

# Data Quality Assessment Report

Company: MedTrack Ghana

Prepared by: **Joseph Lubandi** (Python Backend Developer - Kigali)

Date: February 26, 2026

Subject: Investigation of Patient Appointment Database Issues

## Summary:

This report investigates a sample of 50 patient appointment records to identify data quality issues causing operational failures, including undelivered SMS reminders, inaccurate patient reports, and billing errors. The analysis reveals systemic weaknesses across all six key data quality dimensions. These issues directly impact Operations, Finance, and Clinical functions. This document outlines the specific problems, their business impacts, and proposes actionable technical solutions for the three most critical issues to restore data integrity and operational reliability.

## Task 1: Identification of Data Quality Issues

After examining the provided dataset, the following violations across all six data quality dimensions have been documented.

Dimension	Definition	Example from Dataset	Description of Violation
Accuracy	Data that is factually incorrect.	P002, Ama Serwa, 244789012, 15/10/2025, dr. osei, paid	The phone number 244789012 is missing the leading '0' and has only 9 digits. A valid Ghanaian number typically has 10 digits and starts with '0' (e.g., 0244789012).
Completeness	Missing required information.	P004,, 0555234567, 2025-10-17, Dr. Mensah, Paid	The PatientName field is completely empty for record P004. This is critical identifying information.

---

Consistency	Same data represented differently across the dataset.	<code>Dr. Osei</code> vs <code>dr. osei</code> <code>Paid</code> vs <code>paid</code> <code>2025-10-15</code> vs <code>15/10/2025</code> vs <code>10/16/2025</code>	The same doctor's name and payment status are entered with inconsistent casing. Multiple date formats are used, making data unreliable for sorting and analysis.
Timeliness	Data that is outdated or has wrong dates, making it unreliable for current use.	<code>P002, Ama Serwa,</code> <code>244789012, 15/10/2025,</code> <code>dr. osei, paid</code>  <code>vs</code>  <code>P002, Ama Serwa,</code> <code>0244789012, 2025-10-15,</code> <code>Dr. Osei, Paid</code>	The first record for patient P002 contains an old/incorrect phone number, while a later record shows a valid one. This indicates contact information is not being kept current.

---

---

Validity	Data that doesn't follow the required format rules.	244789012 15/10/2025	The phone number 244789012 fails the 10-digit format rule. The date 15/10/2025 fails the system's expected ISO format (YYYY-MM-DD), which will cause parsing errors.
Uniqueness	Unnecessary duplication of records.	P001, Kwame Mensah, 0244123456, 2025-10-15, Dr. Osei, Paid  P001, Kwame Mensah, 0244123456, 2025-10-20, Dr. Adjei, Pending  P002, Ama Serwa, ... , 2025-10-15, Dr. Osei, Paid (two identical records)	Patient P001 has two distinct appointments, which is valid. However, patient P002 has two identical records for the same appointment on the same date with the same doctor. This is a true duplicate that will inflate counts.

---

## Task 2: Assessment of Business Impact

Each data quality issue identified has a direct and negative impact on MedTrack Ghana's operations.

Quality Issue	Operational Problem	Business Function Affected
Accuracy/Validity (Invalid Phone Numbers)	SMS reminder messages fail to be delivered or are sent to the wrong person. This leads to missed appointments and reduced clinic efficiency.	Operations (Patient Engagement)
Completeness (Missing Patient Name)	Billing cannot be processed without a patient's name. Staff cannot properly check in the patient, leading to confusion and delays in clinical care.	Finance & Clinical

---

Consistency (Inconsistent Doctor Names)	Reports that analyze doctor performance or patient load will split one doctor's data into multiple entries (e.g., "Dr. Osei" and "dr. osei"). This leads to inaccurate workload analysis and poor management decisions.	Operations / Clinical (Reporting & Resource Allocation)
Timeliness (Outdated Phone Number)	SMS reminders are sent to an old or incorrect number, causing the same operational failures as an invalid number. This undermines patient trust.	Operations (Patient Engagement)
Validity (Invalid Date Format)	The system fails to parse dates correctly, leading to appointment scheduling errors, reminders being sent on the wrong day, and incorrect billing dates.	Operations & Finance

---

---

Uniqueness (Duplicate Records)	Patient counts in reports are inflated, giving a false picture of growth. It can also lead to billing attempting to charge a patient twice or failing to reconcile a single payment against two entries.	Finance / Management (Reporting & Billing)
--------------------------------	--	--

---

### Task 3: Recommended Solutions

The three most critical issues are: invalid phone numbers, duplicate records, and data inconsistency. The following solutions are proposed.

#### 1. Phone Number Validation and Standardization (Addressing Validity & Accuracy)

- Technical Solution:
  - Backend Validation: Implement a regular expression (regex) validation rule in the backend API: `^0\d{9}$`. This ensures all phone numbers are exactly 10 digits and start with a '0'.
  - Data Cleaning Script: Run a one-time Python or SQL script to clean existing data. The script should remove non-numeric characters and add a leading '0' to 9-digit numbers that start with a valid prefix (e.g., '24').
  - Form Validation: Update the frontend appointment form to use a library like `intl-tel-input` which provides country-specific phone number validation and formatting.
- Responsible Party: Backend Developer (for API/database rules) and Frontend Developer (for form validation).

- Verification:
  - Run a query on the database to confirm 100% of phone numbers match the `^0\d{9}$` pattern.
  - Conduct a user acceptance test (UAT) by attempting to submit the form with invalid numbers (e.g., 9 digits, letters); the system should reject them.
  - Monitor SMS delivery failure rates for a week post-fix; they should drop significantly.

## 2. Duplicate Record Detection and Prevention (Addressing Uniqueness)

- Technical Solution:
  - Database Constraint: Add a unique composite key constraint on the `appointments` table for the combination of `PatientID` and `AppointmentDate`. This prevents the same patient from being booked twice for the same day.
  - Deduplication Script: Write a script to identify and resolve existing duplicates. For records with the same `PatientID`, `AppointmentDate`, and `DoctorName`, the script should keep the record with the most complete information or the latest entry.
- Responsible Party: Database Administrator (DBA) / Backend Developer.
- Verification:
  - Run a query to check for any remaining duplicate appointments after the script is executed (e.g., `SELECT PatientID, AppointmentDate, COUNT(*) FROM appointments GROUP BY PatientID, AppointmentDate HAVING COUNT(*) > 1`). The result should be zero.
  - Test the application by attempting to insert a duplicate appointment; the database should throw an error, and the application should handle it gracefully.

### 3. Data Standardization (Addressing Consistency & Validity)

- Technical Solution:
  - Controlled Vocabularies: Replace free-text input fields with dropdown menus. Create a master `doctors` table and link the `DoctorName` field via a foreign key. Change the `PaymentStatus` field to an `ENUM` type with allowed values ('`Paid`', '`Pending`', '`Failed`').
  - Date Standardization: Update all data entry points (forms, APIs) to accept and store dates strictly in ISO 8601 format (`YYYY-MM-DD`). Use a date picker UI component to eliminate manual text entry.
  - Normalization Script: Run a script to convert all existing dates to `YYYY-MM-DD` and update all doctor names to a consistent case (e.g., Title Case).
- Responsible Party: Backend Developer (for database changes and script) and UI/UX Designer (for implementing dropdowns/date pickers).
- Verification:
  - Run a query to show distinct `DoctorName` values; it should return each doctor only once in a clean format.
  - Check the `PaymentStatus` column for any values not in the defined `ENUM` list.
  - Query the database to ensure all `AppointmentDate` values are stored in a single, consistent `DATE` format, allowing for correct chronological sorting.

## Task 4: The Biggest Risk of Poor Data Consistency

From a backend developer's perspective, the biggest risk of poor data consistency is the creation of unreliable system logic and increased technical debt.

When the underlying data is inconsistent—for example, having multiple representations of the same doctor ("Dr. Osei", "dr. osei") or date formats ("2025-10-15", "15/10/2025")—developers are forced to write complex, fragile, and repetitive code to clean and parse the data before any business logic can be applied. This leads to:

- **Brittle Code:** Every script, report, or integration must include error-prone "cleanup" logic, making the system difficult to maintain and debug.
- **Integration Failures:** When sending data to external systems (like a billing platform or an SMS gateway), inconsistent formats will cause hard-to-diagnose failures and data rejection.
- **Erosion of Trust:** If developers cannot trust the data source, they cannot build reliable features. Management will eventually lose faith in the reports the system generates, leading to poor strategic decisions and a reliance on manual, error-prone workarounds.