JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

SCHOOL OF ELECTRICAL, ELECTRONIC AND INFORMATION ENGINEERING

DEPARTMENT OF ELECTRONIC AND COMPUTER ENGINEERING

DBMS PROJECT REPORT

GROUP 5 MEMBERS

NAME	ADMISSION
GEORGE JOHN OCHIERE	ENE212-0281/2021
MUGAISI EUGENE MURULI	ENE212-0083/2020
KOECH CINDY CHEBET	ENE212-0069/2020
KIRUI ABIGAEL CHEROTICH	ENE212-0070/2020
IAN KIOGORA MWENDA	ENE212-0251/2021

COURSE: B.Sc. ELECTRONIC AND COMPUTER ENGINEERING

ICS 2206

INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS

TITLE: KENYA POWER ELECTRICAL SYSTEMS MANAGEMENT APPLICATION

LECTURER: MR. PETER M KARANJA

DATE OF SUBMISSION: JANUARY 23RD, 2026

Table of Contents

PART I: PROJECT DOCUMENTATION

1. Executive Summary

1.1 Project Overview

The Kenya Power Electrical Systems Management Application is a comprehensive web-based solution designed to manage electrical utility operations for Kenya Power. The system provides two distinct portals:

1. **Staff Portal:** For Kenya Power employees (administrators, managers, technicians, and customer service agents) to manage customers, electrical connections, fault reports, maintenance schedules, and generate performance reports.
2. **Customer Portal:** A self-service interface for Kenya Power customers to view their connections, report faults, submit service requests, communicate with customer support, and manage their profiles.

1.2 Key Features

- **Customer Management:** Complete CRUD operations for customer accounts with unique account numbering
- **Connection Management:** Track electrical connections/meters with status management
- **Fault Management:** Report, assign, track, and resolve electrical faults
- **Maintenance Scheduling:** Schedule and track preventive and corrective maintenance
- **Performance Reporting:** Analytics dashboards with charts and KPIs
- **Role-Based Access Control:** Four staff roles with granular permissions
- **Real-Time Notifications:** System notifications for important events

1.3 Technology Stack

Technology	Purpose	Justification
Python Flask	Web Framework	Lightweight, flexible, extensive ecosystem
MySQL	Database	Industry-standard RDBMS, ACID compliance
SQLAlchemy	ORM	Pythonic database operations, relationship management
Flask-Login	Authentication	Secure session management for staff users
Bootstrap 5	UI Framework	Responsive design, modern components
Jinja2	Templating	Server-side rendering, template inheritance

2. System Architecture

2.1 High-Level Architecture

The application follows a three-tier architecture pattern:

```
                    CLIENT LAYER
    Staff Portal (Bootstrap 5)  |  Customer Portal (Bootstrap 5)
                              ↓
                  APPLICATION LAYER (Flask)
            Routes | Models | Utils | Templates
                              ↓
                    DATA LAYER (MySQL)
  kenya_power_db: users | customers | connections | faults | maintenance
```

2.2 Blueprint Architecture

The application is organized into 9 blueprints:

Blueprint	URL Prefix	Purpose
auth_bp	/	Staff authentication (login, logout, register)
main_bp	/	Landing page and dashboard
customers_bp	/customers	Customer account management
connections_bp	/connections	Electrical connection management
faults_bp	/faults	Fault reporting and tracking
maintenance_bp	/maintenance	Maintenance scheduling
reports_bp	/reports	Performance reporting and analytics
staff_bp	/staff	Staff member management (admin only)
customer_bp	/portal	Customer self-service portal

3. Database Design

3.1 Entity-Relationship Diagram

The following diagram illustrates the complete database schema with all 10 interconnected tables:

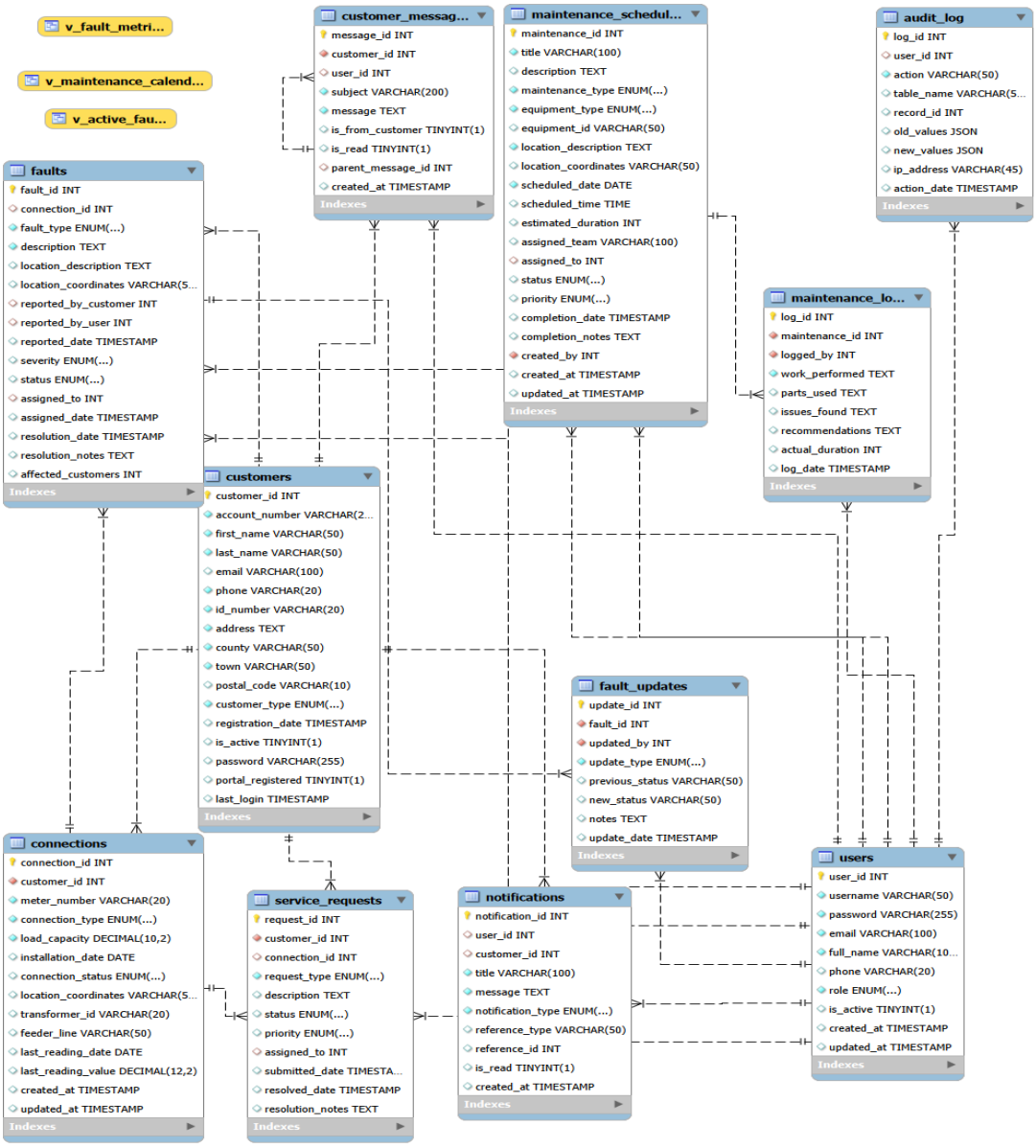

Figure 1: Entity-Relationship Diagram

3.2 Database Tables Overview

Table	Description
users	Staff/admin accounts for system authentication and authorization
customers	Kenya Power customer information and portal credentials
connections	Electrical connections/meters linked to customers
faults	Electrical fault reports with full lifecycle tracking
fault_updates	Status changes and notes added to fault reports
maintenance_schedules	Scheduled maintenance activities with assignments
maintenance_logs	Work performed during maintenance activities
service_requests	Customer service requests (new connection, upgrade, etc.)
notifications	System notifications for users and customers
customer_messages	Two-way messaging between customers and support staff

3.3 Key Relationships

- **Customer → Connections:** One-to-Many (a customer can have multiple electrical connections)
- **Connection → Faults:** One-to-Many (a connection can have multiple fault reports)
- **Fault → FaultUpdates:** One-to-Many (a fault can have multiple status updates)
- **User → Faults:** One-to-Many (technicians are assigned to faults)
- **MaintenanceSchedule → MaintenanceLogs:** One-to-Many (maintenance can have multiple work logs)

4. Authentication and Authorization

4.1 Dual Authentication System

The application implements two separate authentication systems:

Staff Authentication (Flask-Login)

- **Library:** Flask-Login with server-side session management
- **User Model:** User class extending UserMixin

Customer Authentication (Custom Session)

- **Implementation:** Custom session-based authentication in app/utils/customer_auth.py
- **Storage:** Flask session with customer_id tracking

4.2 Role-Based Access Control Matrix

Role	Customers	Connections	Faults	Maintenance	Reports	Staff
Admin	Full	Full	Full	Full	View	Full
Manager	Full	Full	Full	Full	View	None
Technician	View	Add/Edit	Assigned	Assigned	None	None
Customer Svc	Full	View	Report	View	None	None

PART II: USER GUIDE & VERIFICATION

5. Design Objectives Verification

This section details each design objective with step-by-step procedures to demonstrate compliance.

5.1 Customer Connection Management

OBJECTIVE: Track customer connections, handle connection requests, manage installation dates, and monitor service status.

Requirement	Status	Implementation
Track customer connections	**MET**	Connection model with full lifecycle tracking
Connection requests	**MET**	ServiceRequest model with new_connection type
Installation dates	**MET**	installation_date field in Connection model
Service status	**MET**	connection_status: pending/active/suspended/disconnected

HOW TO PROVE THIS OBJECTIVE IS MET:

3. Login as admin at /login

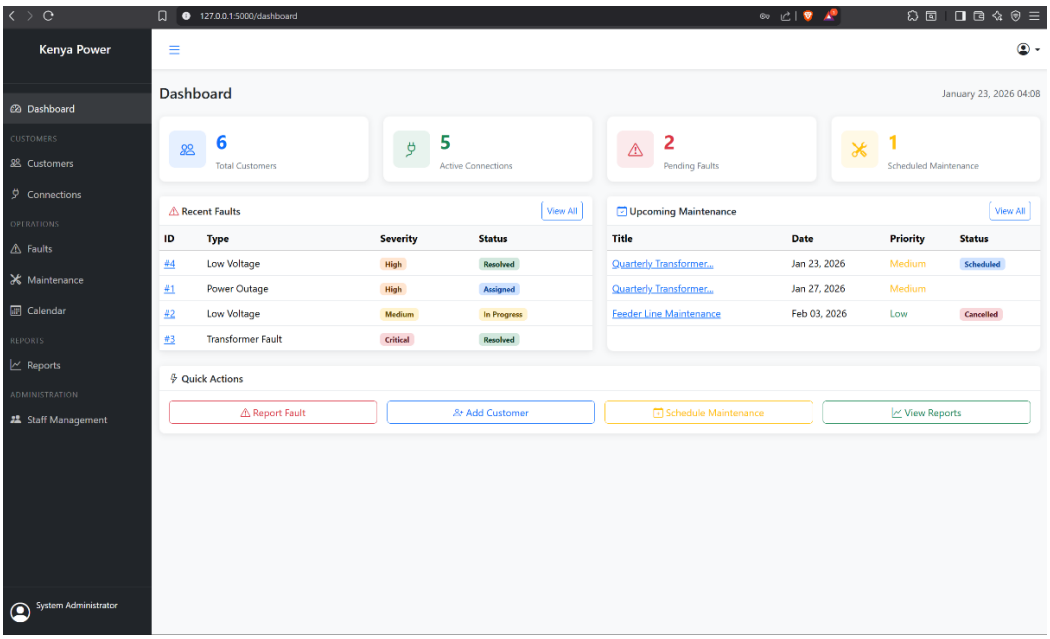

Figure 2: System Administrator's Dashboard

4. Navigate to Customers > Add Customer to create a new customer

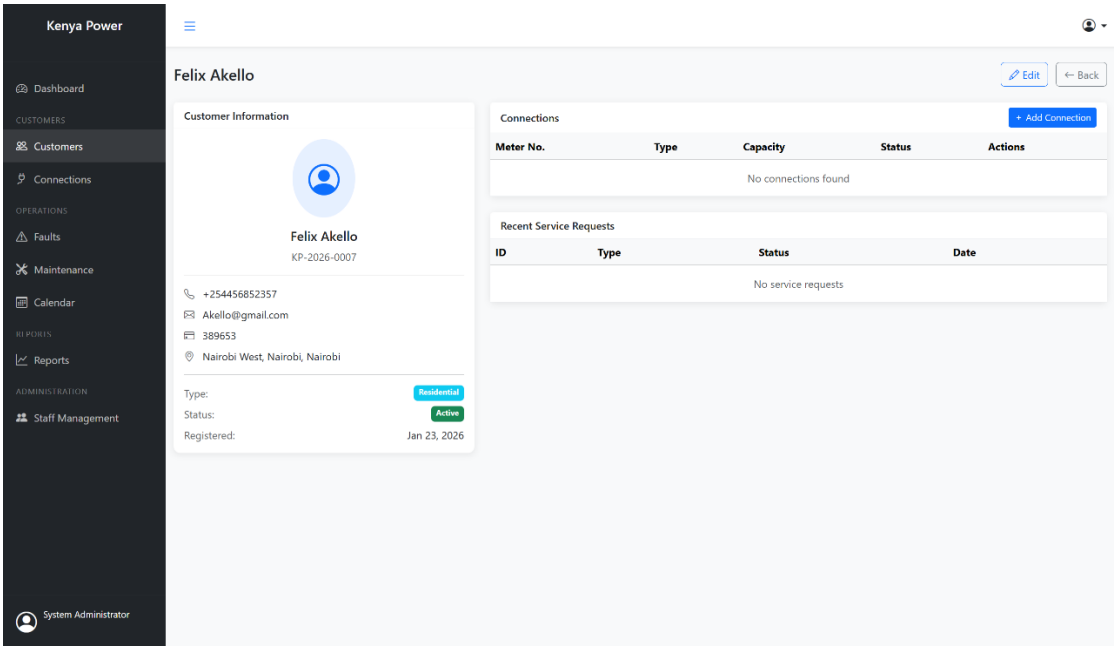

Figure 3: New Customer Created

5. Observe the auto-generated account number (format: KP-2026-XXXX)

6. Navigate to Connections > Add Connection for the new customer

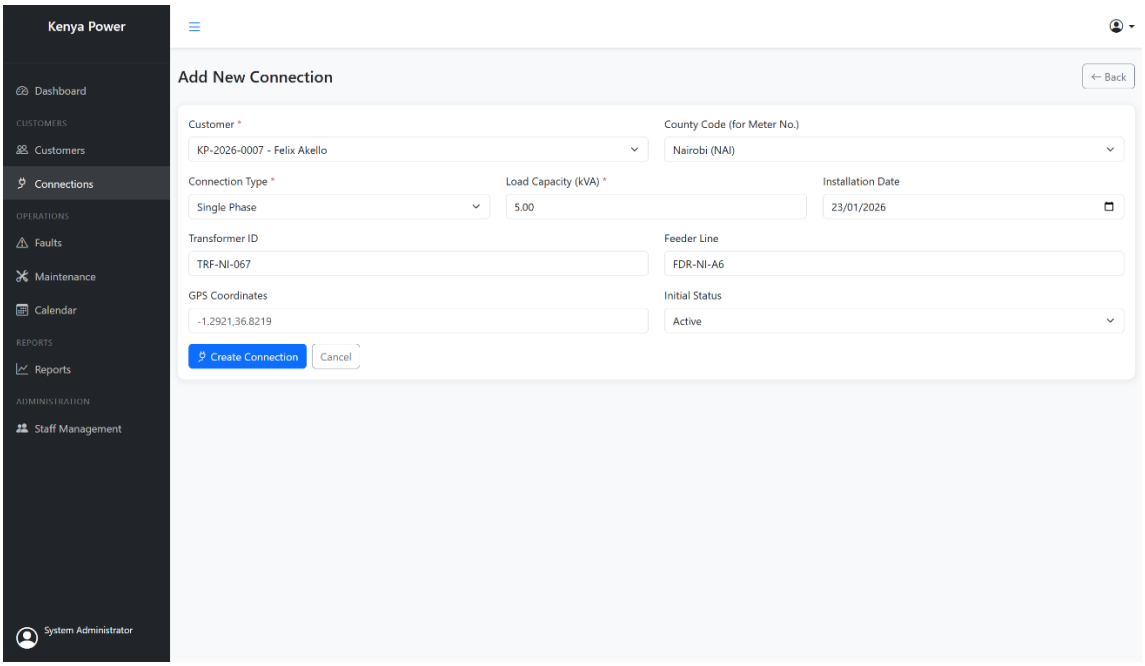

Figure 4: Application for connection for new customer

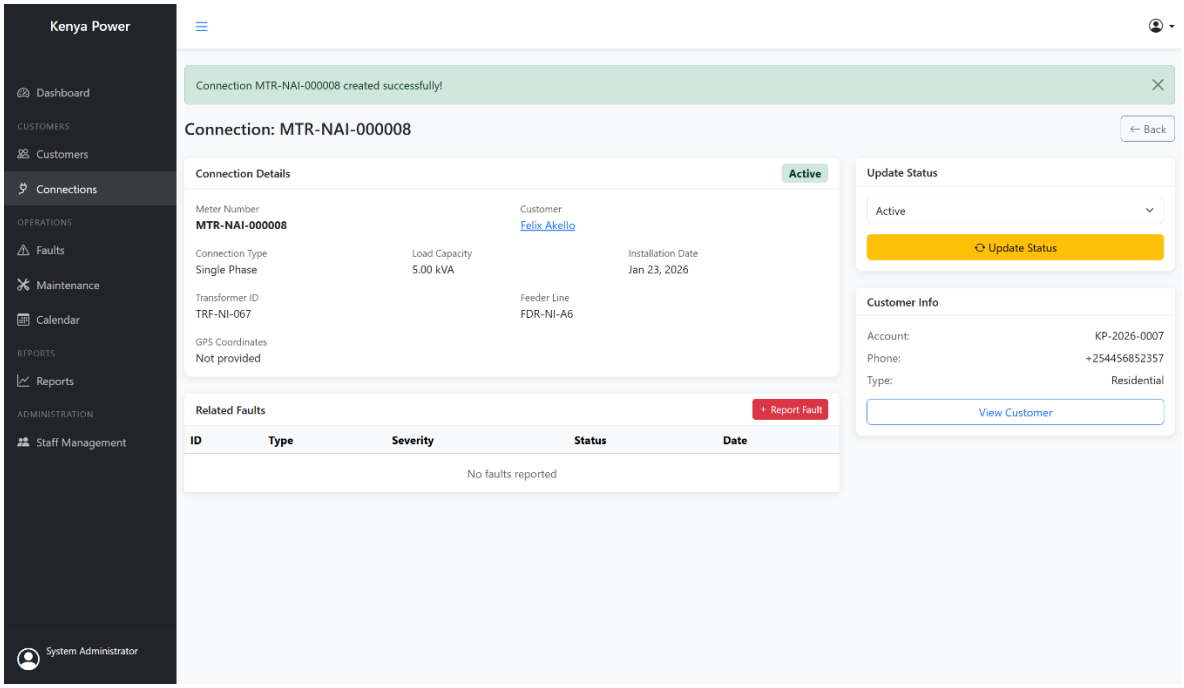

Figure 5: Coonection for customer created

7. Observe the auto-generated meter number (format: MTR-XXX-NNNNNN)
8. Demonstrate status lifecycle: pending > active > suspended > disconnected

5.2 Fault Reporting and Resolution

OBJECTIVE: Enable fault reporting, log repair requests, track resolution process, and ensure timely repairs.

Requirement	Status	Implementation
Report faults	**MET**	Staff: /faults/report, Customer: /portal/faults/report
Log repair requests	**MET**	FaultUpdate model tracks all changes
Track resolution	**MET**	6-stage workflow: reported → resolved → closed
Timely repairs	**MET**	Resolution time calculation, technician notifications

HOW TO PROVE THIS OBJECTIVE IS MET:

9. Login as customer service agent (cs_agent1 / password123) or as any customer

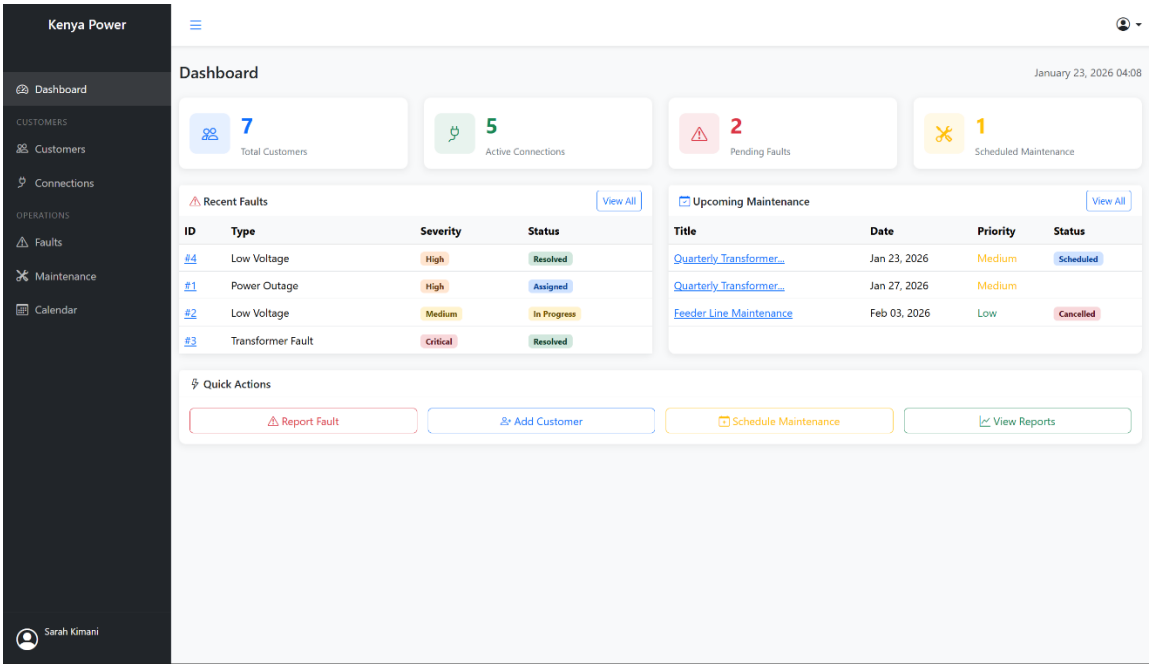

Figure 6: Customer service agent dashboard

10. Navigate to Faults > Report Fault and submit a new fault

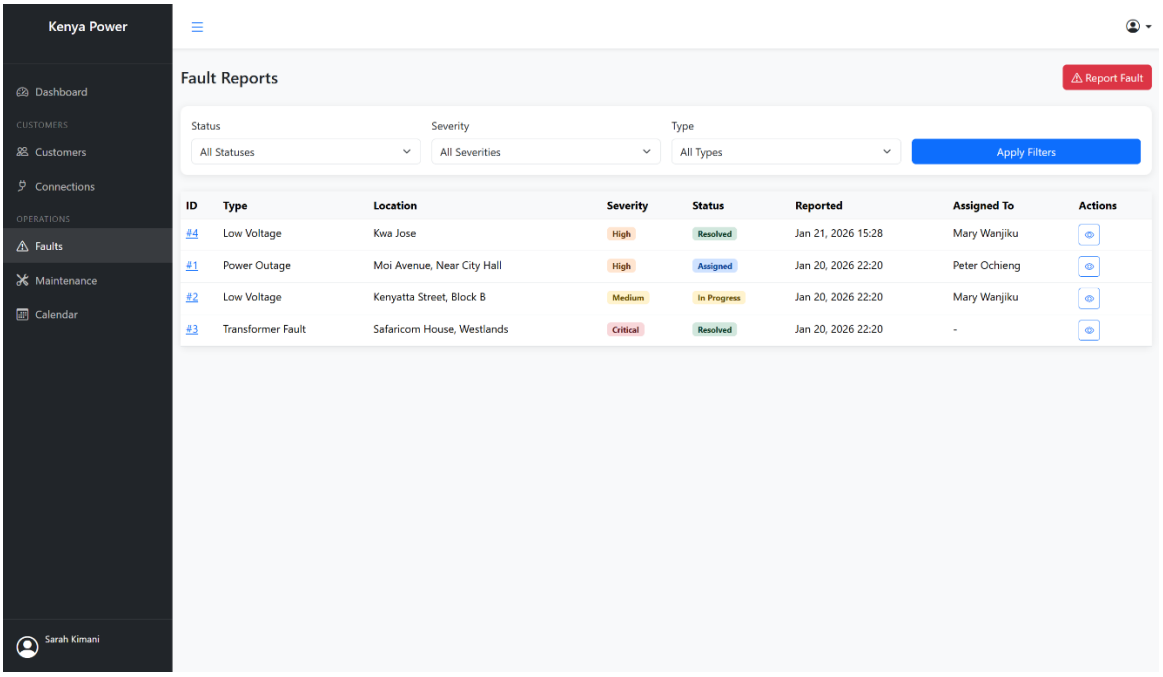

Figure 7: Fault dashboard

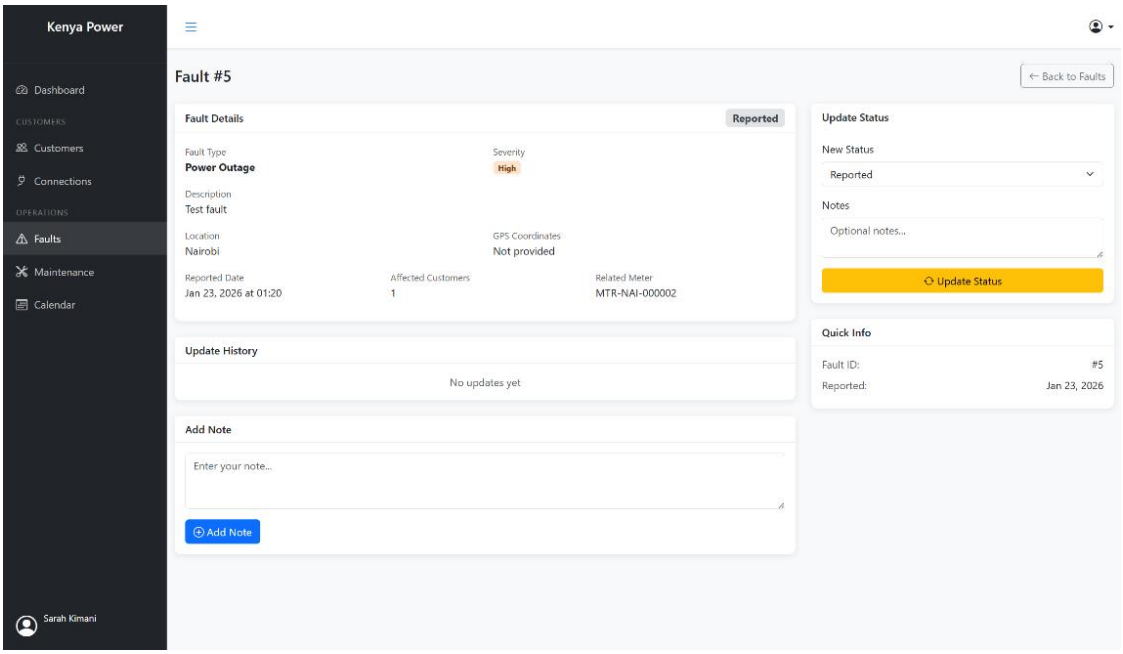

Figure 8: New Fault created

11. Login as manager (manager1 / password123) and assign fault to technician

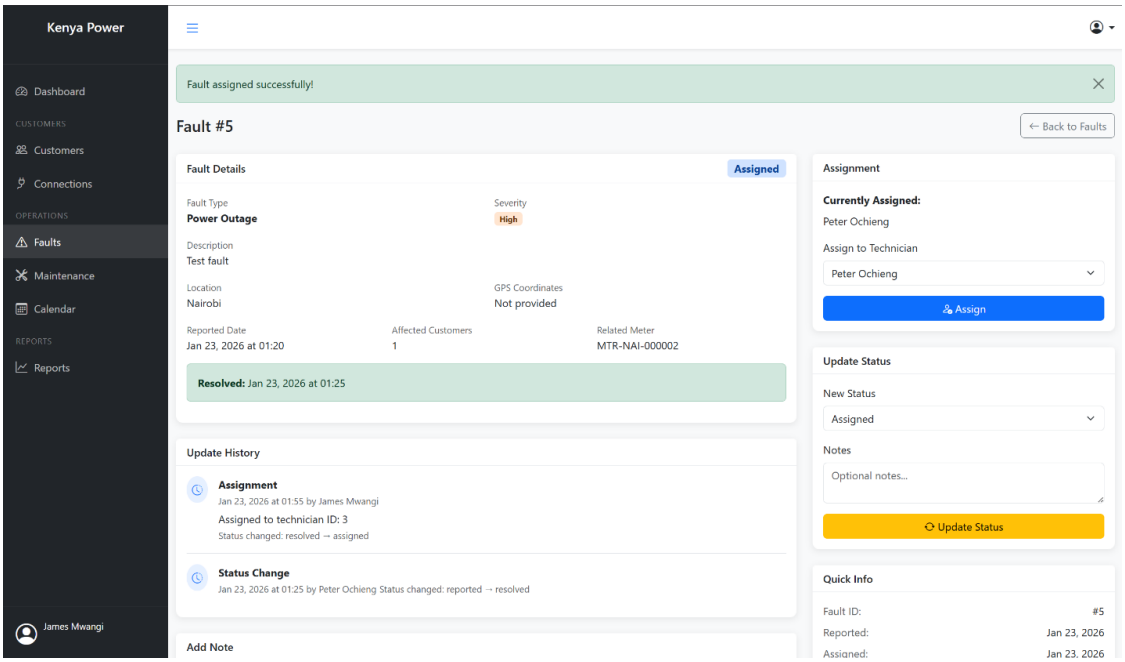

Figure 9: Fault assigned to technician by manager

12. Login as technician (tech1 / password123) and update status to "Resolved"

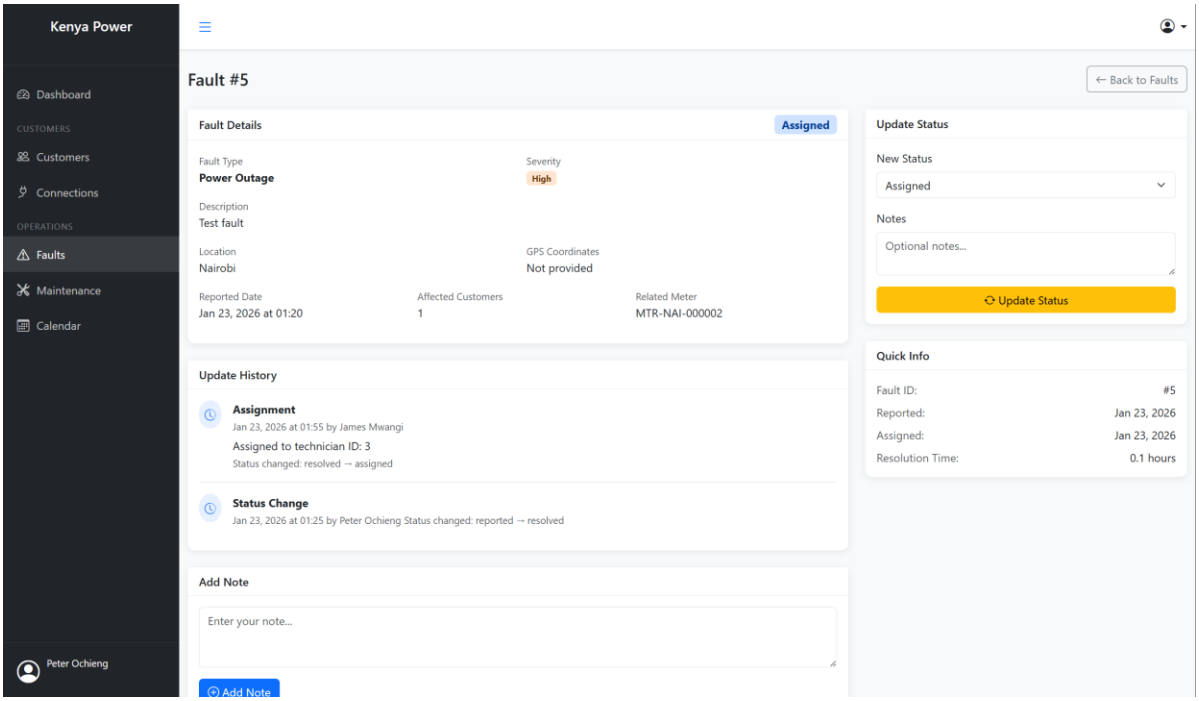

Figure 10: Fault status updated by assigned technician

13. View fault details showing resolution time calculation

5.3 Maintenance Scheduling

Requirement	Status	Implementation
Preventive maintenance	**MET**	MaintenanceSchedule model with type="preventive"
Distribution lines	**MET**	equipment_type: feeder_line, transformer, etc.
Reduce outage risk	**MET**	Calendar view, priority levels, assignments

5.4 Performance Reporting

Requirement	Status	Implementation
Maintenance reports	**MET**	/reports/maintenance with completion rates
Fault resolution times	**MET**	/reports/faults with avg resolution calculation
Performance metrics	**MET**	/reports/performance with KPI dashboard
Visual representations	**MET**	Chart.js: doughnut, bar, line charts

6. Default Credentials & Quick Reference

6.1 Staff Login Credentials

Role	Username	Password	Access Level
Administrator	admin	Password123	Full system access
Manager	manager1	password123	Operations + Reports
Technician	tech1	password123	Assigned work only
Technician 2	tech2	password123	Assigned work only
Customer Service	cs_agent1	password123	Customer management

6.2 Application URLs

Function	URL
Landing Page	http://127.0.0.1:5000/
Staff Login	http://127.0.0.1:5000/login
Staff Dashboard	http://127.0.0.1:5000/dashboard
Customer Portal Login	http://127.0.0.1:5000/portal/login
Customer Registration	http://127.0.0.1:5000/portal/register
Reports Dashboard	http://127.0.0.1:5000/reports/

6.3 Quick Start Guide

14. **Initialize Database:** `python init_db.py`
15. **Start Application:** `python run.py`
16. **Access Application:** Open browser to http://127.0.0.1:5000/
17. **Login as Admin:** Username: admin / Password: admin123

6.4 Demonstration Checklist

- Start application: python run.py
- Login as admin to show full access
- Create a customer (observe auto-generated account number)
- Create a connection (observe auto-generated meter number)

- Report a fault and assign to technician
- Schedule maintenance and view calendar
- View reports and charts
- Login as different roles to show access restrictions
- Access customer portal to demonstrate self-service features

---

This documentation was prepared for academic assessment purposes.

Kenya Power Electrical Systems Management Application

Comprehensive Project Documentation & User Guide