TiffinBox

Product Requirements & Functional Walkthrough

MVP v1.0 · Mobile-First Web · Students & PG Residents

Students & PGs Primary Audience	**3 User Roles** User · Cook · Admin	**18 Screens** End-to-End Documented	**10 Sections** Full SRS + Edge Cases

TABLE OF CONTENTS

10 Key Success Metrics

SECTION 01 · PRODUCT OVERVIEW & VISION

Product Overview & Vision

TiffinBox is a hyperlocal subscription marketplace built for students, PG residents, and young working professionals who need daily home-cooked meals but lack a reliable, trustworthy source. The platform connects them with verified nearby home cooks who want stable monthly income. This document is the complete product specification — every screen, every interaction, every edge case — for the TiffinBox MVP.

The Core Problem

Urban students and young professionals face an unsolved daily problem: eating well, affordably, and consistently. Their current options are either expensive (Swiggy, restaurants), unhealthy (hostel mess), or completely unreliable (WhatsApp-based informal tiffin vendors with no accountability). There is no structured, trustworthy platform for daily home-cooked meal subscriptions.

Simultaneously, thousands of home cooks — mostly homemakers and small kitchen operators — supply food informally but struggle to find customers beyond their immediate social circle. They have no marketing reach, no payment tools, and no way to manage growing demand. TiffinBox bridges this gap.

Product Principles

- **Subscription first.** One-time orders are not the product. Daily recurring value is the product.
- **Trust before features.** Every design decision that builds trust outranks one that adds features.
- **Simplicity for cooks.** The cook-side app must feel as easy as WhatsApp. Zero learning curve.
- **Retention over acquisition.** A user staying 6 months is worth more than 6 users staying 1 month.
- **Mobile-first, always.** Primary audience is on phones. Every screen designed for mobile web first.

User Roles

END USER

Student / PG Resident / Working Professional
Discovers cooks, subscribes to meal plans, manages daily deliveries, raises issues, and rates food quality.

HOME COOK

Verified Supplier / Home Kitchen Operator
Manages menu, views daily orders, updates delivery status, and tracks earnings. Must be extremely simple.

ADMIN

Platform Operator (Internal)

Approves cooks, resolves disputes, manages payouts, monitors quality signals, controls platform settings.

MVP Scope

What's IN the MVP

User registration · Cook discovery · Cook profile with trust signals · Subscription checkout (3-step) · Daily delivery tracking (manual status) · Pause/skip with wallet credits · Ratings & feedback · Complaint handling · Cook dashboard · Menu management · Earnings & payout visibility · Admin approval & dispute resolution

■ Explicitly OUT of MVP (Phase 2): Real-time GPS tracking · In-app chat · Map view discovery · Multi-cuisine advanced filtering · Referral program · Backup cook auto-assignment · Owned kitchens

SECTION 02 · INFORMATION ARCHITECTURE

Information Architecture

The app has three completely separate flows — one per role. A single login determines which flow the user enters. There is no shared navigation between roles. Each role sees a different home screen, a different navigation structure, and a different visual theme.

User-Facing App

Public Zone (No login required)

- Landing Page — value prop, location capture, featured cooks
- Browse Cooks — location-gated discovery with filters
- Cook Profile Page — menu, pricing, trust signals, subscribe CTA
- How It Works — explainer page for new visitors
- Sign Up / OTP Login — phone number + 6-digit OTP, no passwords

Authenticated User Zone

- Home Dashboard — today's meal status, delivery tracker, quick actions
- Discover / Browse — same as public but with personalisation
- Subscribe Flow — 3-step checkout (plan → address → payment)
- My Subscriptions — all active and past subscriptions
- Pause & Skip Manager — calendar-based daily control
- Delivery Tracker — per-day status per subscription
- Feedback & Rating — post-delivery star rating flow
- Complaints & Support — issue raising and resolution tracker
- Wallet & Credits — pause credits, balance display
- Profile & Settings — address, notification prefs, account

Cook-Facing App

- Cook Onboarding — multi-step registration, verification, setup
- Cook Dashboard — today's orders, earnings summary, alerts
- Today's Orders — delivery list with one-tap status toggles
- Menu Manager — weekly menu CRUD with 'copy last week' shortcut
- Subscription Manager — view all subscribers, manage capacity
- Earnings & Payouts — balance, payout schedule, transaction history
- Availability Toggle — open/close for new subscriber requests

- Profile & Settings — kitchen info, delivery radius, cook photos

Admin Panel

- Cook Approvals Queue — review and approve or reject new applications
- User Management — view, support, or suspend users
- Complaints & Disputes — full resolution workflow with action tools
- Payout Management — monthly batch payout approval and tracking
- Quality Dashboard — rating trends, flagged cooks, delivery completion
- Analytics — GMV, retention, subscriber counts, churn

SECTION 03 · END USER FLOW — SCREEN BY SCREEN

End User Flow

All user-facing screens are designed for mobile web (375–430px viewport). Premium dark aesthetic with amber (#F5A623) as the primary accent. The experience should feel like a modern consumer product — not a food directory.

Screen 1 — Landing Page

Purpose

Convert first-time visitors into sign-ups. Must answer three questions in under 5 seconds: What is this? Is it near me? Can I trust it? This is the primary growth lever.

Layout & Visual Design

- **Hero section:** Full-screen dark background (#0D0D0D) with a warm amber radial glow behind the headline. Large display headline — e.g. 'Home food. Daily.' — in 2 lines. Subheadline: 'Subscribe to verified cooks near you.'
- **Primary CTA:** A prominent location search bar is the first and only interaction above the fold. Not a sign-up button. Location first, always.
- **Trust bar:** Directly below the search bar — '340+ verified cooks · Pause anytime · From Rs. 80/day' in small muted caps.
- **Cook preview strip:** Horizontally scrollable row of 3–4 cook cards with photo, name, cuisine, rating, price. Shows real supply exists.
- **How It Works:** 3-step section with icons — 1. Discover, 2. Subscribe, 3. Receive. Clean, icon-driven layout.
- **Testimonials:** 3 short quotes from students with first name, city, and a star rating.
- **Cook CTA:** 'Are you a home cook? Start earning monthly income →' at the very bottom. Drives supply acquisition.

Key Interactions

- Location input is the primary entry. User types locality or allows GPS. On submit — if service available → Browse Cooks. If not → waitlist email capture.
- Sign up link visible in top nav but deprioritised. Location discovery is the preferred first step to ensure contextual relevance.
- Returning users with an active session → auto-redirect to Dashboard. Do not show landing again.

Edge Cases

- User denies GPS permission → show manual input with locality autocomplete (Google Places API).
- No cooks in entered area → show waitlist screen with email capture and 'We're expanding to your area' message. Do not show a dead-end error.
- Entering a PIN code that maps to a serviceable area → auto-populate locality and proceed.

> **Design Principle**
>
> The location input is not a filter — it is the gateway. Nothing meaningful can happen without knowing where the user is. Make it impossible to miss. Centre it. Make it the largest interactive element on the page.

Screen 2 — Sign Up / OTP Login

Purpose

Get users into the app with zero friction. No passwords. No email verification. Phone number is the identity.

Flow

1. Single field: mobile number entry. Large, centred, dark background.
2. App sends 6-digit OTP via SMS. Auto-read on Android (via SMS Retriever API).
3. 6 large OTP input boxes. Auto-focus advances with each digit entered.
4. New users: 2-field onboarding — Name + Primary delivery address.
5. Redirect to Browse Cooks (pre-filtered by their address).

Edge Cases

- Wrong OTP 3 times → 10-minute lock with clear countdown timer and error message.
- OTP not received after 30 seconds → 'Resend OTP' button appears with countdown.
- WhatsApp OTP fallback → offer if SMS fails (Gupshup/Twilio WhatsApp Business API).
- Existing account detected → skip onboarding, restore session, redirect to Dashboard.

> ✓ Feasibility: OTP auth is fully feasible via Firebase Auth or MSG91. Standard 2-day integration. No complex infrastructure needed for MVP.

Screen 3 — Browse Cooks

Purpose

The discovery engine. Must feel like scrolling a curated, high-quality feed — not a cluttered food directory. Quality of curation matters more than quantity of results.

Layout

- **Sticky top bar:** Location name with a 'change' link. Search input for cook name or cuisine keyword.
- **Filter chips row:** Horizontally scrollable — All · Veg · Non-veg · Lunch · Dinner · Both · 4+ Stars · Under Rs. 100/day.
- **Sort dropdown:** Right-aligned — Nearest · Top Rated · Price: Low to High.
- **Cook card list:** Vertical scroll. Full-width cards on mobile. 4–6 cards visible before scroll.

Cook Card Design (each card contains):

- Cover image or food photo (16:9 ratio). Verified badge top-left. 'X spots left' urgency badge top-right (shown only when 3 or fewer spots remain).
- Cook name in bold and cuisine + diet type (e.g. 'North Indian · Veg').
- Star rating, review count, and distance from user's address (e.g. '1.2 km').
- Delivery timing summary (e.g. 'Lunch by 1 PM · Dinner by 8 PM').
- Monthly price prominently displayed (e.g. Rs. 2,800/mo). Per-day equivalent below in smaller text.
- 'View Plan' button — amber, full-width on mobile.

Logic & Behaviour

- Default sort: nearest first, within 5km of user's saved address.
- Cooks with 'Accepting new subscribers: Off' appear at bottom with 'Currently full' label — not hidden.
- New cooks (no ratings yet) show 'New Cook' badge instead of stars.

Edge Cases

- Zero results for filters → 'No cooks match your filters' with a 'Reset filters' link.
- User changes location to unserviced area → redirect to waitlist prompt.
- Cook account suspended → removed from results in real time.

Screen 4 — Cook Profile Page

Purpose

The most critical conversion page in the entire app. The user has shown intent — now the platform must convince them to commit. Every element is either a trust signal or a commitment trigger. This page is essentially a personal landing page for each cook.

Hero Section

- Large cook photo (full-width, 1:1 on mobile) with dark gradient overlay at the bottom.
- Cook name in large bold type overlaid on the photo. Verified badge beside the name.
- 4 stat pills below photo: Rating · Reviews · Active Subscribers · On-time delivery %.
- Cuisine and diet tags: e.g. 'North Indian' · 'Veg' · 'Lunch & Dinner.'

Trust Signals Strip

A dedicated strip — rendered slightly lighter than the background — with four trust signals shown as icon + text pairs:

- ID Verified — government ID confirmed by admin during onboarding review.
- Kitchen Hygiene Check — date of last check (e.g. 'Checked Jan 2025').
- Response Rate — 'Replies within 2 hrs' or '98% response rate.'
- Member Since — 'Cooking on TiffinBox since Aug 2023.'

This Week's Menu

A horizontally scrollable weekly calendar (Mon–Sun). Each day column shows two cells: Lunch dish and Dinner dish. Days with no service show 'Off' in muted text. This is the most important content element on the profile — users decide based on what they'll eat.

Subscription Plans

- Two plan cards side by side (stacked on mobile): Weekly Plan and Monthly Plan. Monthly plan has a 'Best Value' badge and shows per-day savings vs weekly.
- Below plans: Subscriber protection promise — 'If your cook misses a delivery, you get a wallet credit or backup meal within 2 hours.' This directly addresses the #1 user concern.

Reviews Section

- Latest 5 reviews in compact cards: star rating, reviewer name + initial, date, text. 'Load more' expands list. Sorted newest first.

Delivery Info

- Delivery radius in plain text: 'Delivers within 3 km of Koramangala 4th Block.'
- Delivery timing: 'Lunch by 1:00 PM · Dinner by 8:00 PM.'
- Packaging: cook-filled field from onboarding (e.g. 'Leak-proof steel containers').

Sticky Bottom Bar

Always visible while on the cook profile. Left: plan name + price (auto-updates as user scrolls past different plan cards). Right: large amber 'Subscribe Now' button. This is the single most important CTA in the app.

Edge Cases

- No reviews yet → 'Be the first to try and review this cook.'
- Cook at capacity → sticky bar shows 'Join Waitlist' instead of 'Subscribe.'
- Menu not updated this week → 'Sunita updates her menu every Sunday. Check back soon.'
- Cook on temporary leave → amber banner: 'Sunita is on leave until March 5. Subscribe to start after that date.'

> **Design Principle**
>
> The cook profile must feel like a personal brand page — not a database listing. Show personality. Show real food photos. Use the cook's name throughout. The user should feel like they know who is cooking for them before they commit.

Screen 5 — Subscribe Flow (3 Steps)

Purpose

Convert profile interest into a paid subscription with the least possible friction. Three steps, no detours, no distractions. Progress indicator always visible.

Step 1 — Choose Your Plan

- Step indicator at top: 3-step breadcrumb with current step filled in amber.
- Meal type selector: Lunch / Dinner / Both — visual cards with icon and price change on selection.
- Duration selector: Weekly / Monthly — Monthly pre-selected. Savings highlighted on monthly option.
- Start date: Tomorrow / Next Monday / Pick a date — smart defaults based on current day of week.
- Pause policy callout: 'You can pause up to 4 days/month. Skipped days become wallet credits.'

Step 2 — Confirm Delivery Address

- Show user's saved primary address with a map pin thumbnail preview.
- Option to add a new address or edit the existing one.
- Real-time radius validation: Green '✓ Within Sunita's delivery radius' or Red '✗ Outside delivery area.'
- Delivery instructions field (optional): gate code, landmark, flat number.

Step 3 — Payment & Confirmation

- Clean order summary: cook name, plan type, duration, start date, price breakdown.
- Payment options: UPI AutoPay (default, show app icons), Debit/Credit Card, Wallet.
- Auto-renewal toggle: On by default, with label 'Auto-renews on [date]. Cancel anytime.'
- 'Subscribe & Pay' — large amber button, full-width.

Post-Subscription Confirmation Screen

- Subtle success animation — a single amber checkmark that pulses once. Not a confetti explosion.
- Summary text: 'You're subscribed to Sunita's lunch plan. First meal arrives tomorrow by 1:00 PM.'
- Add to Calendar button (creates recurring daily event).
- 'Go to Dashboard' as primary CTA.

Edge Cases

- Payment fails → clear error message, retry with different method. Do not lose plan selection state.
- Cook fills up between profile view and payment → 'This cook just filled up. You've been added to the waitlist.'
- Address outside radius → block Step 3, prompt user to update address or return to browse.
- User abandons at Step 2 → save progress in session. Resume within 24 hours returns to Step 2.

> ✓ Feasibility: Razorpay Subscriptions API handles recurring billing, UPI AutoPay mandates, and auto-renewal natively. Well-documented, production-ready for Indian market.

Screen 6 — User Dashboard (Home)

Purpose

The daily-use home screen. A user opens this app at 12:45 PM to check if their food is on the way. This screen must answer that question within 1 second of opening, then surface every useful action within one tap.

Today's Meal Card (Hero Element)

- 'Today's Lunch' label in muted caps at the top.
- Cook avatar (circular photo), cook name, today's dish name in large bold type.
- Expected delivery time shown prominently (e.g. 'Expected by 1:00 PM').
- 4-step delivery status bar: Confirmed → Preparing → Out for Delivery → Delivered. Current step highlighted in amber with a soft pulse animation.
- Three quick action buttons: 'Skip Tomorrow' · 'Rate Yesterday' · 'Report Issue.'

This Week Preview Strip

Horizontally scrollable strip showing next 7 days as compact day-cards. Each shows day name, truncated meal name, and a green dot (active) or grey X (paused/skipped). Tapping a day opens the Pause Manager.

Quick Action Grid

2x2 grid of tappable cards: Pause Days · Find More Cooks · View Wallet Credits · Get Help.

Edge Cases

- Day skipped → hero card shows 'You skipped today. Rs. 93 credit added to your wallet.'
- No active subscription → show 'Find your first cook →' onboarding prompt as the hero element.
- Cook missed delivery and hasn't updated status → amber alert card 'Your meal hasn't been marked delivered. Report an issue.'
- Renewal due in 3 days → banner 'Your plan renews on Feb 28 — Rs. 2,800 will be charged automatically.'

Screen 7 — Pause & Skip Manager

Purpose

Pausing is churn prevention. Users who can pause instead of cancel stay significantly longer. This screen must make pausing feel instant, transparent, and consequence-free.

Layout

- Credits summary card: 'You have Rs. 186 in pause credits this month (2 days used, 2 remaining).'
- Full monthly calendar view with colour-coded days: Green = active, Amber = skipped (credit earned), Grey = past date (not tappable), White with border = future tappable day.
- Tap any future day to toggle skip/active. First skip shows a confirmation tooltip explaining the credit system.
- 'Pause a range' button → date-range picker for bulk pausing (e.g. travel).

Pause Policy (Displayed On Screen)

- Maximum 4 skip days per monthly billing cycle.
- Skips must be made before 10:00 PM the night before.
- Skipped days earn wallet credits equal to that day's pro-rated amount.
- Credits automatically apply to next billing cycle — no manual redemption.

Edge Cases

- Attempt to skip more than 4 days → 'You've used all your pause days this month. Need to cancel? Visit subscription settings.' Never block cancellation.
- Skip requested after 10 PM cutoff → 'Too late to skip tomorrow. You can skip the day after.'
- Cook already confirmed delivery for tomorrow → skip blocked, prompt to contact support for exceptions.

Screen 8 — Delivery Tracker

Purpose

Answer the question every subscriber asks between 12:30 and 1:30 PM: 'Where is my food?' Simple, manual status without requiring GPS hardware from the cook.

Status States & Layout

- **Confirmed** — subscription active, meal planned for today. Set automatically at midnight.
- **Preparing** — cook taps this by 11 AM in their app to signal they've started.
- **Out for Delivery** — cook taps this when they leave for deliveries.
- **Delivered** — cook marks delivered per subscriber. User gets a push notification immediately.
- **Issue Flagged** — either user or cook has raised a problem. Admin auto-notified.

Vertical timeline with 4 steps. Completed steps filled amber. Current step pulses softly. Timestamp shown next to completed steps. If delayed 15+ minutes past expected time → status bar turns amber with 'Running late — cook has been notified.'

Post-Delivery

After 'Delivered' is set, screen transitions to rating prompt. 5 large star buttons. Optional tag chips. User can dismiss and rate later. Never block them.

Screen 9 — Feedback & Rating

Purpose

Collect quality signals without annoying users. Every rating improves the platform's ability to surface good cooks and identify bad ones. One tap to submit, optional to elaborate.

Rating Flow

1. Push notification sent 20 minutes after cook marks delivery complete.
2. 5 large star buttons, centred on screen. Single tap to select.
3. If 4–5 stars: Positive tag chips — 'Loved it' · 'Great portion' · 'On time' · 'Good packaging.' Multiple selectable.
4. If 1–3 stars: Issue chips — 'Late delivery' · 'Cold food' · 'Less quantity' · 'Taste issue' · 'Packaging problem.' Selecting these creates a soft admin escalation.
5. Optional text comment field. Submit → thank you screen with meal streak counter.

Edge Cases

- No rating within 48 hours → prompt expires. Shown as 'unrated' in admin analytics for tracking.
- User rates 1 star for same cook 3 times in a row → automatic admin flag for review.
- Cook disputes a 1-star review → admin reviews, can remove if found fake. Both parties notified.

Screen 10 — Complaints & Support

Purpose

When something goes wrong, users need to feel heard and see action. Slow complaint resolution is the #1 cause of churn. This screen exists to prevent that.

Issue Categories

- Meal not delivered
- Meal arrived very late (30+ min past expected)
- Food quality issue (taste, hygiene, packaging)
- Wrong meal delivered
- Quantity was insufficient
- Billing or payment issue
- Cook is unresponsive
- Other

Flow

1. User selects issue category.
2. Brief description field (optional). Photo attachment (optional).
3. Submit → ticket created with reference number and expected resolution time.
4. Admin reviews, takes action (credit, refund, warning, or cook removal).
5. User notified of resolution via push + in-app message with reference number.

Resolution Policies (Visible to User)

- Missed delivery → full day credit to wallet. Automatic.
- Late delivery (30+ min) → 50% credit for that day.
- Hygiene complaint → immediate admin investigation. Cook flagged and reviewed.
- Billing dispute → resolved within 24 hours with full transaction evidence.

> **Design Principle**
>
> The complaints screen is not a dead end. Every issue must show a visible resolution path and a timeframe. Users must see a complaint status tracker — just like the delivery tracker for their food. 'We'll look into it' without a timeline destroys trust.

SECTION 04 · HOME COOK FLOW — SCREEN BY SCREEN

Home Cook Flow

The cook app must be the simplest interface in the product. Cooks are not always tech-savvy. Every screen should feel as familiar as a WhatsApp chat list — no learning curve required. The cook app uses a light theme (vs dark for users) to visually distinguish the two contexts.

Cook Screen 1 — Onboarding & Registration

Purpose

Sign up a new cook, collect all verification information, and configure their service before going live. This is a multi-step wizard. Cooks are never visible to users until admin approves.

Onboarding Steps

1. Mobile number + OTP verification (same mechanism as user login).
2. Personal details: Full name, profile photo (required), years of cooking experience.
3. Kitchen details: Home address, kitchen type (home/small commercial), FSSAI number if available (optional for MVP).
4. Cuisine & diet: Select cuisines offered and diet type (Veg / Non-veg / Both).
5. Meal types: Lunch / Dinner / Both. Set delivery timings for each.
6. Delivery radius: Set maximum distance (1 km / 2 km / 3 km / 5 km). Map preview shows coverage area.
7. Subscriber capacity: Maximum number you can serve (10 / 15 / 20 / 25 / Custom).
8. Pricing: Set monthly plan price for Lunch, Dinner, and Both. Platform shows market-rate suggestions.
9. Sample menu: Enter at least 3 days of weekly menu before submitting.
10. Bank account: Account number + IFSC code for Razorpay Route payouts. Stored encrypted.
11. Submit for admin review. Cook enters 'Pending Approval' state immediately.

Post-Submission

- Cook sees application submitted screen with expected review timeline: 24–48 hours.
- Admin receives notification of new cook application.
- On approval → cook receives SMS + push notification and is live immediately.
- On rejection → cook receives reason with specific items to fix before resubmitting.

> **Regulatory Note**
>
> FSSAI registration is not mandatory for home cooks under India's 2020 FBO guidelines for petty food businesses (turnover < Rs. 12L/year). Platform can require it voluntarily in Phase 2 for trust signalling, but it must not be a blocker for MVP supply acquisition.

Cook Screen 2 — Cook Dashboard

Purpose

The cook's command centre, opened every morning. Must show everything needed to fulfil today's orders without navigating anywhere else.

Layout

- Header: Date + Day name. Cook's name. Status toggle top-right: 'Open for orders' (green) / 'Closed' (grey). One tap.
- Today's delivery progress bar: 'X meals to deliver today' with '3 of 12 delivered' progress indicator.
- Earnings summary: Today's earnings / This month's earnings / Next payout date.
- Alert cards (highest priority at top): New subscriber joined · Complaint raised · Payout processed · Menu not updated this week.
- Quick nav tiles: Today's Orders · Menu · Subscribers · Earnings.

Edge Cases

- Cook hasn't updated delivery statuses by 3 PM → automated push notification reminder.
- Subscriber cancels → 'Priya cancelled her subscription. 1 spot is now available.' card on dashboard.
- Complaint raised → red alert card with complaint summary and 'View & Respond' button.

Cook Screen 3 — Today's Orders

Purpose

The most-used screen in the cook app. Opened every morning before deliveries begin. Must show the complete delivery list with one-tap status updates. Fast and simple above everything.

Layout

- Delivery progress bar at top: '8 of 12 delivered.'
- 'Mark all delivered' button — bulk action for when cook completes all deliveries at once.
- Ordered list of all active subscribers for today. Each row shows:
- ■ Subscriber name and delivery address (large, readable text).
- ■ Meal type (Lunch / Dinner).
- ■ 3-state status toggle: Preparing → Out for Delivery → Delivered. Active state highlighted.
- ■ Masked phone number (tap to call via intermediary — opens dialer).
- ■ 'Paused today' label in muted text for subscribers who skipped — no action required.

Edge Cases

- New subscriber added mid-cycle → appears in tomorrow's list automatically. Push notification to cook.
- Subscriber paused for today → shown as greyed-out row with 'Paused today' label.
- Cook marks delivered but user reports non-delivery → admin auto-notified. Cook receives a flag on their account.

> **Anti-Disintermediation**
>
> Phone number masking is critical. All contact between cook and user routes through a masked intermediary number (Twilio/Exotel). Cook and user never see each other's real numbers. This removes the simplest path to going off-platform.

Cook Screen 4 — Menu Manager

Purpose

Allow cooks to update their weekly menu with minimal effort. Fresh menus pushed to subscribers as push notifications increase engagement and reduce churn.

Layout

- Weekly grid: 7 columns (Mon–Sun), 2 rows (Lunch, Dinner). Each cell is a simple tap-to-edit text field.
- Tapping a cell opens a short text input: 'Enter dish name (e.g. Dal Makhani).' Single line.
- 'Copy last week' button — pre-fills all cells with the previous week's menu instantly.
- 'Same every day' shortcut — lets a cook set one dish that repeats across all days for a meal type.
- 'Publish Menu' button at bottom — disabled until at least 5 days are filled. On publish → subscribers notified.
- Status indicator: 'Updated 3 days ago' or 'Not updated this week — subscribers cannot see your menu.'

Edge Cases

- Menu not updated for 7+ days → automated reminder push + admin flag for review.
- Cook updates menu after subscribers have already seen it → subtle 'Menu updated' notification to subscribers, not intrusive.

Cook Screen 5 — Earnings & Payouts

Purpose

Financial transparency is the primary driver of cook retention. If cooks don't trust they'll get paid reliably and can see exactly what they're owed, they leave the platform. This screen must be as clear as a bank statement.

Layout

- Top card (dark, amber accent): Current escrow balance — amount earned and pending payout. Large number, impossible to miss.
- Next payout date and expected amount: 'Next payout: March 1 — Rs. 4,200.' Payouts on 1st of each month.
- Earnings toggle: This month / Last month / All time.
- Per-subscriber earnings table: name, plan amount, days delivered, amount earned from that subscriber.
- Transaction history: Chronological — each subscription payment received and each payout sent.
- Commission rate displayed transparently: 'TiffinBox takes X%. You keep Y%.' No hidden fees.

- 'Update bank account' link at the bottom.

Edge Cases

- Invalid bank account details → payout fails, cook notified immediately with instructions to update.
- Subscriber cancels mid-month → cook earns only for days delivered. Pro-rated amount shown clearly.
- Disputed amount → shown separately as 'On hold' with reason. Not included in payout until resolved.

SECTION 05 · ADMIN PANEL — FUNCTIONAL WALKTHROUGH

Admin Panel

The admin panel is a web-based internal tool. Not mobile-optimised. Used by 1–5 operators in early stage. Priority: operational efficiency over visual aesthetics.

Admin Screen 1 — Cook Approvals Queue

New cook applications arrive in a priority queue. Each application card shows cook name, submission date, and a completeness score (photo uploaded, bank details filled, menu entered, radius set). Admin clicks to open the full application, reviews every field, and either Approves (cook goes live immediately) or Rejects (mandatory rejection reason is sent to the cook by SMS).

Key review fields: Identity (photo match to name), kitchen photo if submitted, sample menu quality and realism, pricing within market range, delivery radius feasibility relative to kitchen address.

Admin Screen 2 — Complaint & Dispute Resolution

All unresolved complaints in a priority-sorted queue. High priority: missed deliveries (SLA breach after 4 hours), hygiene complaints, billing disputes. Each complaint card shows user name, cook name, issue type, date raised, and days open.

Admin opens a complaint and sees: full complaint thread, delivery status logs, timestamps, both sides' accounts of events. Available actions: Issue wallet credit · Issue partial/full refund · Send formal warning to cook · Flag cook for review · Remove cook from platform · Close as resolved · Escalate internally.

Admin Screen 3 — Payout Management

On the 1st of each month, admin initiates a payout run. A table shows all cooks with earned amounts, holds, and final payout amount. Admin reviews the batch, approves, and Razorpay Route initiates bank transfers. Individual cook payouts can be held pending dispute resolution, with the reason logged and visible to the cook.

Admin Screen 4 — Quality Dashboard

Real-time quality monitoring: average platform rating (rolling 30 days), cooks rated below 4.0 (red flag), cooks with 3+ unresolved complaints, delivery completion rate per cook, and subscription churn rate by cook. Outliers highlighted in red for immediate action. Admin can click any flagged cook to view their full profile and history.

Admin Screen 5 — Analytics Dashboard

Core metrics tracked in real time and as weekly/monthly trends:

- Total active subscribers and new subscriptions this week/month
- Churned subscriptions with cancellation reasons
- Gross Transaction Value (GTV) and platform commission revenue
- Average subscription duration (tracks retention trend)
- Cook count: active / pending / suspended
- Complaint rate per 100 deliveries (quality signal)
- Top cooks by subscriber count and by average rating

SECTION 06 · FEATURE FEASIBILITY & PRIORITY MATRIX

Feature Feasibility & Priority Matrix

Every feature in the MVP has been assessed for technical and operational feasibility. Priority tiers: P0 = must-have for launch, P1 = important but not blocking, P2 = Phase 2.

Feature	Feasibility	Priority	Notes
Location-based cook discovery	High	**P0**	*Geolocation API + radius DB query. Standard implementation.*
OTP login — no passwords	High	**P0**	*Firebase Auth or MSG91. 2-day integration.*
Cook profile with trust signals	High	**P0**	*All data collected at onboarding. Static display.*
Weekly menu display	High	**P0**	*Simple CRUD. Cook enters text, user sees table view.*
Subscription checkout (3-step)	High	**P0**	*Razorpay Subscriptions API. Well-documented.*
UPI AutoPay recurring mandate	High	**P0**	*Razorpay supports UPI AutoPay natively. India-critical.*
Manual delivery status updates	High	**P0**	*Cook taps status in app. No GPS required. Simple.*
Pause/skip with wallet credits	High	**P0**	*Calendar logic + wallet ledger. Straightforward.*
Cook earnings dashboard	High	**P0**	*Ledger table + Razorpay Route for payouts.*
Admin approval workflow	High	**P0**	*Internal CRUD admin panel. 2–3 day build.*
Push notifications	High	**P0**	*Firebase Cloud Messaging. Free tier sufficient for MVP.*
Post-delivery rating flow	High	**P1**	*Push notification trigger + 5-star UI. Standard.*
Complaint resolution flow	High	**P1**	*Ticketing system. Freshdesk or custom simple build.*
Phone number masking	Medium	**P1**	*Twilio or Exotel. ~Rs. 0.50/min cost. Acceptable.*
Real-time GPS tracking	Low	**P2**	*Requires live cook GPS. Infrastructure-heavy. Defer.*
In-app chat between parties	Medium	**P2**	*Adds complexity and disintermediation risk. Defer.*
Backup cook auto-assignment	Low	**P2**	*Requires supply density. Manual admin assignment in MVP.*
Map view on browse screen	Medium	**P2**	*Google Maps embed. Defer until list view is validated.*

Feature	Feasibility	Priority	Notes
Referral program	High	**P2**	*Coupon + credit logic. Phase 2 growth feature.*

SECTION 07 · ANTI-DISINTERMEDIATION STRATEGY

Anti-Disintermediation Strategy

Disintermediation — users and cooks transacting directly, bypassing the platform — is the single greatest existential risk to a marketplace. This section documents every design mechanism used to mitigate it. The core principle: never restrict. Instead, make the platform more valuable than going direct.

Mechanism 1 — Pause/Skip with Credits

If a user goes direct, they lose the ability to pause or skip days and receive credits. Managing this manually with a cook is awkward and error-prone. The platform makes this effortless. Credits have real monetary value that disappears the moment they transact off-platform.

Mechanism 2 — Subscriber Protection Guarantee

If the cook misses a delivery or a hygiene issue arises, the platform intervenes with a credit or backup meal arrangement. This guarantee disappears entirely if the user goes direct. It is a tangible, high-value benefit that a direct relationship cannot replicate.

Mechanism 3 — Phone Number Masking

Cook and user never see each other's real phone numbers. All calls and messages route through a masked intermediary (Twilio/Exotel). This removes the simplest and most direct path to off-platform contact. Cost: approximately Rs. 0.50–1.00 per call minute, acceptable at MVP scale.

Mechanism 4 — Delayed Payout Schedule

Cooks are paid on the 1st of each month for the previous month's earnings. If a cook takes a subscriber off-platform mid-month, they forfeit any unpaid earnings. This creates a meaningful financial lock-in for cooks and makes leaving costly.

Mechanism 5 — Cook Reputation Is Platform-Bound

As a cook builds their TiffinBox profile — ratings, reviews, subscriber count, badge tier — their customer acquisition capability is entirely tied to the platform. A cook with 4.9 stars and 20 subscribers leaving means starting from zero elsewhere. The platform is their business asset.

Mechanism 6 — Subscription Inertia

A user with a configured UPI mandate, saved address, established routine, and accumulated wallet credits has significant switching cost. Cancelling the subscription and renegotiating privately with the cook requires real effort. Most users will not bother — inertia is powerful.

Reality Check

Disintermediation cannot be fully prevented. Some user-cook pairs will transact directly regardless of design. The goal is to make this the exception, not the norm. If platform value is clearly high and friction is clearly low, the vast majority will stay on-platform. Focus energy on adding value, not building walls.

SECTION 08 · RECOMMENDED TECH STACK

Recommended Tech Stack

Optimised for a small team (2–4 engineers) building fast with production reliability. Opinionated choices over maximum flexibility.

Layer	Technology	Rationale
Frontend	Next.js (React)	*SSR for SEO on public pages. SPA navigation for app screens. Industry standard.*
Styling	Tailwind CSS	*Utility-first, fast to iterate, no design system overhead for MVP.*
Database	PostgreSQL	*Relational — subscriptions, orders, users, transactions all require relational integrity.*
Backend / BaaS	Supabase	*Postgres + Auth + Realtime + Storage in one. Dramatically faster for a small team.*
Authentication	Firebase Auth or Supabase Auth	*OTP via MSG91/Gupshup. Phone-number-first identity. Zero-password.*
Payments	Razorpay	*India-native. UPI AutoPay. Razorpay Route for automatic cook payouts. Not Stripe.*
Push Notifications	Firebase Cloud Messaging	*Web push support. Free tier sufficient for MVP scale.*
SMS	MSG91 or Gupshup	*OTP + delivery SMS. Both have strong India carrier coverage at low cost.*
Maps & Location	Google Maps Platform	*Geocoding API + Places API (autocomplete) + Distance Matrix (radius validation).*
Phone Masking	Exotel (India)	*Cheaper than Twilio for India-only deployments. Well-supported.*
Image Storage	Cloudinary	*Cook photos + food images. Free tier + built-in image optimisation.*
Frontend Hosting	Vercel	*Zero-config Next.js deployment. Global CDN. Free for early stage.*
Backend Hosting	Railway or Render	*Simple Postgres + Node.js hosting. Affordable. Scales when needed.*

SECTION 09 · PRE-LAUNCH & GO-LIVE SEQUENCE

Pre-Launch & Go-Live Sequence

The order in which you build and launch matters as much as what you build. This sequence is designed to avoid the most common early-stage marketplace failure: launching to users before supply is ready.

Phase 0 — Supply Seeding (Before Any User Acquisition)

> 🟥 Critical: Do not launch to users until you have at least 5–8 verified, high-quality cooks in your target locality. A user who finds only 1 cook will leave and never return. Supply density is everything.

- Manually recruit 5–8 home cooks in a single 3–5 km cluster (one neighbourhood).
- Run through the onboarding flow with each cook in person. Identify every friction point.
- Place actual test orders. Eat the food. Verify quality before going live.
- Do not build the platform until you have verbal commitments from at least 5 cooks.

Phase 1 — Soft Launch (Target: 100 Users)

- Invite-only access in the seeded cluster. No public ads.
- Manual operations as needed — admin managing exceptions via direct contact while platform is validated.
- Focus entirely on retention: are users subscribing beyond week 1? Beyond month 1?
- Collect qualitative feedback weekly from both users and cooks. Interview. Do not just track numbers.
- Success signal: 70% of Month 1 subscribers renew for Month 2.

Phase 2 — Product Polish & Geographic Expansion

- Fix all friction points identified in Phase 1 before expanding.
- Add backup cook manual assignment, referral mechanics, and rating-based cook tiering.
- Expand to adjacent clusters — 2–3 more neighbourhoods in the same city.
- Begin building the dataset needed to de-risk owned kitchen launch.

Phase 3 — Hybrid Owned Kitchens

- Use marketplace data (popular cuisines, peak demand timing, pricing tolerance) to plan the first owned kitchen.

- Run owned kitchen alongside marketplace. Compare margins, quality consistency, and subscriber NPS.
- Owned supply improves margins, quality control, and brand. But it increases capital, complexity, and regulatory requirements. Only enter Phase 3 when marketplace is profitable.

SECTION 10 · KEY SUCCESS METRICS

Key Success Metrics

MVP success is not about scale. It is about proving the retention hypothesis: if quality and reliability are solved, users stay. These are the metrics that tell that story.

> **North Star Metric**
>
> Monthly Recurring Subscribers — the count of users who have an active subscription that auto-renewed at least once. This is the single number that proves product-market fit. Everything else supports or explains this number.

Metric	Definition	MVP Target
Subscriber count	Total active subscribers	Grow to 50 by end of Month 2
Monthly retention	Subscribers who renew month-over-month	Target: >70%
Avg. subscription duration	Mean weeks active per subscriber	Target: >8 weeks by Month 3
Daily delivery completion	% of planned deliveries completed	Target: >95%
Platform rating	Average star rating across all deliveries	Target: >4.2 stars
Complaint rate	Complaints per 100 deliveries	Target: <5%

Cook Metric	Definition	MVP Target
Active cook count	Cooks with at least 1 active subscriber	Target: 8 by Week 2
Cook retention	Cooks active after 3 months	Target: >80%
Avg. subscribers per cook	Mean active subscribers per cook	Target: >10 by Month 2
Cook on-time rate	Deliveries marked on time by cook	Target: >95%

Financial Metric	Definition	Target
Gross Transaction Value	Total subscription value processed	Track weekly from Day 1
Platform commission revenue	GTV × commission rate	Target positive unit economics by Month 3
Customer Acquisition Cost	Total marketing spend / new subscribers	Track from first paid campaign

Financial Metric	Definition	Target
LTV : CAC ratio	Lifetime value vs acquisition cost	Target: >3x