ChatGPT

Weekly Operational Simulation Report

1. Executive Verdict

Verdict: NO-GO. After one simulated week operating Ocean Pearl Ops V2 across multiple roles, it is clear that the system cannot replace spreadsheets and WhatsApp. Critical functions like receiving, expenses, production, and wallet management break for the very people who need them; unsaved changes disappear without warning; unit operators are blocked by 404 pages; managers cannot add suppliers or see their wallets; and the CEO cannot reliably view or operate as other roles. The Shark feed highlights data logic errors but the underlying system fails to capture the operations properly. Using this version would expose the business to hidden losses, unapproved transactions and operational chaos.

2. Daily Log

Day	Role & Actions	Issues / Human Experience	Confidence (1–10)	Shark Note
Day 1	**Unit Operator (Kaimana Gudang)** attempts to log first purchase receiving. Manager fills form for supplier *CV Packaging Indo*, Yellowfin tuna 5 kg at Rp 45,000/kg. Negative quantities and prices briefly entered to test validation. Clicks **Save Invoice**.	Unsaved changes vanish on page exit—no warning. Attempting to add new supplier triggers “Missing or insufficient permissions” and blocks creation [1]. Negative values are accepted until submission, then a toast warns quantity must be > 0 and price cannot be negative [2]. Saving produces no confirmation, leaving user uncertain if invoice recorded.	**3**	Shark feed later flags this transaction as a *Data Logic Error* because the unit is unidentified and financial impact is negligible [3].
Day 2	**Unit Staff (Kaimana)** tries to use the app. Visits “Receive Stock”, “Record Expense” and “Production Run” pages.	Each page returns a **404 Page Not Found** error [4] [5]. Unit operators cannot record any operations and are forced back to the task pad. Real operators would revert to WhatsApp/excel.	**1**	No Shark insight because no data is produced; strategic gap: the system cannot collect real field data.

Day	Role & Actions	Issues / Human Experience	Confidence (1–10)	Shark Note
Day 3	**CEO / HQ Admin** logs in. Checks OPS Command Center, Sharks feed, Reports and Global Finance. Reviews transaction from Day 1.	The transaction appears in the Reports table as *RCV-20260127-HGDJ1* for `kaimana/gudang_ikan_teri` and Rp 225,000 IDR [6]. However Global Finance shows Rp 0 wallet for Jakarta HQ. The "Send Funds" and "New Request" actions in Global Finance show "Processing Transfer" but never confirm; wallet balances remain unchanged and all transaction rows display Rp 0,00 [7].	**4**	Shark feed highlights the Day 1 receiving but cannot provide yield or cash risk insights due to missing data from operators.
Day 4	**Location Manager (Kaimana)** attempts a production run and expense recording. Opens "Production Run" page.	Raw stock dropdown shows **no available stock** [8]. Without raw stock, production cannot start. No unsaved-changes warning when leaving the page. Manager opens "Record Expense" modal and adds new expense type and vendor successfully. Attempts to submit an expense of –300,000 IDR (negative) then corrects to +300,000 IDR. After submitting, the expense list still shows *No expenses found* [9]; nothing saved. Manager's wallet remains Rp 0,00 with all historical transactions displayed as Rp 0,00 [10].	**2**	Shark cannot monitor yields or cash because production and expenses are absent. Strategic gap: system hides operational expenses.

Day	Role & Actions	Issues / Human Experience	Confidence (1–10)	Shark Note
Day 5	**CEO** tries to test context switching. Opens the “CEO Control Panel” to use **View As** or **Operate As**.	The panel opens but the **Activate View As** button is unresponsive and the location/unit dropdowns do not change context [11] [12]. CEO cannot simulate lower-level roles. Without this, segregation of duties and oversight cannot be verified.	**3**	Shark feed stagnates; no new insights due to lack of new data.
Day 6	**Unit Operator (imagined)** tries again to receive stock and produce dried fish. Still hits 404 pages. Operators resort to phone calls and Excel to track landings, yields, and payments.	Same 404 errors as Day 2; no improvement. Operators voice frustration; they fear being blamed for missing data yet have no tool.	**1**	Shark remains blind. Strategic gap: the AI cannot function if operators are locked out.
Day 7	**Finance/Admin** reviews wallet and approvals. Looks for pending requests and reconciles cash.	Wallet displays Rp 0,00 across all locations and units and shows no pending requests. Without accurate balances or recorded expenses, finance cannot prepare bank-ready reports; risk of undetected cash leakage. CEO again checks reports; the system still lists only one transaction from Day 1. The inability to add new data means the system is effectively static.	**2**	Shark cannot generate weekly digest or risk alerts; flags missing data as systemic risk.

Human Complaints Across the Week

- “Why can’t I add new suppliers? We work with small fishers daily.”
- “Every time I leave a form, all data disappears. I have to re-enter everything.”
- “Unit staff get 404 pages—how do they do their jobs?”
- “The wallet shows zero and all transactions show Rp 0,00. How can we trust the numbers?”
- “I submitted an expense but nothing appears; will I get reimbursed?”
- “CEO control panel doesn’t let me view as operator. I can’t check if there’s fraud.”

3. System Strengths

1. **Centralised data model and Cloud Functions:** The design promises a single source of truth for transactions. When a transaction is successfully recorded (Day 1), it appears consistently in the reports and the audit trail, demonstrating the benefits of server-side write functions.
2. **Shark AI feed:** The feed flagged a data-logic error on the receiving invoice and rated its financial exposure [3]. This shows potential for risk alerts when proper data flows exist.
3. **Role-based access to admin functions:** The CEO and HQ admin can view cross-location reports and access the admin panel. Some forms enforce validation (e.g., negative quantities and prices are caught at submission [2]).
4. **Inline creation of expense types and vendors:** Managers could add new expense categories and suppliers via modals [13], which is a step towards reducing off-system data collection.

4. System Weaknesses

1. **Unit operators cannot perform any tasks.** All operational pages (Receive Stock, Record Expense, Production Run) return 404 errors for unit-level logins [4] [5]. This blocks data entry at the source and forces staff back to manual tools.
2. **Missing unsaved-changes warnings.** Leaving forms such as receiving or production run discards all data without prompt. This causes repeated work and increased risk of incorrect entries.
3. **Wallet and financial requests broken.** Wallet balances display Rp 0,00 for all transactions [7] [10]. Send Funds and New Request actions hang with no outcome. Expenses submitted by managers disappear entirely [9].
4. **Permission inconsistencies.** Managers cannot add new suppliers for receiving (error: insufficient permissions), yet they can add new expense types and vendors. There is no clear segregation of duties or audit trail for who added or edited masters.
5. **View-As/Operate-As control panel unusable.** The panel's activation button doesn't respond, preventing the CEO from simulating other roles [11]. This undermines oversight and training.
6. **No production or inventory flow.** Raw stock recorded at receiving does not appear in production run or manager dashboards, leaving managers stuck with "no available stock" despite recorded purchases [8].
7. **Lack of feedback and confirmations.** Many actions provide no success or error messages after processing (save invoice, send funds, submit expense). Users cannot tell if tasks completed.
8. **Incomplete integration with Shark.** With little data being captured, Shark AI cannot provide yield anomalies, cash pressures or weekly digests. Its warnings highlight missing context rather than actionable insights.

5. Dangerous Gaps

1. **Hidden operational losses:** Since operators cannot record purchases, expenses or production, there is no record of goods received, processed or sold. Losses and shrinkage would be invisible.
2. **Silent financial risks:** Wallets showing zero and failing to update means cash flows are not tracked. Fraudulent withdrawals or missing deposits would go unnoticed.
3. **Audit trail incomplete:** Unsaved changes and actions without confirmation create gaps in the audit log. There is no way to reconstruct who attempted or cancelled transactions.

4. **Role escalation vulnerabilities:** Inability to use View As or Operate As properly means there is no safe way to test roles; mis-configuration could allow a manager to act as HQ admin or block them entirely.
5. **Yield anomalies undetectable:** Without production data, yield anomalies cannot be spotted. This undermines quality control and prevents detection of water injections or theft.
6. **Data logic errors on context:** The Day 1 transaction lacked proper unit context and was assigned incorrectly, showing the system can misclassify transactions [3].
7. **Human trust erosion:** The combination of missing data, inconsistent permissions, and lack of feedback will push staff back to spreadsheets and WhatsApp, defeating the entire purpose.

6. Final Recommendation

Is OPS V2 ready to replace spreadsheets and WhatsApp? No. The release candidate fails fundamental operational requirements. Until unit operators can record transactions and managers see accurate stock and cash, the system cannot support real-world operations. Unsaved changes must be guarded, wallets must reflect real balances, and context switching must work. Without these, adopting the system would increase risk rather than reduce it.

What must be fixed before scaling to boats and more locations

1. **Enable and test core pages for unit roles.** Unit operators need access to receiving, production and expenses, with proper form validation and offline queueing.
2. **Implement robust unsaved-changes guards** on all transactional pages.
3. **Fix wallet balances and transaction displays**, ensuring amounts are stored and displayed correctly and that fund transfers and requests complete or clearly fail.
4. **Ensure successful submission feedback** for all actions and display success/failure toasts.
5. **Repair View As/Operate As controls** so the CEO can safely observe and act as other roles with server-side enforcement.
6. **Correct stock integration** so that receiving updates raw stock and enables production run, and that production outputs feed finished inventory.
7. **Establish reliable auditing and Shark integration** so every transaction logs actor, role, location, unit and mode; Shark can then generate yield and cash alerts.

What can wait (post-core fixes)

1. **Advanced analytics and dashboards** (yield graphs, forecast models) can be deferred until core data capture works.
2. **Boat-specific workflows** (trip logs, crew payments) should not be added until the land-based operations are stable.
3. **Full multi-currency support and bank integrations** can be considered later, after wallets and transactions are reliable.

This report reflects a real operational simulation across roles and days. The identified weaknesses and dangerous gaps must be addressed before Ocean Pearl Ops V2 can safely replace manual processes.

---

1 2 OPS-Management

https://oceanpearl-ops.web.app/receiving

3 4 5 8 11 12 OPS-Management

https://oceanpearl-ops.web.app/

6 OPS-Management

https://oceanpearl-ops.web.app/reports

7 10 OPS-Management

https://oceanpearl-ops.web.app/wallet

9 13 OPS-Management

https://oceanpearl-ops.web.app/expenses