



PRODUCT AND DEVELOPMENT REPORT · 14 OCTOBER 2026

Udyog Manager

Complete product report

Billing, customer records and business management across web, mobile and chatbot. ₹1 per finalized bill through credits.

udyogmanager.com · Domain purchased by the founder

This report covers the product vision, merchant workflows, requirements, pricing, accounting, GST preparation, architecture, data models, APIs, security, testing and delivery plan.

Product status: planning and development. The welcome page and concept materials are built. Application features, integrations and operational targets described here are planned; approved founder decisions are distinguished from proposals.

Contents

- 1 Product direction and decisions
- 2 Merchant coverage and release scope
- 3 Users and permissions
- 4 User journeys
- 5 Functional requirements
- 6 Credit pricing and recharge policies
- 7 Accounting and report definitions
- 8 GST and accountant workflows
- 9 Web and mobile experience
- 10 Chatbot specification
- 11 System architecture
- 12 Data model
- 13 API contracts and consistency
- 14 Security and privacy
- 15 Quality and operational targets
- 16 Testing and acceptance criteria
- 17 Delivery roadmap and team
- 18 Commercial model and launch
- 19 Risks and decisions to close
- 20 Launch checklist and source references

1 Product direction and decisions

1.1 Product promise

Udyog Manager should help a merchant answer five everyday questions: What did I sell? What money did I receive? Who still owes me? What did the business spend? What records should I send to my accountant?

The interface should make billing fast enough for daily use while preserving reliable underlying records. Reports must distinguish sales, collections, balances, and profit. A sale made on credit is revenue under the applicable accounting treatment, but it is not a cash collection.

1.2 Confirmed founder decisions

Area	Confirmed direction
Product name	Udyog Manager
Audience	Merchants across business categories
Channels	Web application, mobile application, and chatbot
Core backend	Django
Chatbot service	FastAPI
Web frontend	Next.js
Pricing unit	₹1 per bill generated through credits
Core business information	Customers, transactions, earnings, revenue, and bills
Reporting goal	GST reports and profit and loss information that merchants can provide to CAs

1.3 Recommended interpretation and implementation

For billing charges, “generated” means successfully finalized and assigned its official invoice number. Opening a draft, previewing a PDF, printing, downloading, or sharing the same issued bill does not create another charge.

Recommended technical choices are React Native with Expo for mobile, PostgreSQL for persistent data, and a modular Django application for business logic. These choices remain engineering proposals rather than founder-mandated decisions.

1.4 Product boundaries

Udyog Manager initially prepares business records and accountant-reviewable reports. Direct GST filing, payment acquiring for merchants, lending, payroll compliance, and advanced industry operations are separate capabilities with their own delivery gates.

The platform should not advertise a supported specialist workflow until it has been implemented and tested. Supporting a pharmacy's simple sales entry does not mean supporting pharmacy inventory controls. A complete long-term product can cover both, with explicit feature availability.

2 Merchant coverage and release scope

2.1 Shared foundation

Every business receives a common foundation: business profile, customers, products or services, bill creation, payment recording, customer balances, expenses, purchases, credits, exports, and permissions. Categories configure the experience without duplicating financial logic.

Merchant category	Shared billing needs	Additional category capabilities	Proposed delivery
General retail	Items, discounts, payments, receipts	Barcodes, stock, returns	Core plus retail inventory
Clothing and footwear	Item billing and exchanges	Size and colour variants	Category expansion
Wholesale and distribution	Customer credit and purchases	Price lists, units, order fulfilment	Category expansion
Professional and local services	Service lines and receipts	Appointments, recurring invoices	Basic services in core; advanced later
Restaurants and cafes	Sales and collections	Tables, kitchen orders, modifiers, split bills	Dedicated module
Pharmacy	Product sales	Batch and expiry controls, applicable sector requirements	Dedicated validated module
Repair businesses	Products, services, dues	Job cards, estimates, device history	Category expansion
Manufacturing and complex trade	Purchases and sales	Bills of materials, production, warehouses	Later programme

2.2 First commercial release

The first commercial release should support ordinary product and service billing, customer ledgers, purchase and expense recording, a bounded inventory model for goods, credit recharges, mobile access, and core reports. Chatbot scope should begin with queries and confirmed draft creation.

Include simple customer returns and appropriate correction documents before accepting production financial records. Avoid launching a system that can issue invoices but cannot handle normal mistakes, partial payments, or refunds.

2.3 Explicit expansion scope

Advanced inventory, multi-warehouse transfers, specialist category workflows, automated GST integrations, WhatsApp conversations, voice entry, offline finalization, and multi-branch consolidation follow separate milestones. Their data model dependencies should be considered early without building every feature into the first interface.

3 Users and permissions

Role	Primary permissions	Restrictions
Business owner	Business settings, users, billing, reports, credits, exports	Sensitive changes require reauthentication
Manager	Daily operations and reports within assigned scope	No ownership transfer or unrestricted credit adjustments
Billing staff	Customers, drafts, authorized bill issuance, payment recording	Profit, exports, refunds, and settings restricted by policy
Accountant or CA	Relevant records, exports, reconciliation, approved adjustments	No account ownership or recharge authority by default
Platform support	Diagnostic metadata and support cases	No routine access to customer financial content
Platform administrator	Platform operations and controlled intervention	Audited privileged access; no silent financial edits

A user can belong to multiple businesses with different roles. Every operation must enforce membership and permissions on the server. Selecting another business in the interface does not itself grant access.

Staff permissions should be granular enough to separate bill creation, discount approval, payment recording, refunds, customer-data export, profit visibility, and accountant access. Owner changes to roles and invited accountant access must be auditable and revocable.

4 User journeys

4.1 Merchant onboarding

1. Create an account and verify the selected sign-in channel.
2. Create a business and choose its category, business location, currency, and financial-year settings.
3. Enter the applicable GST registration status and invoice details.
4. Set invoice series, logo, payment instructions, and business contact information.
5. Add or import products, services, customers, and opening balances.
6. Preview a sample bill and review the ₹1 charge rule.
7. Purchase credits or use an explicitly granted promotional allocation.
8. Issue the first real bill and see its payment status and remaining credits.

Imports must provide validation previews, row-level errors, duplicate handling, and a summary before committing. Importing opening data must not silently create historical revenue or consume bill credits.

4.2 Billing at the counter

The merchant selects an existing customer or a permitted walk-in customer, adds items or services, checks quantities and discounts, and reviews calculated totals. The interface displays the credit cost and available balance. After confirmation, Django finalizes the bill and returns its immutable business snapshot and invoice number. PDF generation can finish asynchronously.

The merchant then records a payment or leaves the invoice unpaid. A failure to print does not undo the sale or consume an additional credit on retry.

4.3 Credit sales and collections

An unpaid or partially paid bill contributes to the customer's outstanding balance. Later receipts can be allocated across invoices. The system records unallocated advances separately and prevents accidental over-allocation. Reversing an incorrect receipt updates allocations and ledger entries with an audit record.

4.4 Returns and corrections

The merchant locates the original bill, records returned items or the correction reason, and creates the applicable linked adjustment. The system separates financial adjustment, physical stock return, and cash refund: these are related events, but one does not always imply the others.

Issued bills are not silently overwritten. The allowed correction or cancellation mechanism depends on document type, tax rules, filing status, and permissions.

4.5 Month end and accountant handoff

The merchant checks unrecorded expenses, missing purchase documents, unallocated receipts, and outstanding balances. An accountant reviews reports for the period, reconciles exceptions, and exports a dated package containing registers and supporting records. A period can then be locked against ordinary edits; reopening requires an authorized reason.

5 Functional requirements

5.1 Business and account management

FR01: Support business profiles, users, memberships, categories, tax settings, invoice templates, and financial-year configuration. Preserve historical invoice values when current profile details change.

FR02: Allow customer and supplier records with names, addresses, optional contact details, tax identifiers where applicable, notes, opening balances, and activity history. Do not require a phone number for every walk-in sale unless the applicable workflow needs it.

FR03: Support secure account recovery, invitation expiry, session revocation, and ownership-transfer controls. An inactive staff member must immediately lose access.

5.2 Catalogue and inventory

FR04: Support products and services with SKU, description, unit, selling price, purchase cost where relevant, tax classification, active status, and optional barcode. Tax and item details must be snapshotted on issued lines.

FR05: Track stock through append-only movements: opening stock, purchase receipt, sale, return, and adjustment. Show stock on hand and low-stock alerts. Negative stock should either be blocked or require an explicitly configured exception.

FR06: Establish a documented costing method for goods, such as weighted average for the initial bounded inventory scope. Handle backdated transactions, purchase returns, and cost corrections consistently. Do not imply that the current catalogue purchase price equals historical cost of goods sold.

5.3 Invoice lifecycle

FR07: Support draft creation, preview, validation, finalization, sharing, retrieval, and controlled adjustment. Drafts may be edited or deleted. Issued financial documents remain traceable.

FR08: Support quantity, unit price, line discounts, invoice-level discount allocation, applicable tax components, taxable value, rounding adjustment, notes, payment terms, and due date. Inclusive and exclusive tax modes must have explicit calculation rules.

FR09: Allocate invoice numbers on the server using the correct business, registration, series, and financial-year scope. Prevent collisions during concurrent requests and retain cancelled numbers in the audit history.

FR10: Produce readable A4 PDF invoices and a tested thermal-receipt layout where supported. Printed output must match stored totals. Downloading and reprinting an issued document must be free.

5.4 Payments and balances

FR11: Record cash, UPI, bank, and other permitted payment methods; reference identifiers; receipt date; and allocation to invoices. A manually marked UPI receipt is merchant-recorded information, not provider-confirmed payment verification.

FR12: Support partial payments, customer advances, refunds, bounced or reversed receipts, and credit balances. Receivable ageing must use outstanding amounts as of the selected date.

5.5 Purchases and expenses

FR13: Record supplier invoices, purchase lines, tax details, attachments, payment status, and relevant inventory receipts. Detect likely duplicate supplier bills without automatically rejecting legitimate duplicates.

FR14: Record expenses with category, date, supplier, amount, tax treatment, attachment, and payment account. Prevent a purchase from being recorded again as an expense. Capital purchases and recoverable taxes require appropriate classification.

5.6 Reporting and administration

FR15: Provide sales, collections, expenses, dues, inventory, and profit reports with date filters, business context, and drill-down to source records.

FR16: Export CSV or Excel-compatible files and PDFs where appropriate. Protect exported spreadsheet cells from formula injection. Large exports should be background jobs with expiring download links.

FR17: Provide an owner-visible credit ledger, recharge receipts, staff activity log, and notification preferences. Platform support needs tools to inspect job failures, investigate recharge disputes, and re-run documents without changing financial state.

6 Credit pricing and recharge policies

6.1 Proposed charge rules

Action	Credits consumed	Rule
Add customer or supplier	0	No charge for maintaining master data
Create or edit draft	0	No official issued document yet
Successfully finalize a new sales bill	1	Exactly one deduction for the bill
Retry the same finalization request	0 additional	Return the original result
Download, print, or share an existing bill	0	Reuse the existing bill
Record a payment or expense	0	Keep records complete
View or export existing data	0	Available even with zero credits
Create a linked correction or credit note	0 proposed	Subject to abuse controls and approved policy
Issue a separate replacement sales bill	1 proposed	It is a new issued document
Import historical issued bills	0 proposed	Separate migration flow; no duplicate issuance

Proposed cancellation policy: A successfully issued bill remains charged even if subsequently cancelled. A technical duplicate or verified platform failure may receive a compensating credit adjustment. Never delete a credit debit to conceal a refund. Publish the approved rule before paid use.

6.2 Recharge packages

Suggested packages are 100, 500, and 1,000 credits. At the base rate, their pre-tax or tax-inclusive presentation depends on the pricing decision below. Do not advertise volume discounts while simultaneously assuming every credit yields ₹1 in revenue.

The founder must choose whether ₹1 is inclusive of applicable tax or whether tax is added at recharge. Checkout must show the exact payable amount, credits granted, and any fees before payment. Promotional credits must be distinguished from purchased credits in both records and revenue analysis.

6.3 Recharge state and payment verification

Use a recharge order lifecycle such as created, pending, paid, failed, expired, and refunded. Grant credits only after server-side verification of a successful provider event or equivalent provider lookup. Client redirects are not proof of payment.

Verify webhook signatures, provider amount, currency, business ownership, and order mapping. Record provider event IDs under a uniqueness constraint. A duplicate webhook or retry must not grant additional credits. A periodic reconciliation job identifies provider-successful payments whose webhooks were missed.

Payment refunds and chargebacks need a separate policy for unused and consumed credits. Never silently remove a merchant's historical invoices to settle a recharge dispute.

6.4 Atomic credit deduction

Within one database transaction, check permissions, lock the relevant credit balance and invoice draft, validate available credits, allocate the invoice number, save the issued invoice, post required ledger and stock entries, insert the credit debit, and record an outbox event. Commit only when all required database operations succeed.

Two staff members attempting to issue bills with one remaining credit must produce at most one successful issuance. A balance cached in Redis is insufficient to enforce this rule.

PDF generation, email, and external services happen after commit. If they fail, retry the delivery task against the same issued bill. The merchant must be able to recover the result after a timeout using the request identifier.

6.5 Balance and access policy

At zero credits, allow drafts, collections, reports, exports, and access to existing data. Block only new chargeable issuance. Show low-balance reminders at configurable thresholds. Proposed paid credits do not expire during the initial offer; final terms require owner approval and clear disclosure.

The credit balance is a balance for using Udyog Manager services. It must not be confused with customer receivables, merchant bank balances, or a transferable payment wallet.

7 Accounting and report definitions

7.1 Accounting foundation

Use a balanced journal underneath financial reports. Business events create postings through explicit accounting services, with links to their source documents. Posted entries are corrected through reversals or adjustments. Draft documents do not affect the books.

The merchant's books and Udyog Manager's own commercial accounting are separate. Recharge receipts belong to Udyog Manager's platform records; a customer's retail purchase belongs to the merchant's records. Classification of prepaid credits and recognition of platform revenue require accounting review.

7.2 Report definitions

Metric	Product definition
Net sales	Recognized sales after discounts and sales adjustments, excluding output tax
Collections	Customer payments received in the selected period, separated from sales
Outstanding receivables	Amount owed after relevant invoices, receipts, advances, and adjustments
Cost of goods sold	Cost assigned to goods sold under the configured inventory costing method
Gross profit	Net sales less cost of goods sold or applicable direct costs
Operating profit	Gross profit less classified operating expenses
Net profit	Result after other recognized income, expenses, and applicable adjustments
Cash movement	Cash and bank receipts less disbursements for selected accounts
GST working summary	Tax-related values organized for accountant review, not a filing confirmation

Loan proceeds are not sales; owner withdrawals are not ordinary business expenses. Tax collected from customers must not inflate net sales. A purchase of inventory does not necessarily become an immediate profit-and-loss expense in full.

7.3 Illustrative transaction

Assume a sample item sale of ₹1,000 before an illustrative 18% tax, total ₹1,180. The merchant receives ₹500 immediately. The item's assigned inventory cost is ₹600. The tax rate is an arithmetic example, not a classification recommendation.

The sale posts ₹1,180 to receivables, ₹1,000 to sales, and ₹180 to output tax. The receipt increases the selected cash or bank account by ₹500 and reduces receivables by ₹500. Outstanding dues are ₹680. The stock posting recognizes ₹600 of cost of goods sold and reduces inventory by ₹600. Gross profit on this sale is ₹400 before other expenses.

Issuing the invoice consumes one Udyog Manager credit. That usage event is tracked separately from the customer's ₹1,180 purchase.

7.4 Report completeness

Show the reporting period, accounting basis, generation time, applied filters, and data completeness warnings. If expenses, opening balances, or inventory costs are missing, label the profit report provisional and identify the gaps. Do not present a partial sales-minus-expenses calculation as a finalized profit and loss statement.

All summary totals should reconcile to supporting registers and the journal. Period locks, documented backdating rules, and an exception report are necessary for dependable accountant handoff.

8 GST and accountant workflows

8.1 Tax configuration

Capture registration status, relevant business registration and location, customer tax identity where applicable, place-of-supply information, item classification, and effective-dated tax configuration. Support the appropriate document types for registered, unregistered, composition, and exempt-supply scenarios after CA validation.

The applicable invoice particulars and bill-of-supply distinction should be implemented against the current rules. Official CBIC guidance identifies prescribed invoice particulars and the use of bills of supply in applicable circumstances. [S1, S2]

Do not hard-code one tax rate across merchants or infer all place-of-supply rules from a customer's phone number or address. Preserve the rule version and calculated components used when a bill was issued.

8.2 Reports for the accountant

Provide a sales register, purchase register, tax-component summary, HSN/SAC summary where applicable, adjustment register, and missing-information report. Include document references so discrepancies can be traced to source records.

GSTR-1 is the outward-supplies reporting workflow. Udyog Manager should prepare reviewable data grouped for the applicable return format, with supported and unsupported transaction types explicitly identified. [S3]

GSTR-2B data can be downloaded from the GST portal. A later reconciliation feature should compare imported data with recorded purchases and classify matched records, missing records, and amount differences. Matching a record alone must not be treated as a final eligibility decision for input tax credit. [S4]

GSTR-3B working summaries should be reviewed and mapped by the accounting adviser against the current official return structure before release. Do not label an ordinary sales-tax total as a complete return.

8.3 Integration stages

Initial stage: Provide accountant-readable exports and supporting records. Clearly distinguish a preparation export from a portal-validated filing file.

Reconciliation stage: Import supported GST data formats, validate schemas, retain source timestamps, and surface exceptions for review.

Integrated stage: Add eligible provider integrations, e-invoice or e-way bill workflows where required, external reference storage, retries, and status reconciliation. For workflows that require an external registration before valid issuance, create a dedicated issuance state machine and credit reservation policy; the simple local-finalization flow alone is insufficient.

8.4 Compliance release gate

A practicing CA should review tax examples, invoice formats, adjustments, date boundaries, return mappings, and representative merchant scenarios. Confirm current applicability thresholds and deadlines at implementation time rather than freezing them into this planning document. Privacy and retention requirements also require launch-specific review.

9 Web and mobile experience

9.1 Primary navigation

Use Dashboard, Bills, Customers, Products and Services, Purchases, Expenses, Payments, Reports, Credits, and Settings. Category modules appear only when enabled. Keep a persistent create-bill action and an obvious business switcher for users with multiple memberships.

9.2 Required screens

Screen	Essential content
Dashboard	Sales, collections, dues, expenses, provisional profit status, credit balance
Bill editor	Customer, line items, taxes, discounts, totals, credit cost, finalization action
Bill detail	Stored bill, payment status, adjustment history, print and share
Customer detail	Contact details, statement, invoices, payments, outstanding balance
Inventory	Items, stock movement, cost visibility by permission, low-stock status
Reports	Period filters, definitions, reconciliation links, export status
Credits	Balance, package selection, verified recharge status, usage ledger
Accountant workspace	Scoped records, exceptions, exports, period review
Chat	Conversation, structured previews, confirmation, source links

9.3 Mobile requirements

Support fast touch entry, searchable products, device sharing, camera barcode scanning where enabled, secure local session storage, and tested printer integrations. Android and iOS are part of the mobile direction; release order should follow pilot demand and available testing capacity.

Start with online finalization and optionally cached read-only data or local drafts. Clearly show synchronization state. A draft saved offline must not display an official invoice number or claim successful issuance.

Full offline issuance requires preallocated numbering or another approved numbering design, secure credit reservations, device identity, conflict handling, and reconciliation. Decide this before promising operation without internet.

9.4 Accessibility and language

Use readable type, clear focus states, labelled inputs, keyboard-accessible web workflows, and errors that identify the correction needed. Never convey payment or synchronization state only through colour. Plan for English and Hindi labels initially if pilot demand supports them, with currency and date formatting separated from stored values.

10 Chatbot specification

10.1 Initial supported intents

The chatbot can answer sales and collection questions, list unpaid invoices, search customer statements, show credit balance, create a bill draft, and request a report export. It should return links to the underlying records and explain the date range and filters it used.

Later intents can include multilingual voice entry, purchase-document extraction, reminders, and WhatsApp access. Each introduces separate accuracy, consent, integration, and operating-cost requirements.

10.2 Safe action flow

1. Authenticate the user and establish the authorized business context.
2. Resolve intent and extract structured fields.
3. Ask for genuinely missing details such as an ambiguous customer or item.
4. Fetch canonical data and compute totals through Django.
5. Show a preview, affected business, totals, and credit deduction.
6. Obtain confirmation for issuance or other financial changes.
7. Submit an authorized Django operation with an idempotency key.
8. Return the saved result and remaining credit balance.

A confirmation must be bound to the specific draft version and expire. If quantities, customer, prices, or totals change after confirmation, a new confirmation is required.

10.3 Operational controls

FastAPI orchestrates conversation; it does not write financial tables directly or execute arbitrary model-generated SQL. Tools are allowlisted, permission checked, and supplied with validated schemas. Customer notes and uploaded documents are untrusted data and cannot override tool permissions.

Apply request limits, model-cost budgets, minimal data sharing, redacted logs, and configurable retention. Report calculations come from the backend rather than model arithmetic. When the model is unavailable, normal billing must still work.

Use a test set of representative utterances, ambiguous names, mixed-language inputs, malicious embedded instructions, and unauthorized queries. Track intent accuracy and confirmed-action correctness separately. The proposed ₹1 charge does not imply unlimited costly AI use; any fair-use or paid AI policy must be explicit before launch.

11 System architecture

11.1 Components

Component	Proposed technology	Responsibility
Web client	Next.js and TypeScript	Merchant and accountant interfaces
Mobile client	React Native with Expo	Shared mobile product for Android and iOS
Business API	Django and Django REST Framework	Identity, billing, accounting, credits, permissions
Conversation service	FastAPI	Model orchestration and structured tool calls
Primary database	PostgreSQL	Authoritative business and financial state
Job processing	Celery and Redis	PDF generation, exports, notifications, retries
File storage	S3-compatible object storage	Attachments and generated documents
Recharge provider	Selected payment gateway	Platform credit purchases and verification
Observability	Structured logs, metrics, error reporting	Operational visibility and incident investigation

Web and mobile call Django. Chat clients call FastAPI, which invokes Django using constrained authorization. Django commits data to PostgreSQL and schedules background work through an outbox. Workers write generated files to protected storage and report job status.

11.2 Backend organization

Start with modules for identity, businesses, parties, catalogue, inventory, invoicing, payments, purchases, expenses, accounting, credits, reporting, and audit. Modules may share one deployable Django application while keeping explicit service boundaries.

Use a service layer for critical operations, not scattered model signals that obscure transaction ordering. Domain events can drive asynchronous side effects, but financial integrity must be enforceable within the authoritative transaction.

11.3 Environments and delivery

Maintain development, staging, and production environments with separate databases, storage, secrets, and payment configurations. CI runs linting, type checks, tests, migration checks, and dependency checks. Production deployment uses reviewed migrations, health checks, and a documented rollback or forward-fix strategy.

Pin supported framework versions during implementation and maintain an upgrade cadence. Do not rely on an unreviewed “latest” dependency in production. Protect database changes with expand-and-contract migrations where compatibility requires it.

12 Data model

12.1 Core entities

Entity	Important fields and relationships
User	Identity, verified contacts, status, security settings
Business	Name, category, timezone, currency, ownership
Membership	User, business, role, permission scope, status
TaxRegistration	Business, identifier, jurisdiction, effective dates
Branch	Business, address, registration relationship, settings
Party	Business, customer or supplier roles, contact and tax details
ProductService	Business, SKU, type, unit, price, classification
Invoice	Business, registration, series, number, date, customer snapshot, totals, state
InvoiceLine	Invoice, item snapshot, quantity, price, discount, tax components
AdjustmentDocument	Original reference, reason, amounts, status
Payment	Business, party, amount, method, date, reference, state
PaymentAllocation	Payment, invoice or adjustment, allocated amount
Purchase	Business, supplier, supplier reference, date, totals, state
Expense	Business, category, account, amount, tax treatment, attachment
StockMovement	Product, location, source, quantity change, assigned cost
JournalEntry	Business, source event, date, posting state, reversal reference
JournalLine	Journal entry, account, debit or credit, currency
CreditAccount	Business, available credit units, version
CreditLedgerEntry	Account, signed credit change, reason, source identifier
RechargeOrder	Business, expected amount, currency, provider reference, state
ProviderEvent	Provider, unique event ID, verification outcome, processing state
AuditEvent	Actor, business, action, target, timestamp, reason
OutboxEvent	Event type, payload reference, delivery state, retry count
ExportJob	Business, requester, filters, status, protected file reference

12.2 Integrity rules

Every business-owned record includes a tenant identifier, and references must resolve within the same authorized tenant. Where possible, enforce this with database constraints in addition to query filtering. Guess-resistant identifiers do not replace access control.

Invoice series and numbers require the correct uniqueness scope. Each finalized invoice has at most one normal bill-credit debit. Recharge events have unique provider identifiers. Journal entries balance before posting. Allocations cannot exceed permitted payment or invoice balances without an explicit advance or credit workflow.

Store currency amounts in decimal fields or exact minor units as appropriate, with explicit rounding. Credit units are integers. Store timestamps consistently and render business dates in the configured timezone. Snapshot historical tax and customer details rather than depending on mutable master data.

12.3 Document states

Separate document state from payment state. For example, an invoice may be issued and partially paid; a credit note may reduce its balance without changing the historical fact that it was issued. External fiscal-registration states, when introduced, require their own transitions.

Financial records use retention and controlled adjustment rules; do not apply unrestricted soft-delete or hard-delete operations to all models indiscriminately.

13 API contracts and consistency

13.1 Representative endpoints

Endpoint	Purpose
POST /api/v1/businesses	Create an authorized business
GET or POST /api/v1/customers	Search or create customer records
GET or POST /api/v1/items	Search or create products and services
POST /api/v1/invoices	Create a draft
PATCH /api/v1/invoices/{id}	Update an editable draft with version checks
POST /api/v1/invoices/{id}/finalize	Issue once and consume one credit
GET /api/v1/invoices/{id}/document	Retrieve document or generation status
POST /api/v1/payments	Record a receipt and allocations
POST /api/v1/adjustments	Create an authorized linked adjustment
GET /api/v1/credits/balance	Retrieve available credits
POST /api/v1/recharges	Create a provider-backed recharge order
POST /api/v1/provider-webhooks/{provider}	Verify and process provider events
GET /api/v1/reports/sales	Return filtered sales totals and drill-down
POST /api/v1/exports	Start an authorized export job
POST /chat/v1/messages	Process a conversation request

13.2 Contract conventions

Return consistent validation errors with a machine-readable code, safe message, field errors where relevant, and correlation ID. Distinguish insufficient credits, stale draft version, forbidden access, tax validation failure, and temporary dependency failure.

List endpoints require pagination and bounded filters. Search must remain tenant scoped. Monetary values should serialize as exact decimal strings rather than binary floating-point approximations.

13.3 Idempotency and concurrency

Require an idempotency key for finalization and other retryable financial writes. Scope the key to the business and operation; retain the request hash and result. Reusing a key with different content returns a conflict. A client timeout must be recoverable through the same key or an operation-status lookup.

Use row locking, uniqueness constraints, and explicit transaction boundaries. Test concurrent deductions and numbering, including financial-year changes and repeated mobile requests. An outbox worker uses retry-safe consumers so notifications and exports can be retried without duplicate financial effects.

14 Security and privacy

14.1 Access and isolation

Validate tenant membership and permission on every API, file download, export, background task, and chatbot tool invocation. Restrict privileged support access, record the reason for access, and maintain an auditable history. Mask sensitive data when full access is unnecessary.

Use secure browser cookies and CSRF controls for applicable web sessions, and securely stored, revocable credentials for mobile sessions. Protect recovery and verification flows with rate limits and abuse detection. Consider stronger authentication for owners and administrators.

14.2 Data protection

Encrypt network traffic and use managed encryption for databases, backups, and object storage. Keep secrets outside source control. Use short-lived signed file links and validate attachment type and size; scan uploads where appropriate. Exclude credentials, full financial payloads, and unnecessary personal data from logs.

Collect only necessary customer data, document the purpose of each use, and provide merchant-controlled notification preferences. Export and deletion requests need an identity-checked workflow with rules for financial retention, backups, and third-party processors. Confirm applicable privacy and record-retention requirements with appropriate advisers before launch.

14.3 Resilience and incidents

Back up the database and object storage, test restoration into an isolated environment, and document incident ownership. Recovery must reconcile credits, invoices, and recharge-provider events; restoring only the visible balance is insufficient.

Maintain runbooks for suspected tenant leakage, account takeover, recharge discrepancies, database failure, and unavailable document generation. Customer communication should state verified impact and recovery steps.

15 Quality and operational targets

These are proposed engineering acceptance targets to validate against deployment cost and realistic pilot traffic.

Area	Proposed initial target
Routine API latency	p95 below 500 ms at an agreed representative load
Bill finalization	p95 below 2 seconds, excluding external fiscal services and PDF delivery
PDF readiness	Most standard bills available within 10 seconds under normal load
Service availability	99.5% monthly target for initial production
Recovery point	15 minutes or better, subject to selected backup architecture
Recovery time	4 hours or better, verified through restore exercises
Credit correctness	No duplicate charges or grants in acceptance and concurrency tests
Tenant isolation	No cross-business access in the defined security test suite
Accessibility	Critical journeys usable with keyboard and labelled controls

Measure finalization success, recharge success, outbox lag, queue depth, failed exports, reconciliation discrepancies, database contention, and chatbot cost. Alert on symptoms affecting merchants rather than merely logging exceptions.

Define an operational dashboard and support escalation schedule before accepting paid users. Availability targets are internal objectives until contractual terms are approved.

16 Testing and acceptance criteria

16.1 Critical acceptance scenarios

ID	Scenario	Expected result
AC01	Finalize one valid bill with available credit	One invoice, one credit debit, correct postings
AC02	Retry finalization after a network timeout	Same result; no second deduction or number
AC03	Two concurrent bills with one remaining credit	At most one succeeds; balance never negative
AC04	Finalization fails validation	No bill issuance, credit debit, or accounting entry
AC05	PDF generation fails after commit	Bill remains issued; retry PDF without a new charge
AC06	Duplicate successful recharge webhook	Credits granted once
AC07	Forged or mismatched provider event	No credits granted; event logged safely
AC08	User requests another business's invoice or export	Access denied without leaking record content
AC09	Partial receipt followed by reversal	Customer balance and journal reconcile
AC10	Return with stock receipt but no cash refund yet	Stock, customer credit, and cash reflect separate events
AC11	Financial-year rollover with concurrent issuance	Numbers remain unique in approved scope
AC12	Zero-credit merchant opens reports and records payment	Allowed; new chargeable bill blocked
AC13	Chatbot receives an ambiguous customer name	Requests clarification before creating a financial action
AC14	Chat confirmation references a stale draft	Requires refreshed preview and confirmation
AC15	Restore production-like backup	Records reconcile and agreed recovery targets are measured
AC16	Imported expenses or costs are incomplete	Profit report signals incompleteness

16.2 Testing layers

Unit tests cover rounding, discounts, accounting rules, allocations, and permission decisions. Database integration tests cover transactions, locks, uniqueness, tenant isolation, and outbox processing. End-to-end tests cover onboarding, recharge, issuance, collection, corrections, and accountant export across channels.

Use CA-reviewed golden examples for tax and accounting. Test more than a single standard-rate sale: include exempt lines, inclusive pricing, mixed tax treatment, discounts, returns, and relevant intra-state and inter-state cases supported by the release.

Test mobile network loss, app restart, slow responses, stale drafts, and duplicate taps. Validate printed documents on the actual supported paper sizes and printers. Run a bounded load test against the agreed merchant and invoice profile before launch.

17 Delivery roadmap and team

17.1 Indicative plan

The following is a planning scenario for a small experienced team working on overlapping streams. Re-estimate after discovery. It does not cover every specialist merchant module or guarantee a launch date.

Phase	Indicative window	Deliverables	Exit condition
Discovery and design	Weeks 1 to 2	Merchant interviews, CA review, flows, scope and policies	Critical requirements and launch segment agreed
Foundation	Weeks 3 to 5	Identity, tenant isolation, data model, CI, catalogue	Access and database foundations tested
Core billing	Weeks 6 to 9	Drafts, issuance, credits, recharges, receipts, PDF	Critical financial scenarios pass
Records and reporting	Weeks 10 to 12	Purchases, expenses, bounded inventory, reports, exports	Accounting examples and reports reconcile
Mobile and chatbot	Weeks 10 to 14	Mobile workflows, query and confirmed-action chatbot	Channel parity for selected flows
Pilot and hardening	Weeks 15 to 16	Merchant pilot, fixes, restore exercise, support runbooks	Launch gates satisfied
Category expansion	After pilot	Prioritized specialist modules and integrations	Module-specific acceptance criteria met

17.2 Suggested responsibilities

The founder owns positioning, policy, pilot recruitment, and commercial decisions. A product designer owns workflows and usability. Backend engineering owns financial integrity and APIs. Web and mobile engineering own channel experiences. QA owns acceptance evidence. An accounting adviser validates financial and tax behavior. Infrastructure responsibility must be assigned even if it is part-time.

A possible staffing model is one backend engineer, one web engineer, one mobile engineer, shared design and QA support, and a retained CA. This is a capacity assumption, not a hiring or cost estimate. A smaller team should reduce concurrent scope or extend the schedule.

17.3 Definition of done

A feature is complete when its acceptance criteria pass, permissions are enforced, accounting effects reconcile where applicable, audit records exist, user-facing errors are usable, documentation is updated, and operational ownership is clear. A working screen alone is not sufficient for a financial workflow.

18 Commercial model and launch

18.1 Positioning

Suggested product statement: “Udyog Manager helps merchants create bills, track dues, and keep business records together for ₹1 per issued bill.” Brand names, domains, and trademarks require separate availability checks before registration or advertising.

The product should compete on understandable pricing, reliable records, easy billing, and useful accountant handoff. Broad category support is an expansion strategy; initial messaging should accurately state available workflows.

18.2 Illustrative usage economics

Monthly active merchants	Average chargeable bills per merchant	Chargeable bills	Usage value at ₹1
100	300	30,000	₹30,000
500	300	150,000	₹150,000
1,000	300	300,000	₹300,000

These are arithmetic scenarios, not market forecasts or recognized-revenue projections. They exclude tax effects, promotional credits, refunds, provider fees, and operating costs. Recharge cash received in a month can differ from credits consumed in that month.

Track contribution after payment processing, document generation, messaging, AI usage, storage, and variable support. A ₹1 price does not establish profitability without real usage and cost data. Bulk recharges reduce the need for tiny individual payment transactions.

18.3 Pilot strategy

Recruit a manageable pilot across two or three categories using the common billing workflows, while retaining the all-category product direction. Observe real counter use and month-end handoff. Record reasons for abandoned bills, duplicate customer entry, support requests, recharge hesitation, and spreadsheet exports.

Suggested channels to test include direct merchant onboarding, local accountant referrals, and merchant associations. Treat these as experiments with measured acquisition cost and retention, not established sales channels.

18.4 Product metrics

Track onboarding-to-first-bill conversion, time to first bill, active billing merchants, issued bills per merchant, recharge conversion, repeat recharge, retention, invoice correction rate, support contacts, and cost per issued bill. Segment by merchant category and channel without exposing unnecessary personal data.

An active merchant should have a clear operational definition, such as at least one issued bill during the month. Monitor customer value through reliable workflow completion, not only sign-ups or chatbot messages.

19 Risks and decisions to close

19.1 Principal risks

Risk	Consequence	Mitigation
Too many specialist workflows at once	Slow release and inconsistent behavior	Shared foundation with explicit module gates
₹1 pricing below operating cost	Unsustainable growth	Measure contribution per merchant and limit unpriced costly extras
Incorrect profit or tax reporting	Loss of trust and accounting errors	Ledger foundation, completeness flags, CA-reviewed scenarios
Duplicate charges or recharge grants	Disputes and financial leakage	Idempotency, database constraints, reconciliation
Tenant data leakage	Serious privacy and business impact	Server-side isolation tests and controlled exports
Poor internet at billing counters	Interrupted sales	Fast online flow; deliberate offline design if validated
Chatbot creates incorrect transactions	Financial errors	Backend calculations, scoped tools, versioned confirmations
Incomplete imported records	Misleading balances and reports	Import validation and explicit opening-balance workflow
Unclear correction and refund policies	Support burden and disputes	Publish understandable policies before paid launch

19.2 Decisions requiring founder approval before implementation commitment

1. Is ₹1 inclusive of applicable tax, or is tax added at recharge?
2. What are the minimum recharge, promotional credits, credit expiry, and refund rules?
3. Is online-only issuance acceptable for launch, or is offline issuance essential?
4. Which category modules must be present in the first commercial release?
5. Are Android and iOS launched together or in sequence?
6. Is the first chatbot in-app only, or must WhatsApp be included?
7. Which languages are required at launch?
8. Are multiple businesses, GST registrations, and branches required immediately?
9. What support hours and service commitments can the team sustain?
10. Who approves tax configuration, report mappings, and accounting scenarios?
11. Which payment gateway, hosting region, and AI provider will be selected?
12. What is the available development budget and team capacity?

These decisions should be captured in a decision log with owner, date, approved choice, and consequences. They do not change the already confirmed name, core stack direction, channels, or per-bill pricing unit.

20 Launch checklist and source references

20.1 Product and engineering readiness

- ☐ Business onboarding and permissions validated with pilot merchants.
- ☐ Supported merchant categories and module limitations clearly documented.
- ☐ Draft, issuance, collection, correction, and refund journeys tested.
- ☐ ₹1 charging trigger and recharge tax presentation approved.
- ☐ Duplicate-request and concurrent-credit tests passed.
- ☐ Payment-provider reconciliation and dispute process operational.
- ☐ Reports reconcile to registers and underlying accounting entries.
- ☐ Invoice formats and supported GST scenarios reviewed by a CA.
- ☐ Mobile network-failure and document-printing scenarios tested.
- ☐ Chatbot permissions, confirmations, and failure behavior verified.
- ☐ Tenant isolation and export authorization tested.
- ☐ Backup restore exercise completed and results recorded.
- ☐ Privacy notice, service terms, credit policy, and support process approved.
- ☐ Brand availability checks and necessary registrations completed.
- ☐ Pilot feedback reviewed and launch-blocking defects resolved.

20.2 Official references

The references below were consulted on 13 October 2026. GST pages may contain legacy examples or change over time; verify the current consolidated rules, notifications, portal schemas, and applicable treatment during implementation. Technical designs and business policies elsewhere in this document are proposals, not claims drawn from these sources.

S1 — CBIC invoice guidance Prescribed invoice particulars and related document guidance. <https://cbic-gst.gov.in/gst-invoice-rules.html>

S2 — CBIC tax invoice provisions Official statutory reference including applicable bill-of-supply treatment. <https://taxinformation.cbic.gov.in/content-page/explore-act/1000300/1000001/invoice/ACTS>

S3 — GST portal GSTR 1 user guide Outward-supplies return workflow and official portal guidance. https://tutorial.gst.gov.in/userguide/returns/GSTR_1.htm

S4 — GST portal GSTR 2B FAQs Official guidance on the input tax credit statement and download facilities. https://tutorial.gst.gov.in/userguide/returns/FAQ_gstr2b.htm

20.3 Immediate implementation handoff

Use this specification to create a prioritized backlog, approve the unresolved commercial policies, design the onboarding and bill-finalization screens, and define CA-reviewed accounting examples. The first engineering milestone should prove tenant isolation, atomic bill issuance, and credit reconciliation before the product expands into specialist modules.