Request for Proposal (RFP) for Digital Platform

1. Introduction

Equatorial Power (EP) is a mini-grid developer with a strong track record in renewable energy and access to productive energy projects in Africa. EP operates in Uganda, Rwanda and DRC and is headquartered in Kampala, with a presence in Kigali and Goma.

EP developed a unique model to address the power deficit whilst also adding value to agricultural and other processing value chains by developing independent Productive Hubs (PHs) which are owned and operated by EP but which offer services such as wi-fi, low-cost briquette manufacture, cold-chain and e-mobility, creating an anchor load for the mini-grid whilst closing gaps in local value chains.

EP has so far connected over 7,500 households and businesses to renewable electricity in east Africa and is currently operating 8 mini-grids and 2 Productive Hubs.

Through this RFP EP seeks to partner with a qualified vendor to develop and implement a cutting-edge digital platform that will support and enhance the operations of EP projects and future endeavors.

Wekeza Energy Limited is a proud technical partner of EP in this endeavor. Wekeza is committed to supporting EP in the successful implementation of this project, leveraging its expertise in renewable energy project development and management, productive use of energy, grid integration, financial modeling, big data and software development and as such will ensure the project's viability and sustainability.

2. Project Goals

The main goals to be achieved by the selected vendor are:

- **Enhance Operational Efficiency:** Streamline operations, reduce operational costs, and improve overall efficiency across all aspects of EP's business.
- **Improve Customer Experience:** Enhance customer satisfaction and loyalty through improved service delivery, efficient payment processing, and responsive customer support.

- **Increase Revenue and Sustainability:** Increase revenue through new service offerings, improve payment collection rates, and ensure the long-term sustainability of EP's operations.
- **Modernize Operations:** Leverage technology to modernize electricity distribution, customer management, and operational processes.
    - **Efficient Customer Management:** Create, update, and manage customer profiles, assign unique IDs, and track usage and payment history.
    - **Seamless Meter Management:** Assign smart meters to customers, monitor energy consumption, and remotely control meter operations.
    - **Value-Added Service Delivery:** Enable EP to offer additional products and services on electricity vending such as data, briquettes, insurance and financing, to enhance customer value and revenue streams.
    - **Secure Payment Processing:** Integrate with mobile money providers to facilitate secure payments, process transactions, and generate financial reports.
    - **Real-time Insights:** Provide real-time insights into system performance and generate customizable reports from a single source of truth
    - **Robust User Management:** Implement strong user authentication and authorization to control access to system functionalities.
- **Data-Driven Decision Making:** Utilize data analytics to gain valuable insights into customer behavior, energy usage, and system performance, enabling data-driven decision-making.
- **Community Development:** Empower local communities through access to reliable and affordable energy, promote economic growth, and improve livelihoods.

3. Scope of Work

These are our specific tasks and deliverables required from the supplier.

3.1 System Design and Development

- **System Architecture:** Design a scalable and robust system architecture, including database design, API design, and security considerations.
- **User Interface:** Develop a user-friendly web interface for system administrators and customer service representatives.

- **USSD Application:** Develop a user-friendly USSD application for customers to purchase EP's goods and services such as electricity.
- **Mobile Application (Optional):** Develop a mobile app for customers to manage their accounts, view usage, and make payments.
- **Data Flow Diagrams:** Create detailed data flow diagrams to illustrate the flow of information between system components.
- **API Integration:** Develop APIs to integrate with mobile money providers, smart meter systems and third party applications.
- **Security:** Implement robust security measures, including data encryption, access controls, and regular security audits.

3.2 Mobile Money Integration

- **Identify and Select Providers:** Research and select suitable mobile money providers based on their coverage, reliability, and security standards.
- **API Integration:** Integrate with the selected mobile money providers' APIs to enable secure and efficient payment processing.
- **Payment Reconciliation:** Develop a system to reconcile payments from different mobile money providers.

3.3 Smart Meter Integration

- **API Integration:** Integrate with the smart meter system's API to retrieve meter readings, trigger remote operations, and update customer accounts.
- **Data Synchronization:** Ensure seamless synchronization of meter data with the platform's database.
- **Error Handling and Retry Mechanisms:** Implement robust error handling and retry mechanisms to ensure reliable data transfer.

3.4 Data Management and Reporting

- **Data Storage:** Design and implement a scalable database to store customer information, meter readings, payment history, and system logs.
- **Data Security:** Implement data encryption and access controls to protect sensitive information.

- **Reporting:** Develop reports on customer usage, payment trends, and system performance.

3.5 Ticketing System

- **Ticket Creation:**
    - Allow EP agents to submit tickets through one or various channels (e.g., web portal, USSD, mobile app).
    - Integrate ticket creation with customer accounts and payment history.
- **Ticket Assignment:** Assign tickets to appropriate agents or technicians based on their availability and expertise.
- **Ticket Tracking and Resolution:**
    - Track ticket status (open, in progress, resolved, closed).
    - Allow agents to add comments and updates to tickets.
    - Implement a system for resolving customer issues efficiently.
- **Customer Communication:**
    - Send automated email and or SMS notifications to customers regarding ticket status updates.

3.6 Testing and Quality Assurance

- **Unit Testing:** Conduct unit testing of individual components to ensure code quality and functionality.
- **Integration Testing:** Test the integration of different system components to identify and resolve issues.
- **User Acceptance Testing (UAT):** Conduct UAT with end-users to validate system functionality and user experience.
- **Performance Testing:** Perform performance testing to assess the system's scalability and responsiveness under load.
- **Security Testing:** Conduct security testing to identify and address vulnerabilities.

3.7 Deployment and Support

- **Deployment:** Deploy the system to a production environment, ensuring smooth transition and minimal downtime.
- **Post-Deployment Support:** Provide ongoing support, including bug fixes, performance optimization, and feature enhancements.
- **Training and Documentation:** Develop comprehensive documentation and provide training to system administrators and end-users.

3.8 Technical Requirements

- **Technology Stack:**
    - Preferred: Django framework, Python, PostgreSQL/MySQL, ReactJS/Angular (or similar frontend framework). – Or your chosen technology stack.
    - Cloud infrastructure (AWS, Azure, GCP) preferred. – Or your chosen technology stack
- **API Integration:**
    - Integration with mobile money provider APIs.
    - Integration with smart meter APIs.
    - Integration with QuickBooks (or similar accounting software) for financial data transfer.
- **Data Security:**
    - Data encryption, access controls, and regular security audits.
    - Compliance with relevant data privacy regulations.
- **Scalability and Performance:**
    - High availability, fault tolerance, and scalability to accommodate future growth.
- **System Architecture:**
    - Detailed description of the proposed system architecture, including components, data flow, and integrations.

3.9 Project Deliverables:

- **Fully functional and integrated digital platform.**

- **Comprehensive user documentation and training materials.**
- **Source codes were applicable and system documentation.**
- **Post-deployment support and maintenance plan.**

4. Evaluation Criteria

- **Technical Proficiency:** Experience with relevant technologies (e.g. Django, Python, APIs, cloud computing).
- **Project Experience:** Demonstrated experience in developing similar projects (e.g., CRM, ERP, utility management systems).
- **Proposed Solution:**
    - Quality and feasibility of the proposed solution, including system architecture and design.
    - Innovation and creativity in addressing EP's specific needs.
- **Cost:** Competitive pricing and cost-effectiveness of the proposed solution.
- **Timeline:** Proposed project timeline and ability to meet deadlines.
- **Team Expertise:** Qualifications and experience of the proposed project team.
- **Understanding of EP's Requirements:** Demonstrated understanding of EP's business needs and challenges.

5. Submission Requirements

- **Proposal:** A detailed proposal outlining the proposed solution, methodology, and team.
- **Technical Specifications:** Detailed technical specifications, including system architecture, technology stack, and integration details.
- **Pricing:** Detailed pricing breakdown, including costs for development, deployment, operating, training, and ongoing support.
- **References:** List of previous projects and client references like this project.

6. Submission Deadline:

- Friday 7th February, 2025 17:00:00 - East African Standard Time

7. Contact Information:

- info@equatorial-power.com
- petra.namugenyi@equatorial-power.com

8. References

- **See further illustrative and relevant information and the end of this document**
    - System Requirements Specification (SRS)
    - Use Case Diagrams
    - Data Flow Diagrams
    - Entity-Relationship (ER) Diagrams

9. Evaluation Process:

The evaluation of proposals will be based on the following criteria:

- **Technical Proficiency:**
    - Demonstrated expertise in the relevant technologies (Django, Python, APIs, cloud computing, database technologies) or your chosen technology stack to deliver this project.
    - Quality of the proposed technical solution, including system architecture, design, and integration.
    - Experience with integrating with mobile money providers and smart meter systems or similar APIs
    - Experience with developing and deploying scalable and secure web and mobile applications.
- **Project Experience:**

- Demonstrated experience in developing and implementing similar projects (e.g., CRM, ERP, utility management systems, digital platforms for energy sector).
- Successful track record of delivering complex IT projects on time and within budget.

- **Proposed Solution:**
  - Quality and feasibility of the proposed solution, including system architecture and design.
  - Innovation and creativity in addressing EP's specific needs and challenges.
  - Alignment of the proposed solution with EP's long-term strategic goals.
- **Cost:**
  - **Competitive Pricing:** Cost-effectiveness of the proposed solution.
  - **Total Cost of Ownership (TCO):**
    - Clear and detailed breakdown of all costs, including development, deployment, training, operating, ongoing support, and maintenance.
    - **Estimation of operating costs, including:**
      - **Hosting fees:** Costs associated with hosting the platform (cloud-based or on-premise).
      - **Integration costs:** Costs associated with integrating with third-party systems (e.g., mobile money providers, smart meter systems).
      - **Communication costs:** Estimated costs of SMS, mobile money transactions fees, USSD Session fees and other communication channels.
      - **Maintenance and support costs:** Ongoing costs for system maintenance, bug fixes, and technical support.
  - **Value for Money:** Overall value proposition considering the cost and benefits of the proposed solution.

- **Timeline:**
    - Proposed project timeline and ability to meet deadlines.
    - Realistic and achievable milestones.
- **Team Expertise:**
    - Qualifications and experience of the proposed project team.
    - Availability and commitment of key personnel.
- **Understanding of EP's Requirements:**
    - Demonstrated understanding of EP's business needs, challenges, and objectives.
    - Ability to address EP's specific requirements and priorities.
- **Communication and Collaboration:**
    - Effectiveness of communication and collaboration throughout the proposal and project lifecycle.

10. Questions and Clarifications:

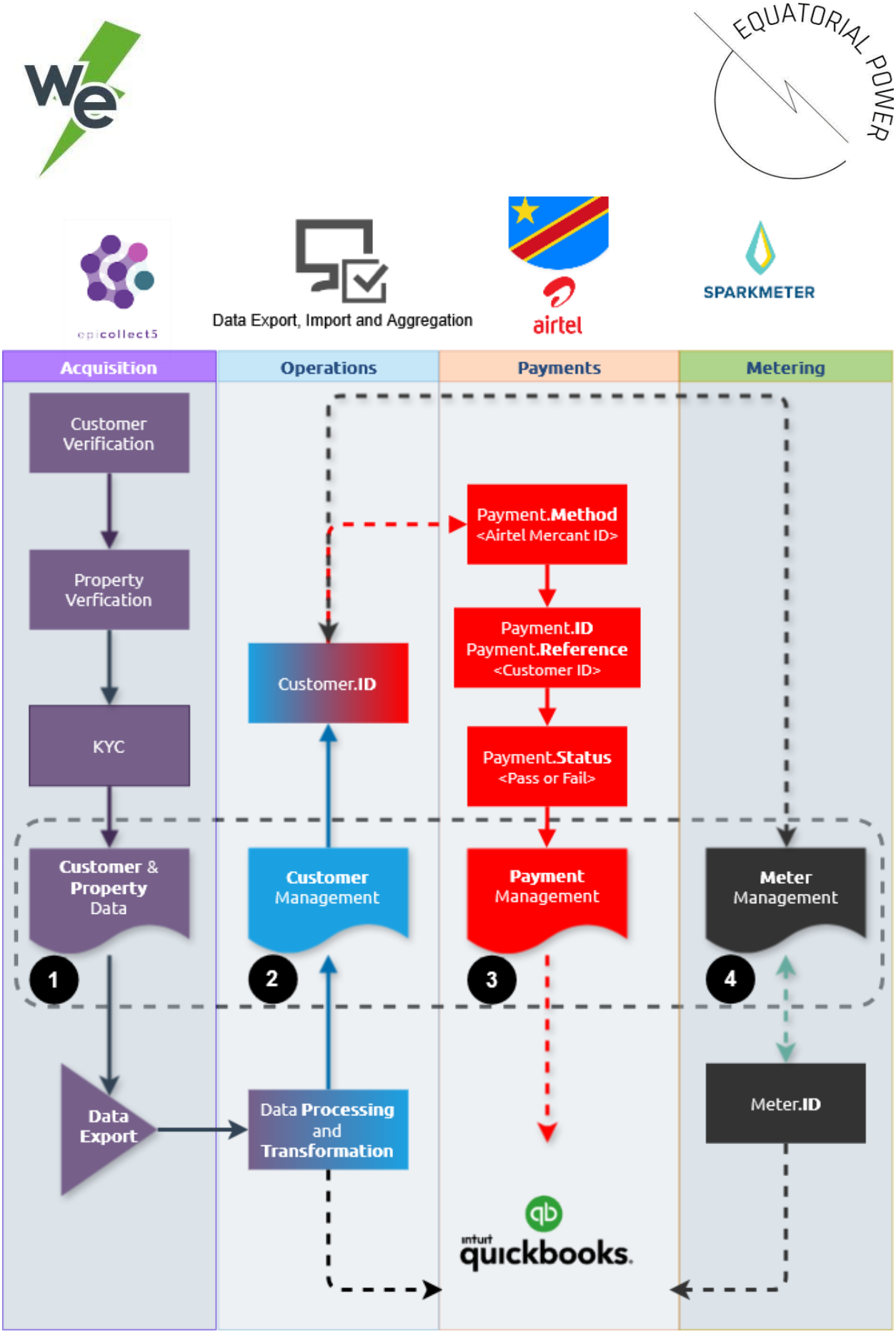
- All questions and requests for clarification regarding this RFP must be submitted in writing to info@equatorial-power.com; petra.namugenyi@equatorial-power.com**.**
- An info session will be held on the 23rd of January 2025 at 1:30 EAT a the following link https://meet.google.com/cme-ftqy-yox

Technical Annex

Current System Architecture for EP

1. Overview

EP currently manages its operations utilizing multiple systems, including Airtel DRC, the Spark Meter Management System, Epicollect5, and QuickBooks. While these systems play crucial roles in different aspects of the business (customer data management, meter management, customer support, financial accounting), they currently operate independently, lacking seamless integration. This fragmented approach presents operational inefficiencies and challenges in data synchronization and analysis. This diagram below outlines the primary components of the existing system.


EQUATORIAL POWER
epicollect5
Data Export, Import and Aggregation
airtel
SPARKMETER
Acquisition
Operations
Payments
Metering
Customer Verification
Property Verfication
KYC
Customer & Property Data
Data Export
Customer.ID
Customer Management
Data Processing and Transformation
Payment.Method
<Airtel Mercant ID>
Payment.ID
Payment.Reference
<Customer ID>
Payment.Status
<Pass or Fail>
Payment Management
Meter Management
Meter.ID
1
2
3
4
intuit quickbooks

2. Key Components

a. Epicollect5

- **Purpose:** Customer Acquisition
- **Functionality:** Facilitates the "Know Your Customer" (KYC) process by gathering essential information. Generates customer and property data.
- **Additional Features:** Offline data collection capabilities, customizable forms, GPS tracking, and barcode scanning.

b. Data Export, Import and Aggregation (ETL)

- **Purpose:** Customer Relationship Management, Accounting and Ticketing (CRM)
- **Functionality:**
  - i. **Customer Onboarding:**
    1. EP team exports customer data from Epicollect5.
    2. This data is imported into the Spark Meter Management System.
    3. Unique Customer numbers are assigned to each customer during this process.
  - ii. **Payment Processing:**
    1. Customers pay for electricity via Airtel Money, including their unique Customer numbers for easy identification.
    2. EP receives SMS notifications from Airtel Money regarding successful payments.
    3. EP subsequently updates the Spark Meter Management System with the payment information.
  - iii. **Financial Accounting:**
    1. EP accesses the Airtel DRC Money Portal to export payment data.
    2. This exported data is then imported into QuickBooks for financial accounting purposes.

c. **Airtel DRC**

- **Purpose:** Airtel Mobile Money Payment gateway.
- **Functionality:** Accepts payments made to EP's Merchant ID/Code and provides a portal to view transactions made to the Merchant ID. Further provides payment notification via SMS to a specified Mobile Number
- **Additional Features:** USSD payments, mobile money payments, SMS messaging, and voice calls.

d. **Sparkmeter**

- **Purpose:** Meter management.
- **Functionality:** Monitors meter status, allocates units based on payments, and provides real-time energy usage data.
- **Additional Features:** Remote meter reading, load management, and demand response capabilities.

3. Data Flow

a. **Epicollect5** collects customer, property, and meter data.
b. **Operations ETL** involves EP team extracting data from Epicollect and preparing it to be aggregated (ETL) and assigning unique customer numbers and imported upstream to Quickbooks and Spark Meter Management system.
c. **Airtel - DRC** processes customer payments and further provides EP with a mobile money portal to view, report and extra customer transactions
d. **Sparkmeter** provides meter management.
e. **Productive Hub Management** handles non-utility product and service sales.
f. **Power Plant Management System** monitors power plant operations and generates data.
g. **PowerBI** analyzes data from all applications, including the Power Plant Management System, to provide insights and reports. Currently, the Power Plant Management System is isolated from the other systems, limiting the ability to analyze and optimize the overall power generation and distribution process.

4. Key Definitions

a. **System Requirements:** System requirements are the minimum hardware and software specifications needed for a particular software, application, or game to run smoothly. These requirements ensure optimal performance and compatibility.
b. **Business Requirements:** Business requirements are the fundamental needs and expectations that a software system must fulfill to align with an organization's strategic goals. They define the "what" of the software, rather than the "how."
c. **Technical Requirements:** Technical requirements are the specific technical specifications that a software system must meet to fulfill its intended purpose. They provide the "how" behind the "what" of business requirements.
d. **Test case:** A test case is a set of conditions or variables used to determine whether a particular program path or software feature functions as expected.
e. **Requirements Traceability Matrix:** A requirements traceability matrix is a document that maps all the requirements with the test cases. By using this document one can verify test cases cover all functionality of the application as per the requirements of the customer.

5. Abbreviations

a. **BR:** Business Requirements
b. **RTM:** Requirements Traceability Matrix
c. **TR:** Technical Requirements
d. **TC:** Test case

6. Mapping of Company Positions to System Permissions

Team (Company Position)	System Permission

Operations	Customer Management
Technical	Meter Management
Finance	Customer Management Payments Management

7. Business Requirements Table

BR#	Module Name	Roles	Description
B1	Login and Logout	Customer Manager Meter Manager Payment Manager(operations team, finance team, tech team, digital team)	All roles can login using the login page. Post Login page will show different links based on role.
B2	Register clients	Customer Manager (operations team, tech team,digital team)	The Customer Manager can onboard new customers. (digital team does bulk addition of customer information into the crm) The Customer Manager can save the client's details and can assign them a customer ID.(operations team does addition of clients who subscribe later after acquisition period, tech team does for Spark records)

BR#	Module Name	Roles	Description
B3	Update client information	Customer Manager(operations Team)	The Customer Manager can change customer details. Eg. address, phone number, email,...(operations team does this and tech team does for spark updates)
B4	View list of clients	Customer Manager (operations team, tech team)	The Customer Manager can view the list of registered customers. (operations team The Customer Manager can view clients with active connections and clients who are not yet connected.(operations team, (tech team should also be able to view the customers not connected to enable their connection)
B5	Assign meter serial number	Meter Manager(tech team)	The Meter Manager can assign a meter serial number to a customer ID.(tech team assigns)
B6	Update meter	Meter Manager(operations team, tech team)	The Meter Manager can turn on a meter. The Meter Manager can turn off a meter(operations , tech team). The Meter Manager can archive a meter.(tech team) The Meter Manager can send units to a meter.(operations team)
B7	View list of meters	Meter Manager(operations	The Meter Manager can view the list of meters.

BR#	Module Name	Roles	Description
		team, tech team)	The Meter Manager can view active, inactive and archived meters.
B8	View payments	Payments Manager(operations team, finance team)	The Payments Manager can view all payments.(operations team, finance team) The Payments Manager can view successful payments.(operations, finance teams) The Payments Manager can view pending payments.(operations, finance teams)

8. Technical Requirements Table

TR#	BR#	Feature Name	Description
T1	B1	Authentication	A user can be authenticated using an email and password.
T2	B1	Verify email address	A user must verify their email address.
T3	B1	Reset Password	A user can reset their password in case they have forgotten it.
T4	B2	Create customer	Create a customer record in the database

T5	B3	Update customer	Update customer record in the database
T6	B4	Query list of customers	Retrieve list of customers from the database.
T7	B5	Assign meter serial number	Send a request to Spark API to add a meter serial number with its customer ID.
T8	B6	Update meter	Update meter details in the database.
T9	B6	Manage meter	Send instructions to meter through the SparkMeter API.
T10	B7	Query list of meters	Retrieve list of meters from the database.
T11	B8	Query list of payments	Retrieve list of payment transactions from the database.

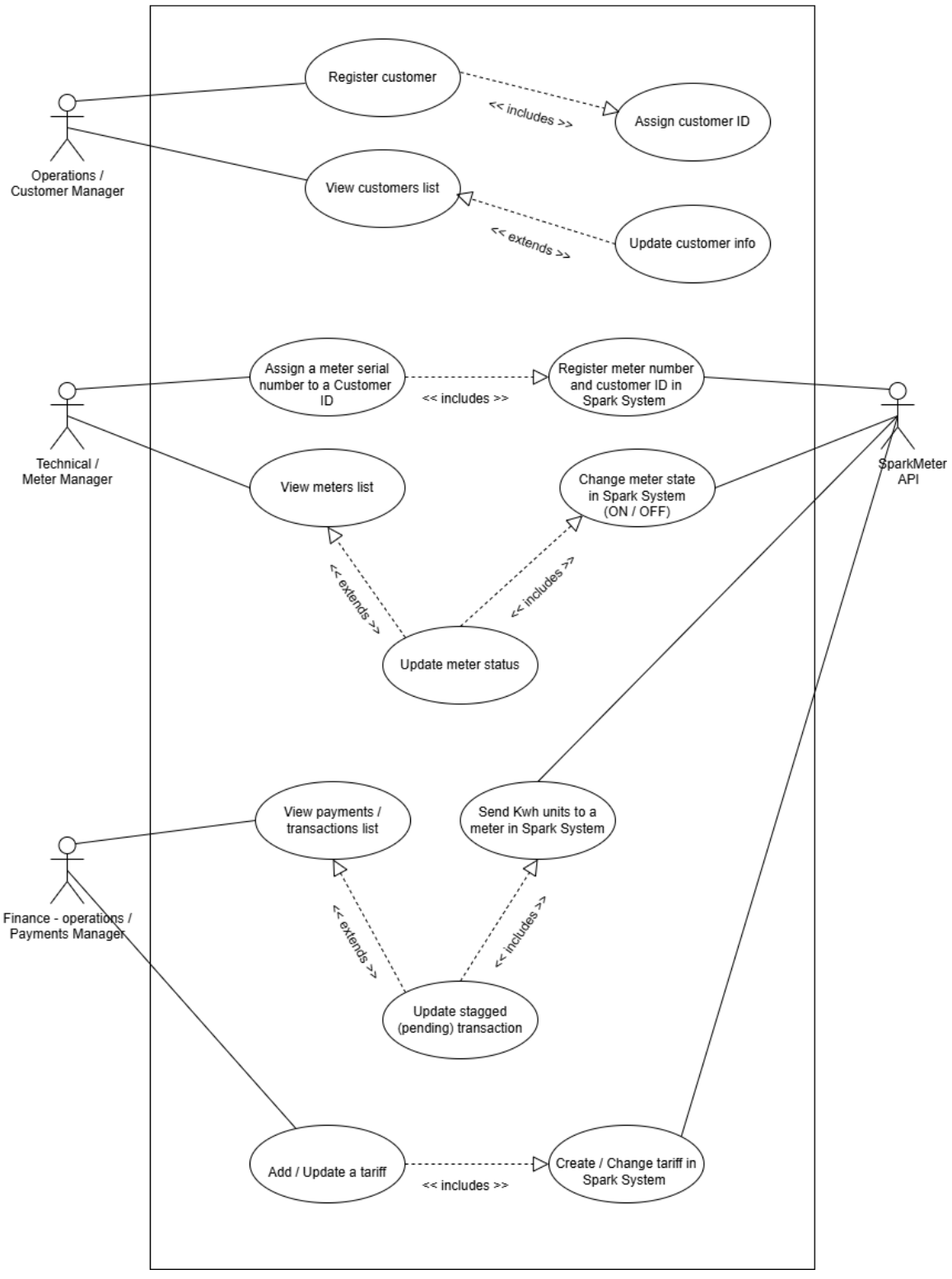
9. Test Cases

TC#	BR#	TR#	Test Case	Test Steps	Test Data	Expected Results
TC1	B1	T1	Check login	1. Go to login page 2. Enter email and password 3. Click login button	Email: user@equatorial-power.com Password: test123	Successful
TC2	B1	T2	Confirm email	1. Login	123456	Successful

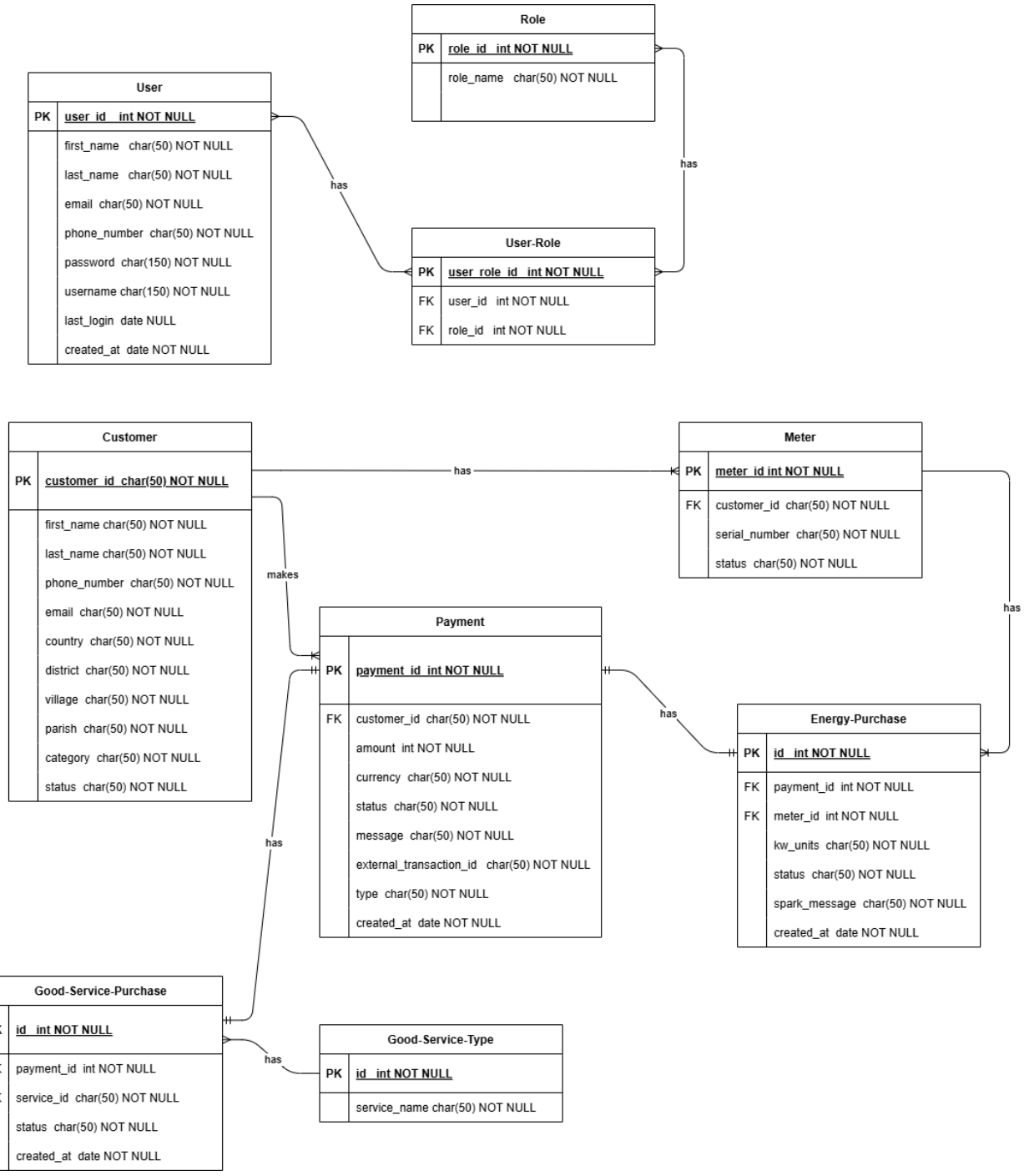
				2. Enter OTP sent to email		
TC3	B1	T3	Change Password	1. Go to login page 2. Click on forgot password 3. Enter email 4. Verify email 5. Enter new password	Email: user@equatorial-power.com New Password: testing123	Successful
TC4	B5	T7	Assign meter serial number	1. Login 2. Go to customer management 3. Go to pending connections 4. Select customer ID 5. Enter a meter serial number 6. Click on connect button	Customer ID: LW2110 Serial Meter Number: SMRS-DRF-3242	Successful

10. Use Case Diagram

EP - Energy Payment Management System



11.Proposed Entity Relationship Diagram



12. RFP Additional Information

Proposed timeline: These are the project timelines, including key milestones

Phase 1: System Design and Development

- **Task 1: System Architecture Design:** 05 Days
- **Task 2: User Interface Development:** 20 Days
- **Task 3: API Integration:** 20 Days
- **Task 4: Data Flow Diagram Creation:** 01 Day
- **Task 5: Security Implementation:** 05 Days

Phase 2: Mobile Money Integration

- **Task 1: Mobile Money Provider Selection:** 10 Days
- **Task 2: API Integration:** 05 Days
- **Task 3: Payment Reconciliation System Development:** 05 Days

Phase 3: Smart Meter Integration

- **Task 1: API Integration:** 02 Days
- **Task 2: Data Synchronization Implementation:** 01 Day
- **Task 3: Error Handling and Retry Mechanisms:** 02 Days

Phase 4: Data Management and Reporting

- **Task 1: Database Design and Implementation:** 02 Days
- **Task 2: Data Security Implementation:** 02 Days
- **Task 3: Report Development:** 05 Days

Phase 5: Testing and Quality Assurance

- **Task 1: Unit Testing:** 05 Days
- **Task 2: Integration Testing:** 03 Days
- **Task 3: User Acceptance Testing (UAT):** 02 Days
- **Task 4: Performance Testing:** 01 Day
- **Task 5: Security Testing:** 01 Day

Phase 6: Deployment and Support

- **Task 1: System Deployment:** 02 Days
- **Task 2: Post-Deployment Support:** 10 days
- **Task 3: Training and Documentation:** 05 Days

13. Expected number of transactions :

	Customers	Transactions per month
Current	2,500	5,000
Projected for 2025	3,000	6,000

14. Technical Requirements

System Requirements:

These are the technical requirements for the Django application or its equivalent, including:

a. Core Functionalities

i. API Aggregation:

1. Consume APIs from payment service providers and smart meter systems.
2. Validate and process incoming data.
3. Transform and normalize data for internal use.

ii. User Management:

1. Implement user authentication and authorization mechanisms (e.g., username/password, OAuth).
2. Manage user roles and permissions.
3. Track user activity and audit logs.

iii. API Creation:

1. Develop RESTful APIs to expose functionalities to third-party applications.
2. Implement API rate limiting and security measures.

iv. Data Storage:

1. Design and implement a robust database schema to store user data, payment information, meter readings, and system logs.
2. Ensure data integrity and security through appropriate database

normalization and security measures.

b. Technical Specifications
   i. **Programming Language:** Python
   ii. **Framework:** Django
   iii. **Database:** PostgreSQL or MySQL
   iv. **Frontend:** ReactJS or similar
   v. **Cloud Platform:** AWS, GCP, or Azure
   vi. **API Gateway:** API Gateway service (e.g., AWS API Gateway, Kong)
   vii. **Security:**
      1. Implement strong password hashing and salting.
      2. Use HTTPS for secure communication.
      3. Implement input validation and sanitization to prevent security vulnerabilities.
      4. Regularly update and patch system components.
   viii. **Scalability:**
      1. Design the application to handle increasing load and future growth.
      2. Consider using load balancing and caching techniques.
   ix. **Performance:**
      1. Optimize database queries and API responses.
      2. Implement caching mechanisms to reduce database load.
   x. **Monitoring and Logging:**
      1. Set up monitoring tools to track system performance and identify potential issues.
      2. Implement logging to record system events and errors.
      3.

c. Specific Technical Requirements
   i. **Payment API Integration:**
      1. Authentication and authorization with the payment provider's API.
      2. Securely handling sensitive payment information.
      3. Processing payment notifications and updating customer electricity accounts.
   ii. **Smart Meter API Integration:**
      1. Authentication and authorization with the smart meter system's API.

2. Retrieving meter readings and updating customer usage.
3. Triggering remote meter operations (e.g., disconnection, reconnection).

iii. User Interface:

1. User-friendly dashboard for system administrators.
2. Intuitive interface for customer service representatives.
3. Secure user authentication and authorization.
4. Role-based access control.

iv. API Documentation:

1. Clear and concise API documentation, including API specifications, usage examples, and error codes. Example of solution integrating with Quickbooks:

2. API Integration with QuickBooks

a. To ensure seamless integration with QuickBooks, the system will include a robust API that will:

i. **Extract Relevant Data:** Capture necessary customer and payment information, such as customer ID, payment amount, payment method, and timestamp.

ii. **Format Data for QuickBooks:** Transform the data into a format compatible with QuickBooks's API.

iii. **Securely Transmit Data:** Use secure protocols to transmit data to QuickBooks.

iv. **Handle API Responses:** Process responses from QuickBooks to confirm successful data transfer or identify errors.

v. **Implement Error Handling:** Implement robust error handling and retry mechanisms to ensure reliable data transfer.

15.API Integration for Payments, Spark Meter, Quick Books

a. API Keys and Authentication
b. Data Formats (e.g., JSON, XML)
c. Communication Protocols (e.g., REST, SOAP)
d. Security Measures (e.g., encryption, authentication)