | `identity_verified_at` TIMESTAMPTZ | `dietary_restrictions` TEXT[] | `allergy_notes` TEXT NULLABLE | `created_at` TIMESTAMPTZ | `updated_at` TIMESTAMPTZ | `deleted_at` TIMESTAMPTZ (soft delete; hard delete scheduled 30 days after deletion request) *Index: identity_hash (unique) Classification:* ***PII and potentially GDPR special-category-adjacent.*** *Dietary restrictions referencing religious observance (halal, kosher) or medical conditions (gluten-free, nut allergy) may qualify as health or religious belief data under GDPR Article 9. Legal review required before launch. Stored encrypted at rest. Identity hash must not be reversible without the salt. Tenant isolation: DinerProfile is* ***diner-owned, not restaurant-owned.*** *No restaurant_id FK on this table. Restaurants access diner preference signals only through the recommendation API, which returns ranked item IDs — never raw DinerProfile records.*

DinerRestaurantInteraction (cross-restaurant linkage table) `id` UUID PK | `diner_profile_id` UUID FK → DinerProfile NOT NULL | `restaurant_id` UUID FK → Restaurant NOT NULL | `first_scan_at` TIMESTAMPTZ | `last_scan_at` TIMESTAMPTZ | `scan_count` INTEGER DEFAULT 0 | `consent_record_id` UUID FK → ConsentRecord NOT NULL *Index: diner_profile_id, restaurant_id (composite unique) This table enables cross-restaurant profile reuse while maintaining the architectural invariant that a restaurant queries only its own interaction rows — it never touches another restaurant's rows.*

ConsentRecord `id` UUID PK | `diner_profile_id` UUID FK → DinerProfile NULLABLE (null if consent declined — no profile created) | `restaurant_id` UUID FK → Restaurant NOT NULL | `consent_version` VARCHAR(20) NOT NULL | `accepted` BOOLEAN NOT NULL | `ip_address` INET NOT NULL (server-captured only, never from client body) | `user_agent` TEXT | `created_at` TIMESTAMPTZ *Index: diner_profile_id, restaurant_id Classification: Legal record. Immutable after creation. Retained for 7 years.*

Order `id` UUID PK | `diner_profile_id` UUID FK → DinerProfile NULLABLE (null for anonymous orders) | `restaurant_id` UUID FK → Restaurant NOT NULL | `items` JSONB (array of {menu_item_id, quantity, price_at_order_time}) | `total` NUMERIC(10,2) | `source` ENUM('qr_menu', 'pos_sync') | `created_at` TIMESTAMPTZ *Index: diner_profile_id, restaurant_id, created_at*

Rating `id` UUID PK | `diner_profile_id` UUID FK → DinerProfile NOT NULL | `menu_item_id` UUID FK → MenuItem NOT NULL | `restaurant_id` UUID FK → Restaurant NOT NULL | `score` SMALLINT CHECK (score BETWEEN 1 AND 5) | `created_at` TIMESTAMPTZ *Index: diner_profile_id, menu_item_id*

EightySixEvent `id` UUID PK | `menu_item_id` UUID FK → MenuItem NOT NULL | `restaurant_id` UUID FK → Restaurant NOT NULL | `operator_user_id` UUID FK → OperatorUser NOT NULL | `created_at` TIMESTAMPTZ NOT NULL | `restore_at` TIMESTAMPTZ NULLABLE | `restored_at` TIMESTAMPTZ NULLABLE | `status` ENUM('active', 'restored') DEFAULT 'active' *Index: restaurant_id, status, menu_item_id*

PricingRule `id` UUID PK | `menu_item_id` UUID FK → MenuItem NOT NULL | `restaurant_id` UUID FK → Restaurant NOT NULL | `created_by` UUID FK → OperatorUser NOT NULL | `discount_percentage` NUMERIC(5,2) NOT NULL | `original_price` NUMERIC(10,2) NOT NULL | `discounted_price` NUMERIC(10,2) NOT NULL | `active_from` TIMESTAMPTZ NOT NULL | `active_until` TIMESTAMPTZ NOT NULL | `days_of_week` INTEGER[] (0=Sun, 6=Sat) | `status` ENUM('pending', 'active', 'expired', 'cancelled') | `jurisdiction` VARCHAR(2) | `disclosure_acknowledgment_id` UUID NULLABLE FK → DisclosureAcknowledgment | `created_at` TIMESTAMPTZ | `audit_log` JSONB (append-only log of status changes) *Index: restaurant_id, status, active_from, active_until*

DisclosureAcknowledgment `id` UUID PK | `restaurant_id` UUID FK → Restaurant NOT NULL | `operator_user_id` UUID FK → OperatorUser NOT NULL | `jurisdiction` VARCHAR(2) NOT NULL |

`disclosure_text_version` VARCHAR(20) NOT NULL | `acknowledged_at` TIMESTAMPTZ NOT NULL | `ip_address` INET NOT NULL (server-captured)

AnalyticsSnapshot `id` UUID PK | `restaurant_id` UUID FK → Restaurant NOT NULL | `menu_item_id` UUID FK → MenuItem NOT NULL | `snapshot_date` DATE NOT NULL | `day_part` ENUM('breakfast', 'lunch', 'dinner', 'late_night') | `impression_count` INTEGER | `add_to_order_count` INTEGER | `order_count` INTEGER | `eighty_six_count` INTEGER | `avg_rating` NUMERIC(3,2) NULLABLE *Index: restaurant_id, snapshot_date, menu_item_id Classification: Aggregated, anonymized. No PII. Retained indefinitely.*

API Design

Authentication Scheme:

- Operators: JWT Bearer token, 8-hour inactivity expiry, issued via `POST /v1/auth/operator/token`
- Diners: Short-lived session token (4-hour TTL), issued after OTP verification via `POST /v1/auth/diner/verify-otp`. Separate device recognition token (30-day TTL, diner-scoped, stored in client localStorage), issued alongside session token.
- Public menu endpoint: No auth required. Rate-limited by IP.

Rate Limits: Public menu endpoint: 60 requests/minute per IP. Authenticated operator endpoints: 300 requests/minute per token. Diner OTP request: 3 requests per phone/email per 15-minute window.

Menu Retrieval

`GET /v1/menus/{location_id}` Auth: None (public) | Optional: `Authorization: Bearer {diner_session_token}` Query params: `table_id` (optional) Response 200:

```
{
  "location_id": "uuid",
  "menu_id": "uuid",
  "restaurant_name": "string",
  "categories": [{"name": "string", "items": [{...}]}],
  "recommended_item_ids": ["uuid"],
  "active_pricing_rules": [{"item_id": "uuid", "discounted_price": 0.00, "original_price": 0.00}
  "ny_disclosure_active": false
}
```

Errors: 404 (location not found), 429 (rate limit), 503 (menu service unavailable — returns cached menu with `stale: true` flag)

Profile & Consent

`POST /v1/auth/diner/request-otp` Body: `{"identity_type": "phone|email", "identity_value": "string"}` Response 200: `{"otp_token": "string", "expires_in": 600}` Errors: 422 (invalid format), 429 (rate limit exceeded — 3 attempts per 15 min)

`POST /v1/auth/diner/verify-otp` Body: `{"otp_token": "string", "otp_code": "string", "location_id": "uuid"}` Response 200: `{"session_token": "string", "device_recognition_token":`

`"string", "profile_exists": bool}` Errors: 401 (invalid OTP), 410 (OTP expired), 423 (locked — too many attempts)

`POST /v1/consent` Auth: None (server captures IP from request context) Body: `{"identity_type": "phone|email", "identity_hash": "string", "restaurant_id": "uuid", "accepted": bool, "consent_version": "string"}` *Note: `ip_address` is NOT accepted in the request body. It is captured server-side from `X-Forwarded-For` after proxy trust validation.* Response 201: `{"consent_record_id": "uuid"}` Errors: 422 (missing required fields), 409 (consent record already exists for this identity + restaurant)

`GET /v1/diner/profile` Auth: Diner session token required Response 200: `{"dietary_restrictions": [], "allergy_notes": "string", "restaurants_visited_count": int}` Errors: 401 (invalid token), 404 (profile not found)

`PATCH /v1/diner/profile` Auth: Diner session token required Body: `{"dietary_restrictions": [], "allergy_notes": "string"}` Response 200: updated profile Errors: 401, 422 (invalid dietary tag value)

`DELETE /v1/diner/profile` Auth: Diner session token required Response 202: `{"deletion_scheduled_at": "ISO8601", "completes_by": "ISO8601 (+30 days)"}` Errors: 401, 409 (deletion already in progress)

Operator: 86 Management

`POST /v1/operator/locations/{location_id}/eighty-six` Auth: Operator JWT (location_manager or org_admin) Body: `{"menu_item_id": "uuid", "restore_at": "ISO8601|null"}` Response 201: `{"eighty_six_event_id": "uuid", "propagation_channel": "sse"}` Errors: 401, 403 (insufficient role), 404 (item not found), 409 (item already 86'd), 422 (menu_item_id not in this location)

`DELETE /v1/operator/locations/{location_id}/eighty-six/{event_id}` Auth: Operator JWT Response 200: `{"restored_at": "ISO8601"}` Errors: 401, 403, 404

Operator: Dynamic Pricing

`POST /v1/operator/locations/{location_id}/pricing-rules` Auth: Operator JWT (location_manager or org_admin) Body: `{"menu_item_id": "uuid", "discount_percentage": 15.0, "active_from": "ISO8601", "active_until": "ISO8601", "days_of_week": [1,2,3,4,5], "disclosure_acknowledgment_id": "uuid|null"}` Response 201: `{"pricing_rule_id": "uuid"}` Errors: 401, 403, 404, 422 (`tn_pricing_blocked` for TN locations, `disclosure_acknowledgment_required` for NY locations without acknowledgment), 409 (conflicting rule exists for item in this time window)

`POST /v1/operator/locations/{location_id}/pricing-rules/disclosure-acknowledgment` Auth: Operator JWT (org_admin only) Body: `{"acknowledged": true}` Response 201: `{"disclosure_acknowledgment_id": "uuid"}` Errors: 401, 403 (non-org_admin), 404 (location not found), 422 (location not in NY — acknowledgment not required)

POS Webhook (Inbound)

`POST /v1/webhooks/pos/stock` Auth: HMAC-SHA256 signature header (`X-POS-Signature`) Body (Toast format): `{"restaurantGuid": "string", "menuItemGuid": "string", "status": "OUT_OF_STOCK|IN_STOCK", "quantity": 0}` Response 200: `{"eighty_six_event_id": "uuid|null"}` Errors: 401 (invalid signature), 404 (restaurant GUID not mapped), 422 (unknown item GUID) *Square equivalent uses `variation_id` and `quantity` fields. The integration layer maps POS-native identifiers to platform MenuItem IDs.*

Analytics Export

`GET /v1/operator/locations/{location_id}/analytics` Auth: Operator JWT Query: `start_date`, `end_date`, `day_part` (optional), `format` (json|csv) Response 200: AnalyticsSnapshot array or CSV attachment Errors: 401, 403, 422 (date range > 365 days)

Architecture / System Design

Three Critical Architectural Decisions

Decision 1: Diner Identity Without App Install

Browser cookies and device fingerprinting are rejected as the primary identity mechanism for three reasons: (1) GDPR ePrivacy Directive requires consent for non-essential cookies, creating a consent-on-consent UX paradox; (2) iOS Safari's Intelligent Tracking Prevention (ITP) deletes localStorage after 7 days of non-interaction, making cross-restaurant persistence unreliable; (3) browser fingerprinting is legally indefensible under CCPA for the cross-restaurant profile use case.

The adopted mechanism: phone number or email address as identity anchor, verified via OTP on first use, with a 30-day device recognition token stored in localStorage as a UX optimization for returning diners. The device recognition token is diner-scoped (tied to the verified identity, not to a restaurant or device), so it can be used to retrieve the profile at any participating restaurant without a new OTP. The token's 30-day TTL is independent of the 4-hour session token TTL — these are two distinct token types serving different purposes. Identity hashes (SHA-256 with per-platform salt) are stored, not raw phone numbers or emails, to limit exposure in the event of a database breach.

Decision 2: Real-Time 86 Propagation to Active Browser Sessions

Server-Sent Events (SSE) is the selected mechanism over WebSockets for the following reasons: (1) 86 propagation is unidirectional (server to client), making WebSocket's bidirectional capability unnecessary overhead; (2) SSE reconnects automatically on connection drop, critical for mobile browsers that may briefly lose connectivity; (3) SSE works over standard HTTP/2, reducing infrastructure complexity vs. WebSocket upgrade negotiation; (4) SSE is supported natively in all target browsers (iOS Safari, Android Chrome) without a library.

Architecture: Each active diner session subscribes to an SSE channel keyed by `location_id`. When an EightySixEvent is created, the application server publishes the event to a Redis pub/sub channel for that `location_id`. An SSE gateway service (horizontally scalable) consumes from Redis and pushes to all subscribed client connections. Fallback: if SSE connection is unavailable (detected via EventSource `onerror`), the client falls back to 15-second polling against `GET /v1/menus/{location_id}/availability-delta`.

Decision 3: Multi-Tenant Data Isolation for Cross-Restaurant Profiles

The DinerProfile is architecturally diner-owned, not restaurant-owned. No `restaurant_id` foreign key exists on the DinerProfile table. Restaurants access diner preference signals only through the recommendation API, which accepts a `diner_session_token` and `location_id` and returns a ranked list of `menu_item_ids` — never raw DinerProfile records. The cross-restaurant linkage is mediated by the `DinerRestaurantInteraction` table, which is queried in the context of the requesting restaurant's `location_id` — a restaurant can only read its own interaction rows.

Row-level security (RLS) policies at the database layer enforce tenant isolation for all operator-facing queries. A query issued with `restaurant_id = A` cannot return rows where `restaurant_id = B`. This is enforced at the database level, not only at the application level.

POS Integration Layer

POS integration is a first-class architectural component, not a roadmap item. The integration layer is a dedicated service responsible for: (1) receiving inbound webhooks from Toast (`stock` and `menus` webhooks) and Square (catalog/inventory webhooks); (2) mapping POS-native item identifiers to platform MenuItem IDs via a `POSItemMapping` table; (3) translating POS stock events into EightySixEvents on the platform; (4) writing back price changes from PricingRules to the POS (best-effort, with operator dashboard warning on failure).

Toast integration uses the `stock` webhook for real-time 86 events and the `menus` webhook for menu sync on onboarding. Square integration uses the Catalog API for menu import and the Inventory API for stock updates. Both integrations authenticate via HMAC-SHA256 webhook signatures. The integration layer is stateless and horizontally scalable. POS API failures are logged, alerted, and surfaced to the operator dashboard — they do not cause menu display failures.

Edge Cases & Error Handling

Diner scans at restaurant with no POS integration: Menu is loaded from the platform's own menu data (entered or imported during onboarding). 86 management is manual (operator-initiated via dashboard). No stock webhook is expected. The menu displays correctly. A banner in the operator dashboard reads: "POS not connected. 86'd items must be updated manually." This is a supported steady state, not an error condition.

Diner declines consent: Anonymous menu is rendered immediately. No DinerProfile, no ConsentRecord, no session data beyond the browser session. The "Browse Without Saving" path must be as fast and complete as the personalized path — no degraded menu content, no missing items, no dark patterns that make the anonymous experience worse to coerce consent.

All items in a category are 86'd simultaneously: The category header is hidden from the diner-facing menu. A single platform-level empty state message appears in the position where that category would have been: "We're refreshing this section — check back soon!" The category remains visible in the operator dashboard with all items shown in 86'd state, so the operator can restore them.

Dynamic pricing activated in a restricted jurisdiction: For Tennessee: hard block at the API layer before any PricingRule record is written. For New York without disclosure acknowledgment: 422 returned, no rule created. The jurisdiction check uses the `state` field on the Restaurant record. If the state field is null or malformed, activation is blocked by default with error `jurisdiction_unknown`. The operator dashboard displays the block reason with a link to documentation.

Diner profile conflict across restaurants (same email, different dietary preferences): The DinerProfile stores a single canonical set of dietary restrictions. When a diner updates preferences at Restaurant B, the update overwrites the profile (not creates a new one). The last-write-wins model is used. A future enhancement (Phase 2+) may allow per-restaurant preference overrides, but Phase 1 uses a single unified profile. The diner is informed: "Your preferences have been updated and will apply at all participating restaurants."

Menu loaded offline or on poor connectivity: The menu page uses a service worker to cache the last-loaded menu state. On subsequent loads with no connectivity, the cached menu is served with a visible banner: "You're offline. This menu may not show the latest availability." The consent flow and profile creation require connectivity and display: "An internet connection is required to personalize your menu. You can still browse the menu below."

Extremely large menu (>200 items): Menu renders with virtual scrolling. Category navigation anchors are rendered at top. Performance budget (P95 < 2s FCP) must be validated with a 200-item menu in load testing.

Concurrent 86 actions on the same item: If two operators attempt to 86 the same item simultaneously, the second request returns 409 Conflict. Idempotency is enforced at the database level via a unique constraint on `(menu_item_id, restaurant_id, status='active')` in the EightySixEvent table.

Diner session token used at wrong location: The session token is location-scoped. If a diner's session token for Location A is used in a request to Location B's API endpoints, the server returns 403 Forbidden. The client must initiate a new session (device recognition token flow) for Location B.

Dependencies & Risks

Risk 1: GDPR/CCPA Cross-Restaurant Profile Architecture

Likelihood: High | **Impact:** Critical | **Mitigation Required Before:** Phase 2 development start

The cross-restaurant DinerProfile is the platform's highest-value feature and its highest legal exposure. CCPA's definition of "sale" of personal information may encompass sharing a diner's profile across participating restaurants. GDPR Article 9 may classify dietary restriction data (halal, kosher, gluten-free for medical reasons) as special-category health or religious belief data requiring explicit consent under Article 9(2)(a), not merely the standard Article 6 lawful basis.

Required action: Legal architecture review by a privacy attorney with GDPR and CCPA expertise must be completed before any cross-restaurant profile data is collected in production. The review must specifically address: (1) whether the platform is a data controller or processor relative to each restaurant; (2) whether cross-restaurant profile sharing constitutes a CCPA "sale"; (3) whether dietary restriction data triggers Article 9 obligations; (4) the minimum consent language required for the two-interaction consent flow. This review is a hard prerequisite for Phase 2.

Risk 2: Tennessee SB 1807 and NY Algorithmic Pricing Disclosure Act

Likelihood: Certain (for affected jurisdictions) | **Impact:** High | **Mitigation:** Jurisdiction gating built into Phase 3

Tennessee SB 1807 is effective July 1, 2026. The dynamic pricing module must be jurisdiction-gated at the Restaurant.state field level before Phase 3 launch. The jurisdiction check is a hard block, not a soft warning. New York's disclosure requirement is a mandatory UX flow, not optional compliance language. Both must be implemented and tested before the dynamic pricing module is made available to any restaurant, regardless of their state.

Risk 3: POS Integration Dependency (Toast and Square)

Likelihood: High (integration complexity) | **Impact:** High (Phase 1 timeline) | **Mitigation:** Begin integration development in Week 1

Toast and Square integrations are Phase 1 launch requirements. The canonical failure mode for QR menu platforms (Mr Yum, Meandu) was building a front-end experience disconnected from POS, requiring manual order re-entry. That failure mode is explicitly avoided by treating POS integration as a first-class architectural component.

Toast's webhook API is well-documented with `stock` and `menus` webhooks available in production. Square's Catalog and Inventory APIs are mature but have a different authentication model (OAuth 2.0 per-location vs. Toast's partner API key). The primary schedule risk is Square's per-location OAuth flow, which requires each restaurant to individually authorize the integration — a friction point in multi-location onboarding that must be designed into the operator setup flow, not discovered during pilot deployment.

Required action: Assign one engineer exclusively to POS integration in Week 1 of Phase 1. Build a POS integration test harness with simulated Toast and Square webhook payloads before any restaurant goes live.

Risk 4: DOJ Hub-and-Spoke Antitrust Risk from Cross-Restaurant Pricing Data

Likelihood: Medium (at scale) | **Impact:** Critical | **Mitigation:** Architectural constraint enforced from day one

The RealPage DOJ enforcement action established the template for scrutinizing SaaS vendors that influence pricing across competing businesses in the same market. A platform that sets dynamic prices across hundreds of competing restaurants using shared data signals could attract identical scrutiny at scale.

Required action: The architectural constraint that each restaurant's PricingRule uses only that restaurant's own historical order data must be enforced at the code level, not only by policy. A dedicated security test must verify that the pricing recommendation algorithm cannot access order data from any restaurant other than the one making the request. This constraint must be documented in the engineering runbook and reviewed by legal counsel before the dynamic pricing module is launched.

Risk 5: Diner Identity Persistence on iOS Safari

Likelihood: High | **Impact:** Medium | **Mitigation:** OTP re-verification UX designed to be low-friction

iOS Safari's Intelligent Tracking Prevention deletes localStorage entries after 7 days of non-interaction with the domain. This means a diner who hasn't visited a participating restaurant in 7 days will lose their device recognition token and be prompted for OTP re-verification on their next scan. This is not a data loss event — the DinerProfile is server-side — but it is a UX friction point that may reduce return scan rates.

Mitigation: Design the OTP re-verification flow to complete in under 30 seconds (pre-filled phone number, single OTP entry field, auto-submit on 6-digit completion). Frame the re-verification as "Welcome back — confirm it's you" rather than "Your session expired." Track re-verification abandonment rate as a product metric; if it exceeds 30%, investigate progressive web app (PWA) installation as an optional enhancement.

Milestones / Phasing

Phase 1: Intelligent Static Menu (Days 0–90)

What's included:

- QR menu display engine (no-app browser, dietary/allergen filtering, WCAG 2.1 AA)
- Real-time 86 management with SSE propagation
- Operator dashboard (menu CRUD, 86 management, dish analytics)
- Toast and Square POS integration (menu sync and stock webhooks)
- Basic diner consent flow (accept/decline, anonymous browsing)
- DinerProfile creation with dietary restrictions (single-restaurant, no cross-restaurant linking)
- Organization, Chain, Location, OperatorUser data model and role-based access
- QR code generation and table card asset export

What's explicitly excluded from Phase 1:

- Cross-restaurant profile persistence and recommendation ranking
- Dynamic pricing module
- Multi-location chain analytics (cross-location aggregation)
- Device recognition token / skip-OTP returning diner flow

Why this sequence: Operator-side analytics value (dish view rates, 86-lag reduction, check average lift) is measurable within 30 days of deployment with zero diner profile data required. This allows the 60-day pilot to generate proof points before personalization is introduced. Building trust with operators before introducing the

cross-restaurant data architecture also reduces legal and reputational risk during the period when legal review of the consent architecture is in progress.

Phase 1 exit criteria: ≥3 pilot operators live, scan-to-order conversion rate ≥65% measured, 86-propagation latency <30s verified in production, legal review of cross-restaurant consent architecture initiated.

Phase 2: Diner Preference Engine (Days 90–180)

What's included:

- Cross-restaurant DinerProfile persistence (phone/email identity anchor)
- Device recognition token (30-day skip-OTP flow for returning diners)
- Recommendation ranking algorithm (order history + ratings + dietary compliance + cross-restaurant taste signals)
- "Recommended for You" menu section (max 5 items)
- Profile management UI (dietary restrictions, allergy notes, deletion request)
- Cross-restaurant profile reuse (diner scans at new participating restaurant, preferences auto-applied)
- Diner-facing rating UI (post-order item rating)
- Multi-location chain analytics dashboard (cross-location dish performance, anonymized/aggregated)

Prerequisites before Phase 2 development begins:

- Legal architecture review of cross-restaurant consent flow completed and signed off
- GDPR Article 9 analysis of dietary restriction data classification completed
- CCPA "sale" analysis of cross-restaurant profile sharing completed
- Consent flow language approved by legal counsel

Why this sequence: Personalization requires multiple return visits to generate a meaningful preference signal. A diner who scans once has no order history. The recommendation engine produces meaningful output only after 2–3 visits, which means Phase 2 value compounds over weeks, not days. Launching Phase 2 before Phase 1 has established a returning diner base would produce an empty recommendation engine with no proof points.

Phase 2 exit criteria: ≥150 active diner profiles per pilot location, recommendation acceptance rate ≥25%, cross-restaurant profile reuse rate ≥15%, return scan rate ≥35%.

Phase 3: Dynamic Pricing and Enterprise Chain Management (Days 180+)

What's included:

- Dynamic pricing module with slow-mover detection and discount surfacing
- Jurisdiction gating (Tennessee hard block, NY disclosure acknowledgment flow)
- DisclosureAcknowledgment entity and operator workflow
- Pricing rule audit trail
- Enterprise tier: Organization-level menu templates, chain-level push, dedicated CSM
- Self-service onboarding for single-location independents (Phase 2 sales motion unlock)
- API access tier for enterprise integrations

Prerequisites before Phase 3 development begins:

- DOJ hub-and-spoke antitrust risk review completed by legal counsel
- Jurisdiction gating logic reviewed and approved for all 50 states (not only TN and NY)

- Dynamic pricing module security test (cross-restaurant data isolation) passing

Why this sequence: The dynamic pricing module is the highest-revenue and highest-risk feature. Launching it after the platform has established operator trust, a diner profile base, and a clean legal architecture reduces the probability that a compliance incident during Phase 3 damages the relationships built in Phases 1 and 2.

Launch / Rollout Plan

60-Day Paid Pilot Program

Structure: 3–5 fast-casual multi-unit operators (3–10 locations each), each paying a reduced pilot fee of $49/location/month. Pilot operators are selected based on: (1) existing Toast or Square POS (reduces integration risk); (2) repeat customer base with ≥40% of covers being return visitors (ensures personalization signal can accumulate); (3) willingness to share POS order data for check average comparison.

Pre-agreed outcome metrics (defined in pilot contract):

- Check average lift ≥5% vs. 90-day pre-pilot baseline
- Scan-to-order conversion rate ≥60%
- 86-item lag time reduced from operator-reported baseline to <60 seconds (conservative target for pilot, <30s is the product SLA)
- Repeat scan rate ≥25% among profiled diners within 60 days

Pilot-to-paid conversion trigger: If ≥2 of 3 outcome metrics are met at the 60-day mark, the operator is presented with a standard contract at full pricing. If fewer than 2 metrics are met, the pilot is extended 30 days with a documented remediation plan.

QR Placement Protocol (Launch Requirement)

The same QR code placed on a sticker with no context achieves approximately 39% scan rates; placed on a clear table card with explanatory framing, the same code achieves 78% (EasyMenus, 2025). This is not a recommendation — it is a launch requirement.

Required placement: Printed table card (minimum 4" × 3"), not a sticker. Table card must include: (1) the QR code at minimum 1.5" × 1.5"; (2) the headline "Scan for your personalized menu"; (3) a one-line explanation: "See dishes matched to your taste. No app needed."; (4) the restaurant's logo. Sticker-only placement is not supported during Phase 1. Operators who cannot implement table cards during the pilot are not eligible for the pilot program.

Framing discipline: The platform must be presented to restaurant staff as "letting servers focus on hospitality" rather than "replacing servers." Operator onboarding materials must include a one-page staff briefing document with this framing. This is a documented finding from EasyMenus' 400-restaurant survey on QR menu failure modes.

Consent UX Review Gate

Before any diner data is collected in production — meaning before the first pilot restaurant goes live — the consent flow must pass a dedicated UX review that verifies:

- Consent is presented before any behavioral data is collected (not after menu load)
- The "decline" path is equally prominent as the "accept" path (no dark patterns)
- Consent language is plain-English, not legalese, and covers cross-restaurant data use explicitly
- The flow completes in ≤2 interactions and ≤60 seconds in usability testing with 5 representative users
- ConsentRecord is created server-side with server-captured IP before the menu is personalized

This review is a hard gate. No production diner data collection before it passes.

Feature Flag Strategy

All Phase 1 features are deployed behind a `restaurant_id`-scoped feature flag. New pilot restaurants are enabled one at a time, not in bulk. The dynamic pricing module (Phase 3) is deployed behind a feature flag that additionally checks `jurisdiction_approved: true` on the Location record — this flag is set manually by the operations team after jurisdiction review, not automatically.

Rollback trigger criteria:

- 86-propagation latency exceeds 30s P99 for >5 minutes in production → automatic rollback of SSE gateway to polling fallback
- Menu page P95 load time exceeds 4 seconds for >10 minutes → incident declared, engineering on-call paged
- Consent record creation failure rate exceeds 0.1% → feature flag for diner profile creation disabled immediately, anonymous-only mode activated

---

Analytics & Instrumentation

Operator Dashboard Events (visible to restaurant operators)

Event	Properties	Maps to Metric
`menu_item_viewed`	item_id, location_id, session_id, diner_profile_id (nullable), timestamp	Dish view rate
`menu_item_added_to_order`	item_id, location_id, session_id, diner_profile_id (nullable), price, is_recommended, timestamp	Add-to-order rate, recommendation acceptance rate
`eighty_six_event_created`	item_id, location_id, operator_user_id, timestamp	86-lag time (delta from this event to `menu_item_hidden` event)
`menu_item_hidden_from_session`	item_id, location_id, session_id, propagation_latency_ms, timestamp	86-propagation latency SLA
`dynamic_price_displayed`	item_id, location_id, original_price, discounted_price, session_id, timestamp	Dynamic price display rate
`dynamic_price_item_ordered`	item_id, location_id, price_paid, session_id, timestamp	Dynamic price acceptance rate

Internal Product Analytics Events (not exposed to operators)

Event	Properties	Maps to Metric
`qr_code_scanned`	location_id, table_id, timestamp, user_agent	Top of funnel
`consent_flow_started`	location_id, session_id, timestamp	Consent funnel step 1
`consent_accepted`	location_id, session_id, identity_type, timestamp	Consent acceptance rate

Event	Properties	Maps to Metric
`consent_declined`	location_id, session_id, timestamp	Consent decline rate
`otp_requested`	identity_type, timestamp (no PII in event payload)	OTP funnel
`otp_verified`	timestamp, is_new_profile, is_returning_device	Profile creation vs. retrieval split
`profile_created`	diner_profile_id (hashed), restaurant_count_at_creation, timestamp	Profile creation rate
`dietary_restriction_set`	restriction_type, diner_profile_id (hashed), timestamp	Profile completion rate
`recommendation_displayed`	item_id, diner_profile_id (hashed), rank_position, timestamp	Recommendation surface rate
`recommendation_accepted`	item_id, diner_profile_id (hashed), rank_position, timestamp	Recommendation acceptance rate
`cross_restaurant_profile_reused`	diner_profile_id (hashed), new_restaurant_id, profile_age_days, timestamp	Network effect metric
`device_recognition_token_used`	timestamp, token_age_days	Skip-OTP rate
`otp_reverification_abandoned`	timestamp, step_abandoned	Re-verification friction metric

Minimum event schema to prove personalization lift within 60-day pilot: The following four events are the minimum required to compute recommendation acceptance rate and check average lift:

1. `menu_item_viewed` with `is_recommended: bool`
2. `menu_item_added_to_order` with `is_recommended: bool` and `price`
3. `otp_verified` with `is_new_profile: bool` (to segment profiled vs. anonymous diners)
4. `consent_accepted` (to establish the profiled diner cohort denominator)

Check average lift is computed by comparing average `price` sum per session for sessions with `diner_profile_id` populated vs. sessions without, controlling for location and day-part. This calculation requires no additional events beyond the four listed above.

Tooling: All events are emitted to a single analytics ingestion endpoint. Internal product analytics are stored in a separate schema from operator-facing analytics, with no cross-contamination. Operator dashboard metrics are computed from the operator analytics schema only. No diner PII appears in any analytics event payload — diner_profile_id is always a one-way hash in event payloads.

Timeline Estimate

Phase 1: Days 0–90

Component	Estimate	Primary Risk
QR menu display engine (browser, dietary filters, WCAG)	3 weeks	WCAG AA compliance testing iteration
SSE gateway + 86 propagation	2 weeks	SSE reconnection edge cases on mobile
Operator dashboard (menu CRUD, 86 UI, basic analytics)	3 weeks	Mobile browser compatibility
Toast POS integration	2–3 weeks	**Primary schedule risk** — webhook HMAC setup and item GUID mapping
Square POS integration	3–4 weeks	Per-location OAuth flow adds onboarding friction
Basic consent flow + DinerProfile (single-restaurant)	2 weeks	Legal review of consent language (external dependency)
Data model, multi-tenant RLS, auth	2 weeks	RLS policy correctness verification
QR code generation + table card asset export	0.5 weeks	—
Load testing (200 concurrent sessions)	1 week	Must occur before first pilot goes live
Phase 1 total	**~12–14 weeks**	Toast/Square integration complexity is the critical path

Note: 12–14 weeks exceeds the 90-day target by 2–4 weeks if Toast and Square integrations are built in parallel by separate engineers. If built sequentially, Phase 1 extends to 16–18 weeks. Parallel POS integration development is required to hit the 90-day target.

Phase 2: Days 90–180

Component	Estimate	Prerequisite
Legal review of cross-restaurant consent architecture	3–4 weeks (external)	**Must complete before Phase 2 dev begins**
Cross-restaurant DinerProfile + device recognition token	3 weeks	Legal review complete
Recommendation ranking algorithm	2 weeks	DinerProfile data accumulation from Phase 1
"Recommended for You" UI + rating UI	1.5 weeks	—
Multi-location chain analytics dashboard	2 weeks	—
Phase 2 total	**~8–10 weeks** (after legal review)	Legal review is the critical path

Phase 3: Days 180+

Component	Estimate	Prerequisite
Antitrust and jurisdiction legal review	3–4 weeks (external)	Must complete before Phase 3 dev begins
Dynamic pricing module + jurisdiction gating	3 weeks	Legal review complete
NY disclosure acknowledgment flow	1 week	—
Enterprise tier (org-level templates, chain push)	3 weeks	—
Self-service onboarding for independents	2 weeks	—
Phase 3 total	**~9–11 weeks** (after legal review)	—

Overall critical path note: The two legal reviews — cross-restaurant consent architecture (Phase 2 gate) and antitrust/jurisdiction review (Phase 3 gate) — are external dependencies that cannot be compressed by adding engineering resources. Both reviews should be initiated as early as possible: the Phase 2 legal review should be commissioned at the start of Phase 1, and the Phase 3 legal review should be commissioned at the start of Phase 2. Treating these as sequential dependencies that begin only when the preceding phase ends will add 6–8 weeks to the overall timeline.
