Vehicle Repairs

1: Overview

The **Repairs module** in BAT Connect manages the full lifecycle of vehicle repair expenses charged to drivers.
Its purpose is to ensure that repair costs are **recorded, structured into manageable repayment schedules, and systematically reconciled through weekly DTRs.**

Key Objectives:

1. **Invoice Capture** – Allow BAT to create and store detailed repair invoices for each vehicle/driver.
2. **Payment Plan Creation** – Automatically generate a weekly repayment schedule using the Repair Payment Matrix.
3. **Installment Deduction** – Deduct only the scheduled weekly installment from the driver's DTR.
4. **Balance Management** – Carry forward unpaid balances until the invoice is fully settled.
5. **Driver Transparency** – Show only the weekly deduction in the driver-facing DTR, while BAT tracks full invoice and outstanding balances.

Business Benefit:
This prevents large repair costs from overwhelming a driver's weekly earnings while ensuring BAT recovers repair expenses reliably and in a structured way.

---

2. Fields & Rules

Field	Source / Entry	Description	Rules / Validation	Entity
Repair ID (Internal)	System-generated	Unique internal ID for linking invoice and installments.	Hidden from UI; used as parent key.	Invoice Master

Invoice Number	Manual entry (from workshop)	Actual invoice number issued by the workshop.	Must be unique per workshop; duplicates not allowed for same vehicle/date.	Invoice Master
Invoice Date	Manual entry	Date repair was billed.	Must be ≤ current date.	Invoice Master
VIN	BAT Connect (Vehicle records)	Vehicle Identification Number tied to repair.	Mandatory; auto-fetched once vehicle selected.	Invoice Master
Plate Number	BAT Connect (Vehicle records)	Plate number of the vehicle.	Mandatory.	Invoice Master
Medallion Number	BAT Connect (Lease records)	Medallion associated with the repair.	Mandatory; must map to active medallion.	Invoice Master
Hack License Number	BAT Connect (Driver records)	TLC license of responsible driver/leaseholder.	Optional if repair is vehicle-only.	Invoice Master
Workshop Type	Dropdown	Workshop where repair was performed.	Values: `Big Apple Workshop`, `External Workshop`.	Invoice Master

Repair Description / Notes	Free-text	Details of the repair (e.g., “Engine replacement”).	Max 500 chars.	Invoice Master
Repair Amount (Total)	Manual entry	Total cost of repair invoice.	Must be ≥ $1.	Invoice Master
Start Week	Dropdown	Determines first installment deduction.	Values: `Current Payment Period` or `Next Payment Period`.	Invoice Master
Payment Plan (Weekly Installment)	System-calculated	Installment amount derived from Payment Matrix.	Auto-generated once Repair Amount is entered.	Payment Schedule
Installment ID	System-generated	Unique identifier for each weekly installment.	Format: `[RepairID]-[Seq]` (e.g., `RPR-2025-012-03`).	Payment Schedule
Week Start Date	System-calculated	Start date of the payment period.	Always Sunday 00:00:00.	Payment Schedule
Week End Date	System-calculated	End date of the payment	Always Saturday	Payment Schedule

		period.	23:59:59.	
Payment (This Period)	System-calculated	Installment amount due for this week.	Posted to ledger on Sunday 05:00.	Payment Schedule
Prior Balance	System-populated	Remaining unpaid balance from earlier cycles.	Auto-carried forward.	Payment Schedule
Balance	System-calculated	Remaining unpaid portion after this installment.	Formula: `Repair Amount - (Sum of Payments)`.	Payment Schedule
Ledger Posting Ref	System-generated	Ledger entry ID created when installment is posted.	Null until posted.	Payment Schedule
Status	System-calculated	Lifecycle status.	Invoice Master: `Open` / `Closed` / `In Dispute`. Payment Schedule: `Scheduled` /	Both

			`Posted` / `Paid`.	
Created By / Timestamp	System-generated	User and timestamp of creation.	Audit field.	Invoice Master
Updated By / Timestamp	System-generated	User and timestamp of update.	Audit field.	Both

Notes

- *Start Week field recalculates the Payment Schedule if changed (default = Current Week).*
- *Ledger is the single source of truth: installments are not posted directly to the DTR, only to the Ledger when due.*

---

3. States & Lifecycle

The Repairs module tracks obligations at **two levels**:

1. **Repair Invoice** – the overall obligation created when a repair is logged.
2. **Repair Installments** – the scheduled weekly deductions derived from the invoice.

3.1 Repair Invoice – States

State	Description	Transition Triggers
Draft	Invoice created but not yet confirmed. Payment Schedule not generated.	User saves invoice without confirming.
Open	Invoice confirmed, Payment Schedule generated, installments scheduled.	Invoice confirmed. Remains until balance = $0.
Closed	All installments fully posted and balance = $0.	Last installment posted, balance cleared.

Hold	Invoice flagged for review (driver challenge, incorrect billing). Installment postings are paused.	Manual action by BAT staff.
Cancelled	Invoice voided before any installment postings occur.	Manual cancellation (only possible if no postings exist).

3.2 Repair Installments – States

State	Description	Transition Triggers
Scheduled	Installment generated but not yet due.	Payment Schedule creation.
Due	Installment's Payment Period has started (Sunday 00:00) but not yet processed at 05:00 batch.	Time-based trigger at start of Payment Period.
Posted	Installment successfully posted to the Ledger at 05:00 AM.	System batch posting.
Paid *(optional)*	Installment reconciled as cleared in Ledger_Balances.	Ledger balance = 0 for this installment.

3.3 Link Between Invoice & Installments

- An **Invoice** cannot be marked **Closed** until all associated **Installments = Paid**.
- If an Invoice is put **on Hold**, all associated Installments remain in **Scheduled/Due** and do not move forward until the hold is lifted.
- **Cancelled** Invoices automatically cancel all associated Installments (if none are Posted yet).

3.4 State Transition – Repair Invoice

Lifecycle Path:

`Draft → Open → (Hold) → Closed`

↘ `Cancelled` (if voided before postings)

From State	Action / Trigger	To State
Draft	User confirms invoice	Open
Draft	User cancels invoice	Cancelled
Open	System posts installments until balance = 0	Closed
Open	BAT staff flags invoice	Hold
Hold	BAT staff clears hold	Open
Hold	BAT staff cancels invoice (if no postings yet)	Cancelled

3.5 State Transition – Repair Installments

Lifecycle Path:

`Scheduled → Due → Posted → Paid`

From State	Action / Trigger	To State
Scheduled	Payment Period begins (Sunday 00:00)	Due
Due	System batch run (Sunday 05:00) posts installment	Posted
Posted	Ledger shows installment fully cleared	Paid

---

4. Calculation Logic

The Repairs module uses a **Payment Matrix** to break down repair invoices into weekly installments. Each installment is aligned with BAT's **weekly Payment Period** (Sunday 00:00 → Saturday 23:59) and is posted to the ledger every **Sunday 05:00 AM**.

4.1 Payment Matrix Rules

Invoice Amount	Weekly Installment
$0 – $200	Paid in full (single installment)
$201 – $500	$100 per week
$501 – $1,000	$200 per week
$1,001 – $3,000	$250 per week
> $3,000	$300 per week

4.2 Start Week Option

When creating a repair invoice, the user selects whether the repayment schedule begins:

- **Current Payment Period** → First installment posts in the DTR generated the upcoming Sunday.
- **Next Payment Period** → First installment posts in the DTR generated the following Sunday.

This ensures flexibility when repairs are billed late in a week or when BAT wishes to defer the start of repayment.

4.3 Payment Schedule Generation

Once the **Repair Amount** is entered:

1. System determines the **Weekly Installment Amount** from the Payment Matrix.
2. System generates installment rows (`Installment ID`) until the invoice balance = $0.
3. Each installment is tied to a **Payment Period** (Week Start / End dates).
4. The **final installment** is adjusted automatically if the remaining balance is less than the standard installment amount.

4.4 Ledger Posting Logic

- Each Sunday 05:00 AM, BAT Connect reviews all installments scheduled for that Payment Period.
- Installments with **Week Start ≤ Current Period** and **Status = Scheduled** are:

- Marked as **Posted**.
- Written to **Ledger_Postings** (debit to driver's account).
- Linked via **Ledger Posting Ref**.

- Once posted, installments cannot be deleted — adjustments must be handled via credits/reversals.

4.5 Invoice Lifecycle

- **Open** – At least one unpaid installment remains.
- **Closed** – All installments posted, Balance = $0.
- **In Dispute** – Invoice flagged by BAT for review (blocks further postings until resolved).

4.6 Example

Repair Invoice: $1,200 (External Workshop), logged on Wednesday Oct 1.
Payment Matrix: $250 per week (since $1,001 – $3,000).
Start Week Option: Current Week.

Generated Payment Schedule (linked to Repair ID RPR-2025-012):

Installment ID	Week Start	Week End	Installment Amount	Status
RPR-2025-012-01	Sep 28	Oct 4	$250	Posted (Sun Oct 5)
RPR-2025-012-02	Oct 5	Oct 11	$250	Scheduled
RPR-2025-012-03	Oct 12	Oct 18	$250	Scheduled
RPR-2025-012-04	Oct 19	Oct 25	$250	Scheduled
RPR-2025-012-05	Oct 26	Nov 1	$200	Scheduled (final adjusted installment)

Posting:

- First installment of $250 posts to the ledger on **Sunday Oct 5, 05:00 AM**.

- Balance reduces accordingly each week until invoice closes.

Notes

- *Installments flow into the Ledger only when the Payment Period arrives (Sunday 05:00). DTR pulls from the Ledger, not from the Payment Schedule directly.*
- *In DTR Repairs Detail, it is recommended to show Original Invoice Amount and Paid Till Date for driver transparency, in addition to weekly installment deduction.*

---

5. UI Workflow

The Repairs workflow enables BAT staff to capture repair invoices, auto-generate repayment schedules, and ensure installments are posted correctly into the Ledger, with transparency for both BAT and drivers.

Step 1 – Identify Driver & Lease

- Staff enters or scans the **Driver's TLC License Number**.
- System retrieves the driver profile and all associated **Medallion / Lease accounts**.
- Staff selects the relevant **Medallion / Lease ID**.

Step 2 – Upload & Enter Repair Invoice

- Staff **uploads the repair invoice** (PDF/image).
- System uses **OCR** to capture key details:
  - Invoice Number
  - Invoice Date
  - Repair Amount (Total)
- Staff reviews the OCR results and completes/edits remaining fields as needed:
  - **Workshop Type** (Big Apple / External)
  - **Repair Description / Notes**
- **VIN, Plate, Medallion, and Hack License are already known** from Step 1 (Driver + Lease Account selection) and are automatically linked to the repair entry.

Step 3 – Generate & Confirm Payment Schedule

- On save, the system applies the **Repair Payment Matrix** to generate a repayment schedule.
- Staff must then select when repayments begin:
  - Default: **Current Payment Period**
  - Optional: **Next Payment Period**

- If Start Week is changed, the schedule is **recalculated automatically**.
- Staff reviews the full schedule (Installment IDs, Week Start/End, Installment Amounts) and confirms.

Step 4 – Schedule Storage

- The system stores:
  - **Invoice Master** (all invoice-level data).
  - **Repair Payment Schedule** (installment rows with dates, amounts, IDs).
- At this point, installments are in **Scheduled** state only — no postings yet.

Step 5 – Ledger Posting

- Each Sunday at 05:00 AM:
  - Installments whose Payment Period has started are marked **Posted**.
  - A **Ledger Posting Reference** is created.
  - Ledger_Balances is updated (balance reduced).
- Installments remain Scheduled → Due → Posted in line with their lifecycle.

Step 6 – DTR Representation

- The DTR does not pull directly from the schedule. It pulls from **Ledger_Balances** only.
- In the Repairs section of the DTR, drivers see:
  - **This Week's Deduction (Installment Posted)**
  - **Prior Balance (unpaid installments from earlier weeks)**
  - **Remaining Balance**
- Optional: add a **column or note** to show:
  - Original Invoice Amount
  - Total Paid Till Date

---

6. Sample

Assumptions

1. Payment Periods run **Sunday 00:00 → Saturday 23:59**, with postings at **Sunday 05:00 AM**.
2. Driver & Lease account already identified in Step 1 (VIN, Plate, Medallion, Hack License auto-linked).
3. OCR used to capture invoice details; staff confirmed invoice amount.
4. Start Week = **Current Payment Period** (default).
5. Repair Payment Matrix applies: $1,001 – $3,000 → $250 per week.

Repair Invoice Details

- **Repair ID:** RPR-2025-012
- **Invoice Number:** EXT-4589
- **Invoice Date:** Oct 1, 2025
- **Driver:** John Doe (Hack License #1234567)
- **Medallion / Lease ID:** MED-2025-045
- **Workshop Type:** External Workshop
- **Repair Description: Brake System Overhaul (pads, rotors, calipers)**
- **Repair Amount (Total):** $1,200
- **Start Week:** Current Payment Period

Generated Payment Schedule

Installment ID	Week Start	Week End	Installment Amount	Status
RPR-2025-012-01	Sep 28	Oct 4	$250	Posted (Oct 5, 05:00 AM)
RPR-2025-012-02	Oct 5	Oct 11	$250	Scheduled
RPR-2025-012-03	Oct 12	Oct 18	$250	Scheduled
RPR-2025-012-04	Oct 19	Oct 25	$250	Scheduled
RPR-2025-012-05	Oct 26	Nov 1	$200	Scheduled (final adjusted installment)

Ledger Posting (Week 1 – Oct 5, 2025)

- System posts Installment **RPR-2025-012-01** ($250) to Ledger_Postings.
- Ledger_Balances reduced by $250.
- Invoice remains **Open** with Balance = $950.

---

7. Validation Rules – Vehicle Repairs

The following validation rules ensure consistency between **Repair Invoices**, **Payment Schedules**, **Ledger Postings**, and the **DTR**.

7.1 Invoice-Level Validation

1. **Mandatory Fields** – Invoice Number, Invoice Date, Workshop Type, Repair Amount must be entered before confirmation.
2. **Identifiers Linkage** – VIN, Plate, Medallion, and Hack License must be auto-linked from Driver & Lease selection (Step 1).
3. **Unique Invoice Number** – Duplicate invoice numbers are not permitted for the same vehicle and date.
4. **Repair Amount Integrity** – Total Repair Amount must equal the sum of all installments generated (Payments + Final Balance).
5. **Start Week Selection** – Must align to a valid Payment Period (Current or Next).

7.2 Payment Schedule Validation

1. **Matrix Compliance** – Installments must follow the Repair Payment Matrix rules, except for the final installment, which may be adjusted.
2. **Installment ID Integrity** – Each installment must have a unique ID linked to its parent Repair ID.
3. **Period Alignment** – Week Start = Sunday 00:00, Week End = Saturday 23:59.
4. **Continuity Rule** – No gaps or overlaps in installment periods once schedule is generated.
5. **Balance Accuracy** – For each installment:

```
Balance = Repair Amount – (Sum of Payments Till Date)
```

7.3 Ledger Posting Validation

1. **Posting Trigger** – Installments are posted to the ledger only when their Payment Period has arrived (Sunday 05:00 AM).
2. **Ledger Linkage** – Every posted installment must include a Ledger Posting Reference (1:1 mapping between Installment ID and Ledger Posting ID).
3. **Reconciliation Rule** – The total of Repairs postings in the Ledger for a Payment Period must match the Repairs deductions in the DTR.
4. **Immutability** – Once posted, installments cannot be deleted. Adjustments must be made through credit or reversal entries.

7.4 DTR Validation

1. **Installment-Only Rule** – DTR must display only the installment(s) posted that week, not the full invoice.
2. **Outstanding Transparency** – DTR must display prior balance and remaining balance for driver clarity.
3. **Contextual Display** – Recommended: show Original Invoice Amount and Paid Till Date for additional transparency.
4. **Cross-Check** – Repairs section in DTR must reconcile exactly with Ledger_Postings for that period.

7.5 Lifecycle Validation

1. **Invoice Closure** – Invoice can only move to Closed when all installments are Posted (and optionally Paid).
2. **Hold Handling** – If Invoice is set to Hold, associated installments remain frozen in Scheduled/Due until hold is lifted.
3. **Cancellation Rule** – Invoices may only be cancelled before first installment posting. Cancellation automatically voids all associated Scheduled installments.
4. **Audit Trail** – Created By, Updated By, and Timestamps must be captured for both invoices and installments.
