

User Stories

Billing & Service Contract Management System

Format: As a <role>, I want to <action>, so that <benefit>. Each story includes acceptance criteria as implemented and verified.

Epic 1 — Concern (Customer) Management

US-1.1

As a **Billing User**, I want to create a Concern with its address, nature, and contact details, so that I can issue bills against it.

Acceptance criteria - Concern Name, Address, Nature, Contact Person, Contact Number are required; Proposed Date, Kick-off Date, Designation are optional. - Nature must be one of Work Order / Proposal / Others. - On save, I am taken to the Concern's detail page.

US-1.2

As a **Billing User**, I want to attach supporting documents to a Concern, so that proposals/work orders are kept with the customer record.

Acceptance criteria - I can upload multiple files at once. - Previously uploaded files are never replaced or deleted when I upload more — all attachments remain visible with their upload date.

US-1.3

As any logged-in user, I want to search Concerns by name, so that I can quickly find the right customer when creating a bill or contract.

Acceptance criteria - Typing into the Concern field on any bill/contract creation form live-searches by name (AJAX) and lets me pick from matches.

Epic 2 — One-Time Billing (License / Service)

US-2.1

As a **Billing User**, I want to create a License bill for a Concern with an amount and effective date, so that the customer has a formal demand for a one-time license fee.

Acceptance criteria - Concern, Effective Date, and Amount (> 0) are required. - The bill receives a unique Bill No automatically. - Immediately after creation, Outstanding equals the full bill amount and Status is "Generated".

US-2.2

As a **Billing User**, I want to create a Service/Other bill the same way as a License bill but labeled for a one-time service, so that ad-hoc service charges (e.g. installation) are tracked without needing a recurring contract.

Acceptance criteria - Selecting "Service / Other" as the bill type relabels the date field "Service Date" and the remarks field "Service Type / Remarks". - No contract is created or required for this bill type.

Epic 3 — Bill Splitting

US-3.1

As a **Billing User**, I want to split a 10,000 Tk. bill into a 5,000 Tk. child bill, so that I can track and collect the customer's partial payment separately from the rest.

Acceptance criteria - The child bill gets its own Bill No ({master}-{01}), its own independent payment/outstanding tracking, and a visible link back to the master. - The master bill's remaining (unallocated) balance decreases by the split amount.

US-3.2

As a **Billing User**, I want the system to stop me from splitting more than what remains unallocated, so that the sum of child bills never exceeds the original bill.

Acceptance criteria - Splitting 5,000 + 3,000 + 2,000 from a 10,000 Tk. bill succeeds and leaves 0 remaining. - A further split attempt (even for 1 Tk.) is rejected with a clear error, and the master bill shows status "Fully Split".

US-3.3

As a **Billing User**, I want to be prevented from splitting a child bill further, so that the split hierarchy never gets more than one level deep.

Acceptance criteria - The Split action/page is unavailable for any bill that already has a `master_bill_id` set.

Epic 4 — Payment Collection

US-4.1

As a **Billing User**, I want to record a payment against a specific bill, so that the customer's outstanding balance is reduced and reflected immediately.

Acceptance criteria - I choose receipt date, amount, and payment account (Cash/Bank). - The bill's Outstanding decreases by exactly the payment amount; a unique Receipt No is generated.

US-4.2

As a **Billing User**, I want the system to reject a payment larger than what's actually outstanding, so that I never accidentally over-collect or create a negative balance.

Acceptance criteria - Attempting to pay more than the current outstanding amount shows an error naming the exact outstanding figure, and no payment record is created.

US-4.3

As a **Billing User**, I want a bill that's fully paid to be clearly marked "Paid", so that I don't accidentally try to collect on it again.

Acceptance criteria - Once Outstanding reaches 0 via payment(s), Status changes to "Paid" and the Payment action is no longer available for that bill.

Epic 5 — Bonus / Waiver

US-5.1

As a **Billing Manager**, I want to close a bill at zero cash collection by applying a bonus/waiver, so that goodwill write-offs are possible without a fake cash entry.

Acceptance criteria - I must provide a reason. - The adjustment amount cannot exceed the bill's current outstanding. - A bill closed entirely by adjustment (no payment at all) shows status "Bonus Closed", distinct from "Paid".

US-5.2

As a **Billing User** (non-manager), I want to be blocked from applying a bonus, so that waivers always go through an authorized approver.

Acceptance criteria - The Bonus/Waiver action is hidden for my role in the UI, and the underlying endpoint returns HTTP 403 if I try to call it directly.

US-5.3

As a **Billing Manager**, I want an over-sized bonus request rejected, so that a bill can never show negative outstanding.

Acceptance criteria - Requesting a bonus larger than the current outstanding is rejected with an error naming the real outstanding amount; no adjustment is recorded.

Epic 6 — Recurring Contracts (AMC / MMC)

US-6.1

As a **Billing Manager**, I want to set up an MMC contract with a monthly amount and a billing day, so that monthly maintenance bills are generated automatically without me re-entering them every month.

Acceptance criteria - I specify Effective From, Effective To, Billing Day (1-31), and the monthly amount. - Running the recurring generation process creates one bill per due month, dated on the configured billing day (clamped to the last day of a shorter month).

US-6.2

As a **Billing Manager**, I want to set up an AMC contract with a yearly amount, so that annual maintenance bills are generated automatically on each contract anniversary.

Acceptance criteria - One bill is generated per anniversary year within the Effective From/To range.

US-6.3

As a **Billing Manager** or the **system (scheduler)**, I want to run the recurring generation process repeatedly without ever creating duplicate bills for the same period, so that it's safe to schedule daily without supervision.

Acceptance criteria - Running the process twice in a row with no contract changes between produces zero new bills on the second run. - This is enforced both by application logic and by a database-level uniqueness constraint on (contract, billing period), so even a race condition cannot create a duplicate.

US-6.4

As a **Billing Manager**, I want to extend a contract's end date when it's renewed, so that billing continues without creating a brand-new contract.

Acceptance criteria - I can edit Effective To on an existing (even expired) contract. - The old and new end dates, who changed them, and when, are recorded and visible on the contract's history. - If the contract had auto-expired, I can also set its status back to Active in the same action.

US-6.5

As a **Billing Manager**, I want to change a contract's billing day without affecting bills that were already generated, so that correcting the cycle going forward doesn't retroactively change historical bills.

Acceptance criteria - Already-generated bills keep their original effective dates after the billing day is changed. - The next generation run uses the new billing day for any period not yet billed.

US-6.6

As a **Billing Manager**, I want an Active contract whose end date has passed to be automatically flagged Expired, so that I don't have to remember to close it out manually.

Acceptance criteria - Viewing the contract (or running generation) after its Effective To date has passed shows status "Expired" without any manual action.

Epic 7 — Reporting

US-7.1

As an **Accounts User**, I want a Bill Register showing every bill with its amount, paid, adjustment, and outstanding, so that I can reconcile total billing activity for a period.

US-7.2

As an **Accounts User**, I want an Outstanding Bill Report with an ageing column, so that I can follow up on old unpaid bills first.

US-7.3

As an **Accounts User**, I want a running statement (ledger) for one Concern showing opening balance, every bill/payment/adjustment, and closing balance, so that I can answer "how much does this customer currently owe" at a glance.

US-7.4

As an **Accounts User**, I want a Collection Report of all receipts, so that I can verify cash/bank deposits against what was recorded as collected.

US-7.5

As a **Billing Manager**, I want a Bonus/Adjustment Report, so that I can review all waivers granted and by whom, for audit purposes.

US-7.6

As a **Billing Manager**, I want a Recurring Billing Report showing each contract's last-generated and next-due period, so that I can spot a contract that's falling behind on generation.

Epic 8 — Access Control

US-8.1

As an **Administrator**, I want every role restricted to only the actions appropriate to it, so that, for example, a Billing User cannot grant bonuses or alter contract terms.

Acceptance criteria - Role is enforced server-side on every action, confirmed by directly calling a restricted endpoint as a lower-privileged role and receiving HTTP 403.

Billing & Service Contract Management System — User Stories