

BR_IMS POS — MRA EIS Compliance Assessment Report

Re-verification against the 22 official MRA test cases (`POS-Test-Cases-for_external-devs.pdf`)
Assessment date: 2026-07-12 | Method: repository-wide static review, PHP syntax validation, and line-by-line re-read of every cited file against the live working tree | Prepared for: BR_IMS project record

1. Executive Summary

This report re-verifies the BR_IMS point-of-sale system against all 22 test cases defined in the Malawi Revenue Authority's (MRA) Electronic Invoicing System (EIS) certification sheet for external POS developers. The previous assessment (2026-06-21) scored 43/44 (97.7%) with one Fail and two Partial items. This re-verification finds the same overall score, 43/44, but with a different composition: the prior Fail item (VAT rate) has been fixed, while a previously-passing item (cash-drawer hardware support) has regressed to Partial after the feature was removed from the codebase with no trace in git history.

43 / 44 OVERALL SCORE (97.7%)	21 PASS	1 PARTIAL	0 FAIL
----------------------------------	------------	--------------	-----------

Certification status: Near-ready, not yet certified. Per the MRA test sheet's own pass/fail criteria, "all test cases must pass" and calculations must be "100% accurate" — code-level evidence supports this for 21 of 22 items, but live MRA API execution and physical hardware testing (printers, cash drawer, mobile/desktop devices) have not been performed, and are required before certification can be claimed.

2. Scoring Method

Each test case is scored Pass = 2 points, Partial = 1 point, Fail = 0 points, against a maximum of 2 points per test (44 points total across 22 tests). A "Pass" reflects that the required logic exists in the current code with correct behavior as read; it is implementation evidence, not proof of live MRA certification. Evidence is cited as `file:line` and was independently re-read for this assessment rather than carried over from the prior audit.

Note on codebase state: The working tree currently differs from the last git commit (`970f5c2`) in 111 files, including the core EIS classes — `classes/MraEisCompliance.php` alone carries +424/-7 uncommitted lines. This assessment evaluates the working tree as it stands, since that is the code that would actually run, not the last commit.

3. Full Scorecard — All 22 Test Cases

#	TEST CASE	VERDICT	EVIDENCE & NOTES
1	EIS terminal activation	PASS	<code>activate()</code> posts <code>/onboarding/activate-terminal</code> , stores <code>mra_terminal_id / jwt_token / secret_key</code> ; <code>confirm()</code> posts <code>/onboarding/terminal-activated-confirmation</code> — <code>classes/MraEisOnboarding.php:525, :576</code> .
2	Download site products	PASS	<code>syncSiteProducts()</code> posts <code>/utilities/get-terminal-site-products</code> , retries with refreshed token on 401, persists via <code>saveSiteProducts()</code> — <code>classes/MraEisOnboarding.php:637-723</code> .
3	B2B and B2C transactions	PASS	B2B forced online-only (blocked when offline/EIS disabled); <code>validateOnlineB2BBuyer()</code> runs before any DB write; rollback on failure — <code>views/sales/create_sale.php:214-234, :339-427</code> ; <code>classes/MraEisCompliance.php:1497-1531</code> .
4	Grouped tax rates A/B/E	PASS	EIS payload per-rate breakdown drives the receipt: extraction <code>sale_receipt.php:129-142</code> ; one VAT row per rate group rendered <code>sale_receipt.php:1025-1031</code> .
5	VAT registration status	PASS	<code>terminalVatRegistrationStatus()</code> (<code>MraEisCompliance.php:2957</code>) consumed in <code>prepareOnlineSaleContext()</code> (:1549). Receipt shows "Not VAT Registered" label + omits VAT row when applicable, "TAX INVOICE" vs "SALE RECEIPT" (:962), Seller TPIN row (:971-973).
6	MRA server time	PASS	<code>getMraServerDateTime()</code> pings <code>/utilities/ping</code> (<code>MraEisCompliance.php:2915-2955</code>); used as <code>\$saleDate</code> (<code>create_sale.php:119-121</code>). Receipt reads <code>invoiceHeader.invoiceDateTime</code> from outbox first (<code>sale_receipt.php:222-241</code>).
7	Insufficient stock prevention	PASS	Item existence/sold/wrong-branch checks before <code>createSale()</code> (<code>create_sale.php:362-397</code>), re-checked in <code>Sales.php:307-324</code> , rollback on failure (:425-427).
8	Percentage/fixed discounts	PASS	Both modes validated server-side (percentage 0-100%, fixed $\leq$ total) — <code>create_sale.php:76-84, :176-190</code> . Receipt shows original total + discount row (<code>sale_receipt.php:1009-1017</code>). Gap: <code>DiscountPolicy::getActiveRules()</code> is fetched (<code>create_sale.php:39-40</code>) but never enforced — dead wiring, does not block the base requirement.
9	VAT accuracy at 16.5%	PASS	Fixed since last audit — resolves to 16.5% in <code>config/system_settings.php:3</code> , <code>SystemSettings.php:6</code> , <code>sale_receipt.php:8</code> , <code>MraEisCompliance.php:16</code> . Cleanup remaining: unreachable ?? 17.5 fallbacks and "Defaults to 17.5%" help text linger in several invoice/quotation views and <code>views/admin/settings.php</code> — cosmetic only.
10	Tourism levy calculation	PASS	Levy rate map keyed correctly (<code>MraEisCompliance.php:3001-3004</code>), <code>levyComponent()</code> (:3023), levy breakdown in payload. Receipt renders levy rows after VAT (<code>sale_receipt.php:145-155, :1044-1049</code>).
11	Offline transaction functionality	PASS	<code>isOfflineMode()</code> / <code>assertOfflineSaleAllowed()</code> gate sales (<code>MraEisCompliance.php:1617-1672</code>), queued in <code>mra_eis_outbox</code> , synced via <code>syncOfflineSales()</code> / <code>submitOutbox()</code> . Offline signature: HMAC-SHA256 via terminal secret key (:2462-2484).
12	Offline time-based threshold	PASS	<code>offlineLimits()</code> reads <code>maxTransactionAgeInHours</code> from the activated terminal's stored config (:2414-2425), enforced in <code>assertOfflineSaleAllowed()</code>

#	TEST CASE	VERDICT	EVIDENCE & NOTES
			(:1631-1654).
13	Offline amount-based threshold	PASS	<code>pendingOfflineAmount()</code> (:2427) + <code>maxCumulativeAmount</code> check (:1656-1663). MRA's test sheet marks this test "Dropped, no longer required" — implementation remains in place regardless.
14	Configuration version refresh	PASS	<code>applyConfigVersionsFromResponse()</code> persists config versions and calls <code>downloadLatestConfiguration()</code> when <code>shouldDownloadLatestConfig</code> is true (:3038-3097, :3136-3170).
15	Receipt QR/barcode generation	PASS	QR generated only when acknowledged with a valid validation URL (<code>sale_receipt.php:181-194</code> , rendered :1097-1105).
16	QR URL validation / timestamp	PASS	"MRA Invoice Time" shown next to QR, sourced from <code>invoiceHeader.invoiceDateTime</code> (<code>sale_receipt.php:222-241</code> , :1101-1103).
17	Invoice number sequencing	PASS	<code>nextDailyTransactionSequence()</code> + <code>transactionIdentifierExists()</code> dedupe against <code>sold_items</code> and <code>mra_eis_outbox</code> (<code>Sales.php:151-189</code>).
18	Invoice number format	PASS	Format built as <code>base64(taxpayerId)-base64(terminalPosition)-base64(julianDate)-base64(sequence)</code> (<code>Sales.php:129-161</code> , :261-298). No separate rejection validator exists, but none is needed since invoice numbers are always system-generated, never user-supplied.
19	Terminal blocking	PASS	<code>assertTerminalNotBlocked()</code> runs before every online sale, re-checks MRA via <code>fetchTerminalBlock()</code> / <code>resolveTerminalBlock()</code> . <code>applyTerminalBlockFromResponse()</code> persists blocks from any submission response.
20	Relief supply handling	PASS	<code>reliefAdjustedPrices()</code> strips VAT only from standard-rated (A) items, B/E unchanged (:1593-1614). Receipt shows "RELIEF SUPPLY" label and "VAT (Relief — Exempt): MWK 0.00".
21	VAT-5 certificate validation	PASS	<code>validateVat5Certificate()</code> denies locally before any API call for an empty certificate number (:1553-1557, ahead of the API call at :1565), invoked pre-persistence from <code>create_sale.php:236-248</code> . Full certificate details forwarded into the invoice payload.
22	Device / printer / cash-drawer usability	PARTIAL	Printer paper-width sizing (30/40/50/80mm) still works (<code>sale_receipt.php:1141-1328</code>). Regression: the "Open Cash Drawer" Web Serial button credited in the prior audit has been completely removed — a repo-wide search across every PHP/JS file and the full git history returns zero matches for drawer/serial code.

4. Change Log vs. Prior Audit (2026-06-21 → 2026-07-12)

#	TEST	PRIOR	CURRENT	REASON
9	VAT accuracy at 16.5%	Fail	Pass	Default VAT rate corrected from 17.5% to 16.5% across config and settings classes. The fix recommended in the prior audit's "Priority fixes" section was applied.
22	Devices/printers/cash drawer	Pass	Partial	The cash-drawer Web Serial feature the prior audit credited no longer exists anywhere in the codebase or git history. This is a genuine regression, not a citation error.

Net effect: score is unchanged at 43/44 overall, but the underlying composition shifted — one certification blocker was resolved while a previously-working capability was lost.

5. Key Findings Requiring Follow-up

F1 — Cash-drawer hardware support has regressed (Test 22)

The previous audit credited a working "Open Cash Drawer" button using the Web Serial API to send an ESC/POS drawer-kick command (`0x1B 0x70 0x00 0x19 0xFA`) to the receipt printer port. That code is now completely absent from `views/sales/sale_receipt.php` and does not appear anywhere in the git history (2 commits total: baseline and a template migration). It was either reverted unintentionally during the shared-header/footer migration, or never actually committed despite being described as implemented.

Recommendation: Re-implement the cash-drawer trigger and re-verify against real hardware, since this is one of the explicit requirements in Test Case TC-UT-017 ("Cash drawer should work accordingly").

F2 — DiscountPolicy rules fetched but never enforced (Test 8)

`classes/DiscountPolicy.php::getActiveRules()` is called in `create_sale.php:39-40` , but the returned rules are never read again in the file. Discount validation still only enforces a flat 0-100% (percentage) / $\leq$ total (fixed) bound, not any admin-configured rule set. This does not violate the MRA test case itself (which only requires percentage and fixed-amount discounts to calculate correctly), but it is unfinished feature wiring that should be completed or removed.

F3 — New automated test suite exists but fails to run

`tests/run.php` was added since the last audit — good progress toward the previously recommended goal of automated coverage. However, running it currently throws a fatal error (`Call to a member function eisZeroRatedTaxIds() on null` at `MraEisCompliance.php:2864`). The test constructs `MraEisCompliance` via `ReflectionClass::newInstanceWithoutConstructor()` and sets the `vatPercentage` property via reflection, but never sets the `$settings` (`SystemSettings`) property that `vatComponent()` depends on.

Recommendation: Inject a real or mock `SystemSettings` instance into the `settings` property before running assertions.

F4 — Stale 17.5% fallback references (Test 9 hygiene)

Although the live VAT rate correctly resolves to 16.5% everywhere it matters, unreachable `?? 17.5` dead-code fallbacks remain in `views/create_invoice.php` , `generate_invoice.php` , `generate_invoice_pdf.php` , `invoice_receipt.php` , `quotation_receipt.php` , `request.php` , and `create_service.php` . More visibly,

`views/admin/settings.php` still shows a form default and help text reading "Defaults to 17.5%", which could mislead an administrator even though the actual configured/active value is 16.5%.

Recommendation: Update these references to 16.5% for consistency, even though they are not currently reachable in normal operation.

6. Repository Health Checks

- All cited PHP files pass `php -l` syntax validation.
- The working tree differs from the last commit (`970f5c2`) in 111 files, including the core EIS classes — this assessment evaluated the working tree, since that reflects what actually runs in production.
- Live MRA API calls, network-loss simulation, mobile device testing, physical printer testing (Bluetooth and USB), and cash-drawer testing were **not executed** as part of this assessment — it is a static code review only.

7. Priority Fixes (Ranked)

1. **Restore cash-drawer support** (Test 22) — re-add the Web Serial ESC/POS drawer-kick trigger to `sale_receipt.php` and confirm it survives future template migrations.
2. **Fix the automated test suite** (`tests/run.php`) so it exercises VAT/levy logic correctly instead of fataling on a null `settings` property.
3. **Finish or remove the dead DiscountPolicy wiring** in `create_sale.php` (Test 8).
4. **Clean up stale 17.5% references** in invoice/quotation views and the admin settings help text (Test 9 hygiene).
5. **Collect live MRA API and physical hardware evidence** — this is the only path from "code-level Pass" to actual MRA certification.

8. Certification Readiness Statement

BR_IMS is **code-complete for 21 of 22 MRA EIS test cases**, with one Partial item (hardware cash-drawer support) that has regressed and needs to be restored. No test case is currently failing. Per MRA's own pass/fail criteria — all cases must pass, calculations must be 100% accurate, offline thresholds must work as specified — the system is close to certification-ready at the code level, but **cannot be certified without live MRA sandbox/production API execution and physical hardware testing** (thermal printers at all four widths, Bluetooth printers, cash drawer, and multiple device form factors as listed in TC-UT-017).

This report was generated from a static, repository-wide review of the BR_IMS codebase and does not constitute MRA certification. It is an internal engineering assessment intended to track progress toward that certification.