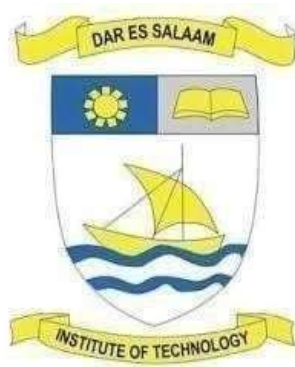


DAR ES SALAAM INSTITUTE OF TECHNOLOGY



DEPARTMENT OF ENGINEERING IN ELECTRONICS AND TELECOMMUNICATION
ENGINEERING
BACHELOR DEGREE OF ENGINEERING
SENIOR PROJECT II
NTA LEVEL 8

MODULE CODE: ETU 08125

MODULE NAME: PROJECT CONCEPTUALIZATION

PROJECT TITLE: ELECTRICITY SUB METERING SYSTEM FOR TENANTS
ENERGY CONSUMPTION.

PROJECT TYPE: PROBLEM SOLVING

SUPERVISORS NAME: Dr. Jordan H. Hosea

ACADEMIC YEAR 2024/2025

S/N	STUDENT NAME	REGISTRATION NUMBER
1.	SHABANI ATHUMANI	2106307228976
2.	RAJABU S MUSHI	220242452563

DECLARATION

SHABANI ATHUMANI and RAJABU MUSHI, we are declaring that to the best of our knowledge, this project presented here as partial fulfilment for the award of Bachelor of Telecommunication engineering (NTA LEVEL 8) at Dar es Salaam Institute of Technology has been done by us and is our own work, except were indicated otherwise.

NAME	SIGNATURE	DATE
1. SHABANI ATHUMANI
2. RAJABU S. MUSHI

CERTIFICATION

The undersigned certifies that has read, and hereby recommend for acceptance by Dar es Salaam Institutes of technology a project titled “ELECTRICITY SUB METERING SYSTEM FOR TENANTS ENERGY CONSUMPTION” in fulfilment of the requirements for the Bachelor degree in telecommunication Engineering. SUPERVISOR: **Dr. Jordan H. Hosea**

SIGNATURE:

DATE:

ABSTRACT

This project presents the design and implementation of an electricity sub-metering system for monitoring individual tenant energy consumption in residential and commercial properties. The system utilizes sub meters and a microcontroller to capture real-time energy usage data for multiple tenants. The display and keypad provide a user interface for checking balances and entering prepaid tokens, while a relay module controls power supply disconnection upon credit depletion.

Data on energy consumption and billing is stored in either a local or cloud-based database, enabling structured record-keeping and integration with tariff structures such as those set by TANESCO. The system supports automated monitoring, tenant-specific usage tracking, and scalable communication functions, allowing deployment across various building types. The overall configuration facilitates accurate energy usage recording, credit management, and data accessibility without manual intervention.

ACKNOWLEDGEMENT

First, we would like to thank the Almighty God for His grace and would like to thank Dar es salaam Institute of Technology for providing this opportunity to do this project as part of our curriculum and we like to thank our project coordinator **Dr. Jordan H. Hosea** for his valuable guidance and motivation. We express our sincerely gratitude to our project supervisor, **Dr. Jordan H. Hosea** for his excellent guidance, comments and views that have helped us towards the accomplishment of this phase. At last, we deeply thank our guidance have inspired and supported us throughout the project period and our friends for their constant encouragement and support.

LIST OF FIGURES

Figure 1: Block Diagram of Existing System	4
Figure 2: Block Diagram of Proposed System	11
Figure 3: Flow Chart of the Designed Proposed System	13
Figure 4: Flow Chart of the Database.....	14
Figure 5: Bar Graph Showing Checklist during Observation.....	19
Figure 6: Bar Graph Showing Data Collected Through Questionnaires.....	20
Figure 7: Sub meter.....	27
Figure 8: LCD Display.....	25
Figure 9: ESP 8266.....	32
Figure 10: Relay switch.....	33
Figure 11: PCB design systematic.....	36
Figure 12: PCB layout	37
Figure 13: PCB 3D.....	37
Figure 14: Prototype.....	41
Figure 15: Prototype connection.....	42
Figure 16: first sub meter.....	42
Figure 17: Admin interface.....	45
Figure 18: Database structure.....	45

LIST OF TABLES

Table

4.2.1: Shown Data Collected Through Observation.....	16
4.2.2: Data Collected Through Questionnaires.....	17
4.2.3: Checklist Used During Observations.....	24
5.2.3: Specifications of different modules.....	28
5.3.3: Microcontroller modules	30
5.4: Relay module.....	32
7.4: Prototype results.....	47
5.2: Prototype and existing sub meter reading.....	47

Contents

DECLARATION	i
CERTIFICATION	ii
LIST OF FIGURES	v
LIST OF TABLES	vi
CHAPTER ONE	1
1.0 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 PROBLEM STATEMENT	1
1.3 CAUSES OF THE PROBLEM	1
1.4 PROJECT OBJECTIVE	1
1.4.1 MAIN OBJECTIVE.....	1
1.4.2 SPECIFIC OBJECTIVES	2
1.5 SIGNIFICANCE OF THE PROJECT	2
CHAPTER TWO	4
LITERATURE REVIEW	4
2.0 INTRODUCTION	4
2.1 PROJECT OVERVIEW	4
2.2 EXPLANATION OF THE EXISTING SYSTEM	5
2.2.1 POWER SUPPLY (TANESCO LINE)	5
2.2.2 MAIN METER (CONVENTIONAL METER).....	5
2.2.3 SUB-METERS	5
2.2.4 MANUAL READING AND BILLING	5
2.2.5 MANUAL DATA COLLECTION	5
2.2.6 BILLING SYSTEM	6
2.3 LIMITATIONS OF THE TRADITIONAL SYSTEM (AS A WHOLE)	6
2.4 SUMMARY OF LITERATURE REVIEW OF EXISTING SYSTEM;	6
CHAPTER THREE	9
METHODOLOGY	9
3.0 INTRODUCTION	9

3.1 EXPLANATION OF THE PROPOSED BLOCK DIAGRAM	11
3.1.1 TANESCO POWER LINE	11
3.1.2 MAIN METER	11
3.1.3 CONTROL UNIT	11
3.1.4 DISPLAY UNIT	12
3.1.5 SUB-METERS	12
3.1.7 RELAY UNIT	12
3.1.8 COMMUNICATION UNIT	12
3.1.9 EXPLANATION OF THE FLOW CHART OF THE DESIGNED PROPOSED SYSTEM:	13
CHAPTER FOUR	15
DATA COLLECTION	15
4.0 INTRODUCTION	15
4.1 PRIMARY DATA COLLECTION	15
4.2 DATA COLLECTED THROUGH OBSERVATION	15
4.3 EXPLANATION OF THE HISTOGRAMS BASED ON RESPONDENTS' Data	18
4.4 SECONDARY DATA COLLECTION	21
4.4.1 ENERGY SUB-METERS	21
4.4.4 RELAY MODULE	22
4.4.5 CENTRALIZED DATABASE	23
4.4 CHAPTER SUMMARY	24
CHAPTER FIVE	25
DATA ANALYSIS	25
5.0 INTRODUCTION	25
5.1 ENERGY SUB-METER UNIT	25
5.1.1 SUB-METER COMPARISON	25
5.2 SUB-METER COMPARISON	26
5.3 DISPLAY UNIT	27
5.3.1 COLLECTED DATA	28
5.3.2 REQUIREMENTS OF THE DISPLAY UNIT	28
5.3.3 DISPLAY MODULE COMPARISON	28
5.4 CONTROL UNIT	29

5.4.1 COLLECTED DATA	30
5.4.2 REQUIREMENTS OF THE MICROCONTROLLER	30
5.3.4 ESP32 MICRO-CONTROLLER DESCRIPTION	31
5.5 RELAY MODULE	32
5.5.1 COLLECTED DATA	32
5.5.2 REQUIREMENTS OF THE RELAY MODULE	32
5.5.3 RELAY DESCRIPTION	33
CHAPTER SIX	35
DESIGN AND SIMULATION	35
6.0 INTRODUCTION	35
6.1 SIMULATION OBJECTIVES	35
6.2 CIRCUIT COMPONENTS USED IN SIMULATION	35
6.3 SIMULATION SETUP	37
6.3 TEST SCENARIO AND RESULTS	38
CHAPTER SEVEN	40
PROTOTYPE IMPLEMENTATION AND TESTING	40
7.1 INTRODUCTION	40
7. 2 HARDWARE PART	40
7. 2.1 HARDWARE SPECIFICATION	40
7.2.2 CIRCUIT SUMMARY	41
7.2.3 OPERATION FLOW.....	41
7.2.4 SYSTEM WIRING	42
7.2.5 SYSTEM BEHAVIOR	44
7.2.5 PROTYPE SUMMARY:	45
7.3 SOFTWARE PART	45
7.3.1. INTRODUCTION	45
7.3.2. DATABASE DESIGN.....	45
7.3.3 SOFTWARE FUNCTIONALITY OVERVIEW	47
7.3.4 SUMMARY OF SOFTWARE BENEFITS.....	47
7.4 PROTOTYPE TESTING AND RESULTS	48
7.5 COMPARISON: PROTOTYPE AND EXISTING SUB-METER READINGS	48

CHAPTER EIGHT	50
CONCLUSION AND RECOMMENDATIONS	50
REFERENCE	51
APPENDICES	53
APPENDEX B	54
APENDEX C.	56
SOURCE CODES	57

LIST OF ABBREVIATIONS

SMS	Short Message Service
UART	Universal Asynchronous Receiver Transmitter
RTOS	Real-Time Operating System
HEMSs	Home Energy Management Systems
IoT	Internet of Think
TANESCO	Tanzania electric supply company limited
IEEE	Institute of electrical and electronics engineers
DC	Direct current
LCD	Liquid Crystal Display

CHAPTER ONE

1.0 INTRODUCTION

1.1 BACKGROUND

The rising global demand for energy, coupled with the need for efficient resource management, has emphasized the importance of electricity sub metering system for tenants' energy consumption. Addresses the challenges of accurately tracking and managing energy consumption in shared or rental properties. Traditional electricity meters are often inefficient in breaking down energy usage among multiple tenants, leading to disputes and inequities. This project aims to develop a modern, automated metering solution that not only provides real-time energy consumption data for individual tenants but also integrates advanced monitoring and reporting features. The system ensures transparency, improves energy efficiency, and promotes accountability among tenants, making it a sustainable and practical solution for residential and commercial properties.

1.2 PROBLEM STATEMENT

In shared rental spaces, manual electricity sub metering causes inefficiencies like time wastage, power outages without alerts, and unfair billing due to the inability to track individual usage. This leads to tenant disputes and dissatisfaction. An Electricity sub metering system for tenants' energy consumption provides real-time monitoring, fair cost allocation, and transparent electricity management, ensuring convenience and equity for tenants and property owners.

1.3 CAUSES OF THE PROBLEM

Conventional electricity meters are designed to measure the total energy consumption of an entire property without distinguishing usage by individual tenants, making it difficult to allocate costs equitably.

1.4 PROJECT OBJECTIVE

The objectives of the project consist of the main objective and specific objectives.

1.4.1 MAIN OBJECTIVE

The primary objective of the Electricity sub metering system for tenants' energy consumption is to design and implement a energy management solution that enables real-time monitoring, accurate billing, and fair distribution of electricity costs among tenants in multi-occupancy buildings.

1.4.2 SPECIFIC OBJECTIVES

- i. Develop a system capable of monitoring and displaying individual tenant energy usage in realtime system.
- ii. Implement a billing algorithm that accurately calculates each tenant's share of the total electricity cost based on his or her individual consumption, ensuring precise and equitable billing.
- iii. Create an intuitive user interface (web or mobile) to allow tenants and property owners to access detailed energy usage reports, with historical data and trends updated at least hourly.
- iv. Integrate a notification system to alert tenants and property owners of power outages or abnormal consumption patterns within 30 seconds of detection.
- v. Design the system to support a minimum of 50 tenants per property and ensure compatibility with existing electrical infrastructures and smart home devices.
- vi. To conduct Field Testing and Refinement from the implemented system.

1.5 SIGNIFICANCE OF THE PROJECT

- I. Ensures tenants are billed accurately based on their actual energy consumption, eliminating disputes over shared electricity costs.
- II. Provides tenants with real-time data on their electricity usage, encouraging energy-saving habits and reducing unnecessary expenses.
- III. Builds trust by allowing tenants to monitor their energy usage and verify billing calculations.
- IV. Provides valuable consumption data that can aid in planning energy policies and infrastructure improvements
- V. Improved Energy Management: Real-time energy monitoring allows property owners and tenants to track usage patterns accurately. This helps optimize resource allocation, identify peak consumption periods, and reduce energy wastage, leading to improved overall management efficiency.
- VI. Fair and Transparent Billing: By using smart sub-meters, the project ensures that each tenant pays only for his or her actual energy consumption. This eliminates disputes over shared utility bills, creates trust, and promotes financial transparency in shared living spaces.

- VII. Enhanced Tenant Satisfaction: Tenants gain control over their energy consumption through real time insights, helping them manage usage and expenses. Transparent billing and the ability to track anomalies foster a more harmonious living environment and improve tenant retention.
- VIII. Streamlined Cost Allocation: Automated energy cost distribution reduces the manual effort required for bill calculations and eliminates errors. Property owners benefit from accurate, efficient systems that free up time and reduce operational overhead.
- IX. Prevention of Interruptions: The system detects anomalies, such as overloading or unusual consumption, and sends alerts to property owners and tenants. This proactive approach prevents unexpected power outages and minimizes the risk of equipment failure.
- X. Energy Efficiency: The system encourages tenants to adopt energy-efficient habits by providing actionable insights into consumption. Over time, this reduces electricity bills and supports broader goals of reducing energy waste and promoting environmental conservation.

CHAPTER TWO

LITERATURE REVIEW

2.0 INTRODUCTION

This chapter reviews existing literature on electricity sub metering system for tenants' energy consumption. It highlights the progress made in these areas, identifies gaps in current systems, and provides a foundation for the proposed project. Summarizes of literature review for an " Electricity sub metering system for tenants' energy consumption" which could be relevant to your project or research.

2.1 PROJECT OVERVIEW

The electricity sub metering system for tenants' energy consumption is designed to address the challenges identified in the literature. The system will incorporate real-time data tracking, automated billing, and advanced analytics to improve energy management in shared spaces.

BLOCK

DIAGRAM OF EXISTING SYSTEM

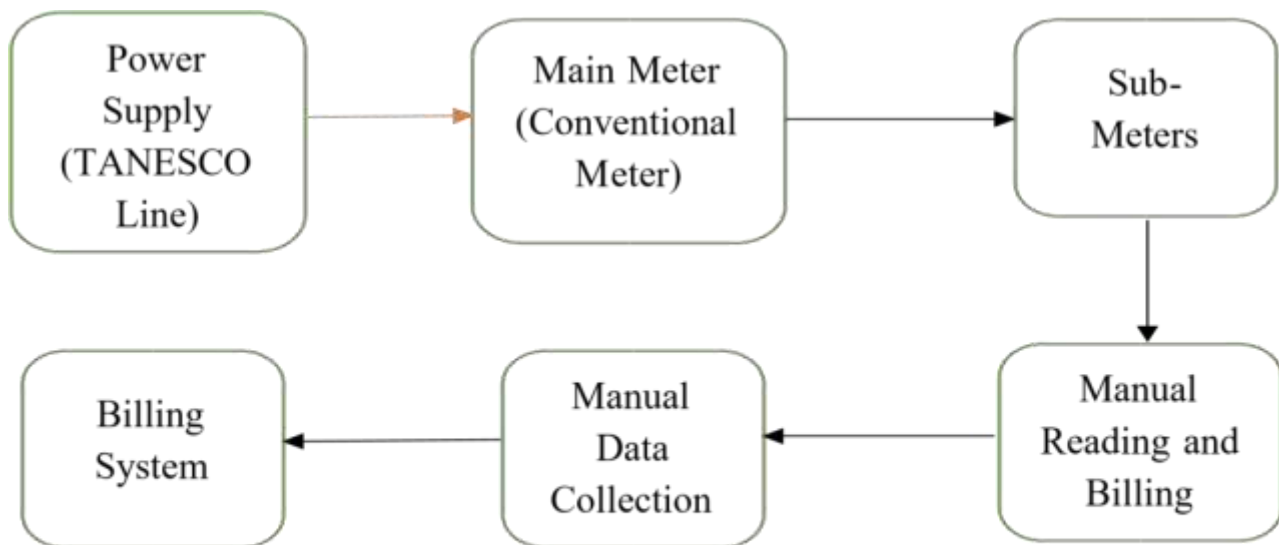


Figure 1: block diagram of existing system

2.2 EXPLANATION OF THE EXISTING SYSTEM

2.2.1 POWER SUPPLY (TANESCO LINE)

This block represents the electricity provided by TANESCO (Tanzania Electric Supply Company), which is the main grid supplier for electricity in Tanzania. TANESCO is responsible for delivering electrical power to the property. The supply is measured by the Main Meter before distribution to individual submeters as you can refer to the figure 1 above.

2.2.2 MAIN METER (CONVENTIONAL METER)

The **Main Meter** measures the total electricity consumption for the entire building or property as provided by TANESCO, it is a **conventional meter**, typically analog or basic digital. It records only the aggregate consumption for the whole property without distinguishing between individual tenants' usage. It provides a single bill for the entire property, which the landlord or property owner must manually divide among tenants.

2.2.3 SUB-METERS

Sub-meters are individual meters installed for each tenant or unit within the property to track their electricity consumption. Sub-meters enable monitoring of electricity usage at the tenant level.

MANUAL PROCESS

Sub-meters are read manually, with property managers or tenants responsible for recording the values periodically. Errors in manual readings can result in incorrect billing. Readings must be manually aggregated, which is time-consuming. Sub-meters are not directly connected to the Billing System, so there is no automation or real-time data tracking.

2.2.4 MANUAL READING AND BILLING

This block represents the process of manually reading the sub-meters and generating electricity bills for individual tenants. A person (e.g., the property manager) physically checks the sub-meters and records the electricity consumption for each tenant. The data is then used to calculate the cost of electricity based on the tariff rate as referred to the figure 1.

2.2.5 MANUAL DATA COLLECTION

This block involves the aggregation of data from the **Main Meter** and **Sub-Meters**. After collecting readings from individual sub-meters, the property manager aggregates the data: The sum of all sub-meter

readings should ideally match the total consumption recorded by the main meter, any discrepancies need to be addressed, which adds complexity to the process.

2.2.6 BILLING SYSTEM

The billing system is used to calculate electricity bills for tenants based on their individual energy consumption

2.3 LIMITATIONS OF THE TRADITIONAL SYSTEM (AS A WHOLE)

i. Manual Process:

Almost all aspects, from reading sub-meters to aggregating data and generating bills, are done manually, which is labor-intensive and prone to human error.

ii. Lack of Real-Time Monitoring:

Neither the main meter nor the sub-meters provide real-time data, making it impossible to track energy consumption instantly or detect issues like power outages promptly.

iii. Data Discrepancies:

Differences between the main meter and sub-meter readings are difficult to reconcile, causing confusion and disputes.

iv. Limited Transparency:

Tenants have no access to detailed reports or usage trends, leading to mistrust in the billing process.

v. No Alerts or Notifications:

The system does not alert tenants or property managers about anomalies, power outages, or abnormal usage patterns, which could lead to delayed responses to issues.

2.4 SUMMARY OF LITERATURE REVIEW OF EXISTING SYSTEM;

Raggi, L. M., Cunha, V. C., Trindade, F. C., & Freitas, W. (2022). Smart Metering in Distribution Systems: Evolution and Applications. In *Planning and Operation of Active Distribution Networks: Technical, Social and Environmental Aspects* (pp. 287-317). Cham: Springer International Publishing.

Raza, S., Khalid, A., Latif, K., & Muhammad, N. (2024). A review of home energy management systems (HEMSs): Technologies, trends, and energy optimization. *Renewable and Sustainable Energy Reviews*, *174*, Article 113334

Wang, W. (2024). Building Occupancy and Smart Metering. In *Circular Economy for Buildings and Infrastructure: Principles, Practices and Future Directions* (pp. 233-257). Cham: Springer International Publishing.

Morello et al. (2017) discuss the use of smart power meters and IoT for monitoring energy flow in smart grids, focusing on IoT, advanced sensing, and algorithm development, but neglecting integration challenges, economic feasibility, and cybersecurity.

Abate, F., Conti, M., & Spirito, M. A. (2019). A low-cost, compact smart power meter for IoT-based real-time monitoring and grid adaptability. *IEEE Internet of Things Journal*, 6(4), 6310–6320.

Caropreso et al. (2018) present an open-source, flexible framework for smart meters based on TCP/IP, using wireless technologies for data exchange. The framework focuses on data security and provides detailed traffic analysis but has limited real-world testing and may be complex for inexperienced users.

Carratu, Ferro, Pietro Santo, and Paciello (2018) describe an IoT-enabled smart meter that integrates electrical power, gas, and water metering. It offers real-time monitoring but has limited deployment data and lacks long-term performance assessment.

Darby, S. J. (2020). Demand response and smart technology in theory and practice: Customer experiences and system actors. *Energy Policy*, 143, 111573.

Hafeez, M., Khan, M. A., Ahmad, S., & Qamar, A. (2019). A flexible modular framework for load scheduling using HGWmEDE algorithm and forecaster module in price-based demand response. *IEEE Access*, 7, 94532–94545.

Jain S, Vinoth KN, Paventhan A, Kumar Chinnaiyan V, Arnachalam V, and Pradish (2014) provide a broad overview of smart grid technologies, including smart metering, IoT, and EMS. However, the work lacks in-depth technical analysis and real-world case studies.

Jordehi (2019) focuses on integrating renewable energy and improving energy efficiency through demand response. However, it has limited focus on technological challenges and does not address realworld scalability.

Kabalci (2016) surveys smart metering and smart grid communication, covering key technologies and protocols, but lacks in-depth coverage of security, economic, and regulatory challenges.

Marwa, R., & Mwanga, E. (2018). Tanzania: Energy policy. In *Encyclopedia of Mineral and Energy Policy* (pp. 1-6). Springer, Berlin, Heidelberg.

Vineeth et al. (2021) address power theft and data security in smart grids, focusing on theoretical approaches with limited exploration of cost feasibility and practical applications.

Salazar, E. (2009, November 20). Building energy metering: Costs and benefits of submetering (CIBSE TM39). CIBSE Publishing.

Efkarpidis, N., Geidl, M., Wache, H., Peter, M., & Adam, M. (2022). Smart Metering Applications. In Smart Metering Applications: Main Concepts and Business Models (pp. 13-124). Cham: Springer International Publishing.

DRAWBACKS

In this literature reviews we found the gaps in integration of relay control, token systems, and microcontroller hardware

Many studies like (Abate et al., 2019; Hafeez et al., 2019) propose smart meters but do not address hardware integration such as:

- a) Use of microcontrollers
- b) Relay-based load control
- c) Token/code entry systems for energy access

This makes us come with our prototype implementation in order to resolve these literature reviews by using token/code entry systems for energy access, Relay-based load control and the use of microcontroller.

CHAPTER THREE

METHODOLOGY

3.0 INTRODUCTION

The methodology outlines the systematic approach employed to design and develop the **electricity sub metering system for tenants' energy consumption**. This step-by-step framework ensures that the project is implemented efficiently, addressing both technical and practical challenges. The methodology integrates theoretical research, data analysis, design, simulation, and prototype testing, aligning all efforts towards achieving the project objectives.

The key components of the methodology include:

i Literature Review

This stage involves examining existing research and technical papers on smart metering systems, tenant energy monitoring, and billing automation. By studying various methodologies and technologies, the review identifies gaps in existing systems and informs the design of an innovative solution tailored for multi-tenant setups. The focus is on key technologies like IoT, and energy data analytics

ii Data Collection

Data is gathered from real-world environments to ensure the proposed system meets practical requirements. This includes collecting energy usage patterns, identifying tenant-specific challenges, and visiting potential implementation sites to understand their infrastructure and energy distribution systems.

iii Data Analysis

The collected data is processed and analyzed to identify trends, inefficiencies, and potential improvements in energy usage and billing processes. This analysis forms the basis for designing algorithms and models that optimize energy distribution and automate billing for tenants.

iv Designing the proposed System

This step involves creating detailed plans for the system's hardware and software components. It includes designing smart meters, selecting communication protocols, and developing software for realtime monitoring, tenant-specific data processing, and automated billing

v. Simulation of the Proposed System

Using mathematical models and simulation tools, the system's behavior is tested to ensure its components function as expected. This step focuses on testing energy measurement accuracy, communication protocols, and billing algorithms under simulated conditions.

vi Circuit Simulation

The electrical circuits powering the metering system are simulated to evaluate their performance. This ensures that the circuits can handle the required loads, are energy-efficient, and operate without errors.

vii Building the Prototype

The system design is translated into a physical model by assembling hardware components such as smart meters, sensors, and communication modules. The prototype is equipped with software to perform real-time energy monitoring and automated billing.

viii Testing the Prototype The prototype undergoes rigorous testing in a real-world environment to evaluate its functionality and reliability. Energy monitoring accuracy, communication efficiency, and billing automation are tested. Any issues identified during testing are resolved to optimize the system.

BLOCK DIAGRAM OF PROPOSED SYSTEM

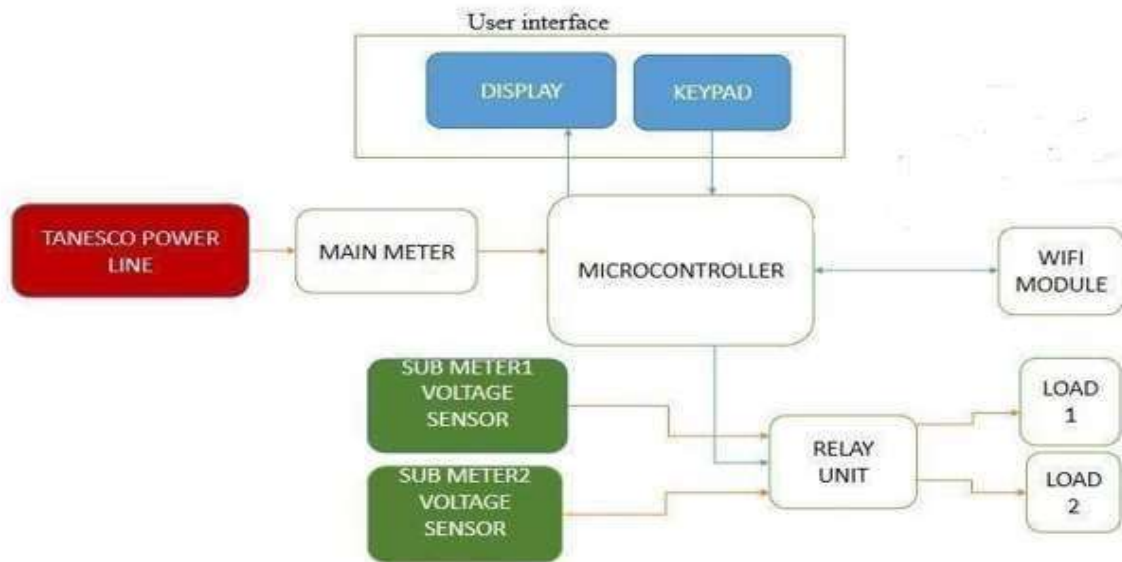


Figure 2: Block diagram of proposed system

3.1 EXPLANATION OF THE PROPOSED BLOCK DIAGRAM

3.1.1 TANESCO POWER LINE

This is the primary power source provided by the national electricity supplier (TANESCO). It serves as the input to the system, Functionality: Provides electricity to the entire system, including the main meter, submitters, and connected loads.

3.1.2 MAIN METER

The main energy meter tracks the total electricity usage for the entire system.

Functionality: Records energy consumption from the Tanesco power line and provides data for billing or monitoring. Acts as the reference point for total energy usage before the distribution to sub-meters.

3.1.3 CONTROL UNIT

Processes data from sensors, including voltage and current measurements. Controls the display of information on the Display. Handles user inputs, Sends data to the micro controller for communication.

Coordinates all system activities and ensures accurate monitoring and control of energy usage.

3.1.4 DISPLAY UNIT

Displays information like current consumption, user-specific energy usage, and system status Acts as the primary feedback interface for users to view energy data and system alerts.

3.1.5 SUB-METERS

Sub-Meter 1: Assigned to User 1 to monitor their individual energy consumption.

Sub-Meter 2: Assigned to User 2 to monitor their energy usage.

Each sub-meter is equipped with a current sensor and voltage sensor to calculate energy usage accurately.

3.1.7 RELAY UNIT

Controlled by the microcontroller based on data from the communication units.

Functionality: Automatically disconnects electricity supply when a user has no units or has not paid the bill. Reconnects power once the user tops up their balance. Role: Executes the system's automatic cutoff and reconnection commands, ensuring effective energy management.

3.1.8 COMMUNICATION UNIT

i. Communication module

A communication module is used for wireless communication via cellular networks. Functionality: Sending Energy Consumption to Users: The module sends notifications to users, informing them of their energy usage, remaining balance, and alerts for low units.

Alerts for Disconnection: Notifies users when their electricity supply is cut off due to insufficient units.

User Interaction: Allows users to send requests (e.g., balance checks or top-ups) and receive real-time responses. Role: Acts as the primary interface between the smart meter system and the user's mobile phone.

ii. Database (Cloud or Local Server)

A centralized system where user energy consumption, billing history, and payment status are stored.

Functionality: Storage Logs energy usage, payment transactions, and unit balances for each user.

Monitoring: Tracks overdue payments and flags accounts for disconnection. Automation: Integrates with the system to trigger relay-based disconnection when a user's balance is zero or negative. Provides historical data and ensures automated processes like billing and electricity cut-off are executed smoothly.

3.1.9 EXPLANATION OF THE FLOW CHART OF THE DESIGNED PROPOSED SYSTEM:

The combined system begins with the initialization of the energy meter, which captures energy consumption through pulse measurements and performs basic calculations. The usage data is displayed locally on a display for consumer reference and transmitted to a remote server via communication module. On the server side, the data is received, processed further, and stored in a centralized database while updating previous records and displaying results on the server interface. The system then checks whether the consumer has cleared their energy bill. If the payment is verified, the process returns to monitoring energy usage; otherwise, a notification is sent to the consumer via text, detailing the outstanding bill and payment instructions. This fully automated system provides real-time monitoring, centralized data management, and consumer notifications, ensuring efficient energy usage tracking and billing.

FLOW CHART OF THE DESIGNED PROPOSED SYSTEM

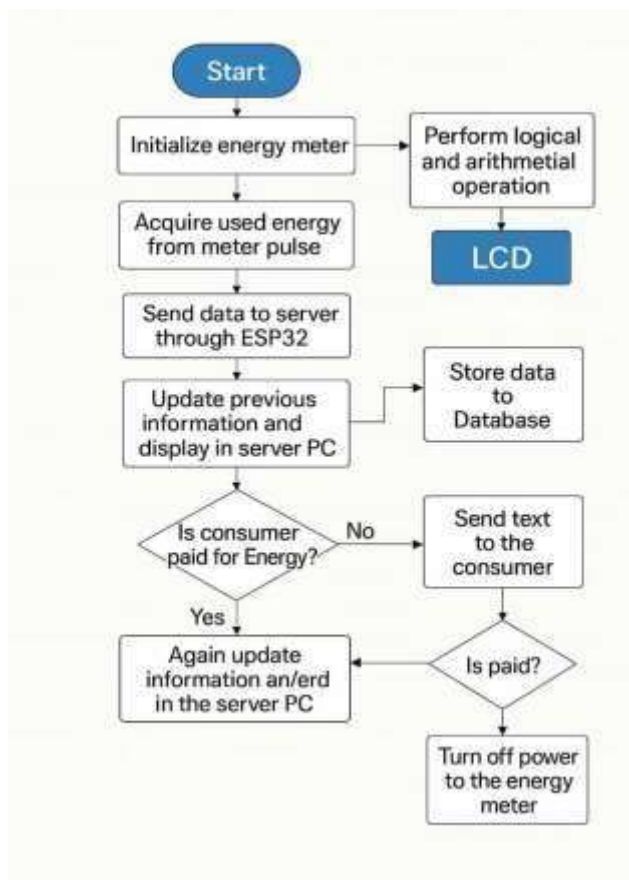


Figure 3: flow chart of the designed proposed system

FLOW CHART OF DATABASE

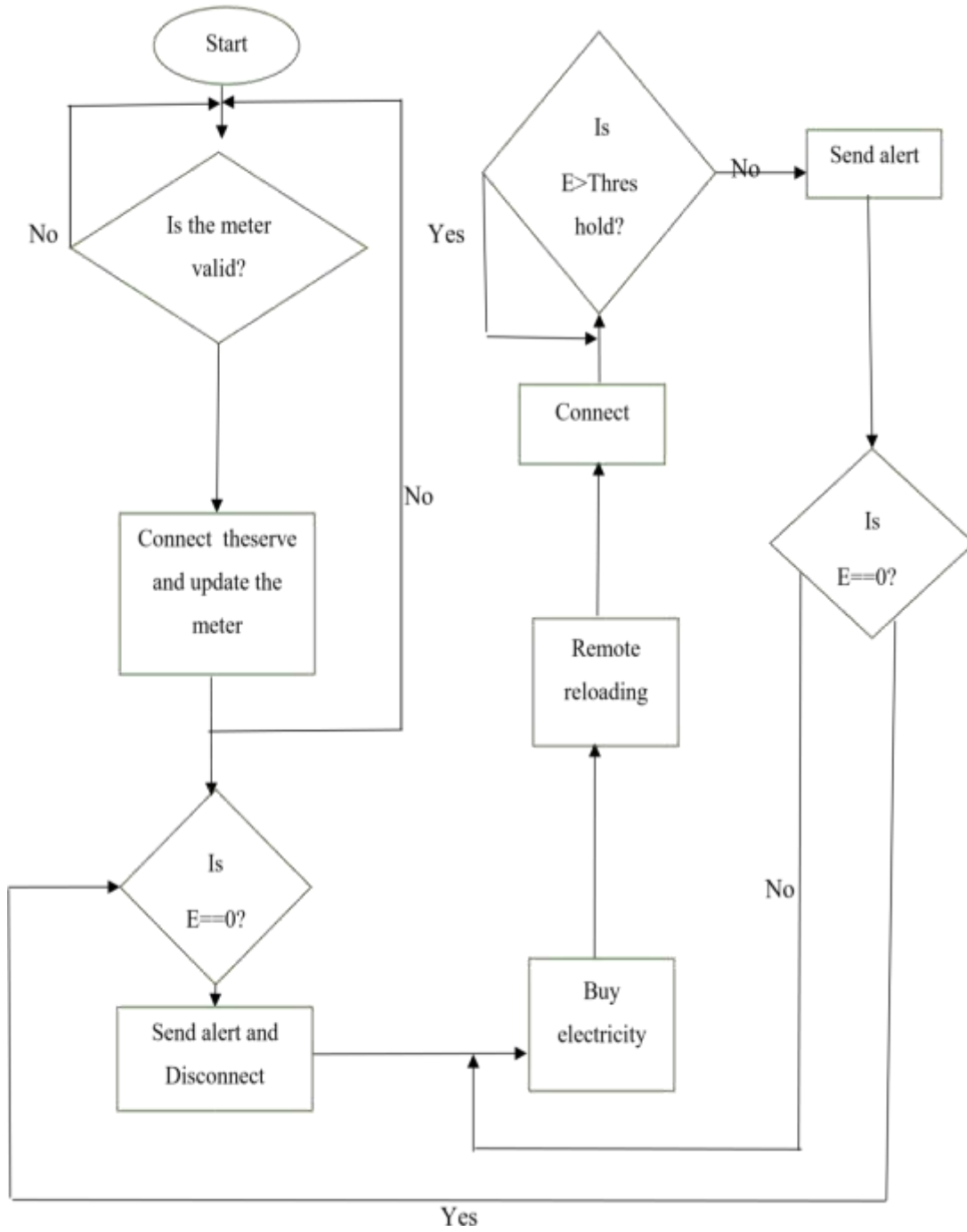


Figure 4: flow chart of the database

CHAPTER FOUR

DATA COLLECTION

4.0 INTRODUCTION

The process of data collection for this system involves identifying the key inputs and parameters required to design, test, and deploy the system effectively. It includes both primary data (collected first hand through experiments, surveys, and direct observation) and secondary data (obtained from existing resources like reports, articles, and standards).

4.1 PRIMARY DATA COLLECTION

The existing systems commonly used for energy metering in multi-tenant facilities lack advanced features such as real-time monitoring and automatic cut-off mechanisms for non-paying users. The data collected for such systems is based on observations, user feedback,

4.2 DATA COLLECTED THROUGH OBSERVATION

This report outlines the insights gained through observing tenants' interactions with the existing electricity metering and billing systems, with a focus on identifying challenges and potential areas for improvement. The aims to gather data on how tenants currently monitor their electricity usage, pay bills, handle disconnections, and interact with the system without influencing their actions. Observations were conducted at residential buildings where the current electricity metering systems are in use.

The focus was on behaviors and environmental factors that affect electricity usage and billing. The researcher utilized non-participant observation, where tenants were observed without direct interaction. A checklist was used to record specific behaviors and environmental conditions as shown on the table.

Table 4.2.1: Shown Data Collected Through Observation

S/N	Observation Aspect	Details Observed
1	Electricity Payment System	Most tenants use prepaid electricity systems with tokens purchased manually from vendors or mobile apps.
2	Monitoring of Electricity Consumption	Tenants rely on physical meters, and some find it difficult to interpret the readings
3	Accessibility of Meters	Meters are often installed in inconvenient or inaccessible locations, making regular monitoring difficult.
4	Real-Time Consumption Awareness	Tenants are not aware of real-time electricity usage, leading to unexpected depletion of credits.
5	Electricity Cut-offs	Frequent disconnections occur due to lack of credit, with no prior notifications or alerts.
6	Usage Awareness	Tenants do not monitor their electricity usage regularly and are unaware of consumption patterns.
7	Billing Issues	Some tenants expressed frustration with inaccurate token balances and occasional delays in activation.

8	Preferences for System Improvements	Tenants expressed interest in a system providing real time monitoring, low-balance notifications, and easy payment methods.
9	Energy Conservation Behavior	Minimal effort is observed among tenants to conserve energy due to lack of feedback on consumption rates.

Table 4.2 .2: Data Collected Through Questionnaires

S/N	Question	ns/Responses in Tsh	Number of Respondents	Percentage
1	How do you currently pay for your electricity?	Prepaid	35	70%
		Post paid	15	30%
2	Do you face difficulties in accessing your electricity bill or meter readings?	Yes	40	80%
		No	10	20%
3	What is your average monthly electricity cost?	Below 15000/=	10	20%
		20,000-50,000	30	60%
		Above 50,000/=	10	20%
4	Have you experienced electricity cut-off due to billing issues?	Yes	38	76%
		No	12	24%
5	Would you prefer a real-time monitoring system for electricity usage?	Yes	45	90%
		No	5	10%

6	Would you be comfortable using a mobile app or service to monitor your electricity usage and receive notifications?	Yes	50	100%
7	What features would you like in a smart metering system?	Real-time monitoring	35	70%
		Low balance notifications	15	30%
8		Less than	40	80%
	How much extra would you be willing to pay for a smart metering system?	15000/=month		
		15000/=month	10	20%

4.3 EXPLANATION OF THE HISTOGRAMS BASED ON RESPONDENTS' Data

The histograms visually represent the responses collected from a questionnaire with sample of 50 people regarding preferences and experiences with electricity payment methods and metering systems. Refer on Figure 5: Bar Graph showing checklist during Observation, this shows Questions 1 to 3 as follow;

Question 1. (How do you currently pay for your electricity?)

The majority of respondents (35 out of 50, 70%) use prepaid electricity, while 15 respondents (30%) use postpaid electricity.

This shows that prepaid payment methods are the most common among the participants. **Question**

2. (Do you face difficulties in accessing your electricity bill or meter readings?)

a) A significant majority (40 out of 50, 80%) indicated Yes, while only 10 respondents (20%) said No.

This highlights that accessibility to billing or meter readings is a widespread challenge.

Question 3. (What is your average monthly electricity cost?)

1. 10 respondents (20%) reported costs below Tsh 15,000.
2. A majority of 30 respondents (60%) reported costs in the range of Tsh 15,000-20,000.
3. Another 10 respondents (20%) reported costs above Tsh 20,000.

Most respondents fall into the middle category of electricity expenses, indicating moderate consumption levels. Figure 5 shows the visualization of data for question 1 up to 3.

BAR GRAPH SHOWING DATA DURING OBSERVATION

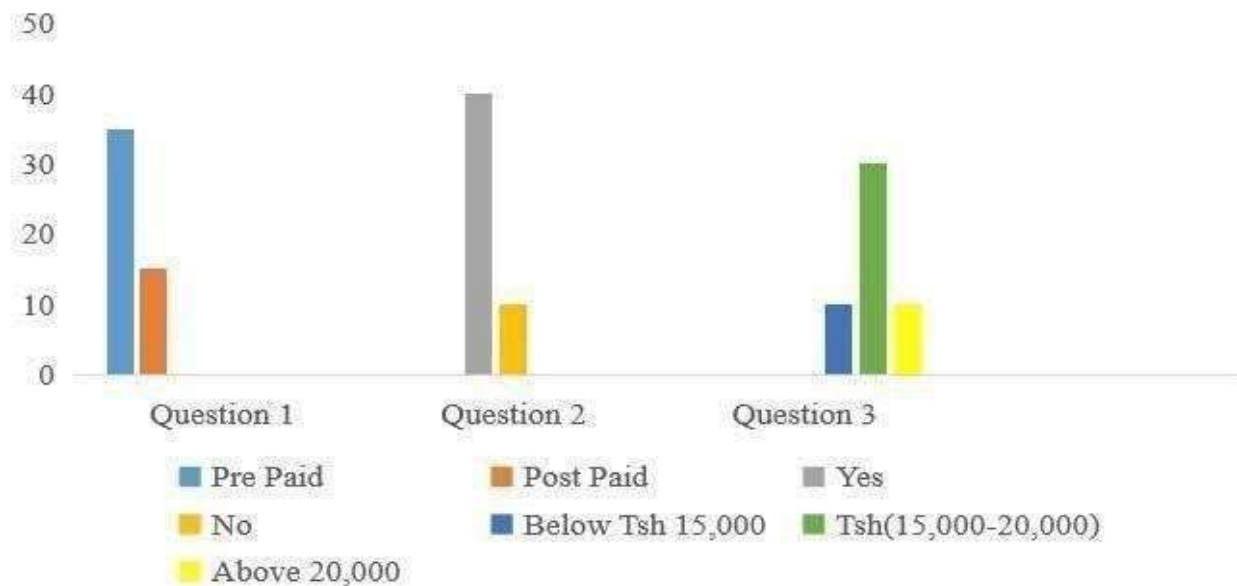


Figure 5: Bar Graph showing checklist during Observation

Question 4 (Have you experienced electricity cut-off due to billing issues?)

38 respondents (76%) experienced billing-related electricity cut-offs, while 12 (24%) did not.

This shows that most participants have faced issues related to delayed or incorrect billing.

Question 5 (Would you prefer a real-time monitoring system for electricity usage?)

The overwhelming majority (45 out of 50, 90%) expressed interest in a real-time monitoring system, while only 5 respondents (10%) said No.

This indicates a strong demand for real-time monitoring of electricity usage.

Question 6 (Would you be comfortable using a mobile app or SMS service to monitor your electricity usage and receive notifications?)

All 50 respondents (100%) answered Yes, showing unanimous interest in technology-driven solutions for electricity monitoring and notifications.

Question 7 (What features would you like in a smart metering system?) 1) 35 respondents (70%) preferred real-time monitoring.

30 respondents (60%) wanted low-balance notifications.

These preferences reveal a high demand for functionality that improves usability and provides actionable insights. In figure 6 shows the visualization on data question 4 up to 7.

BAR GRAPH SHOWING DATA DURING OBSERVATION

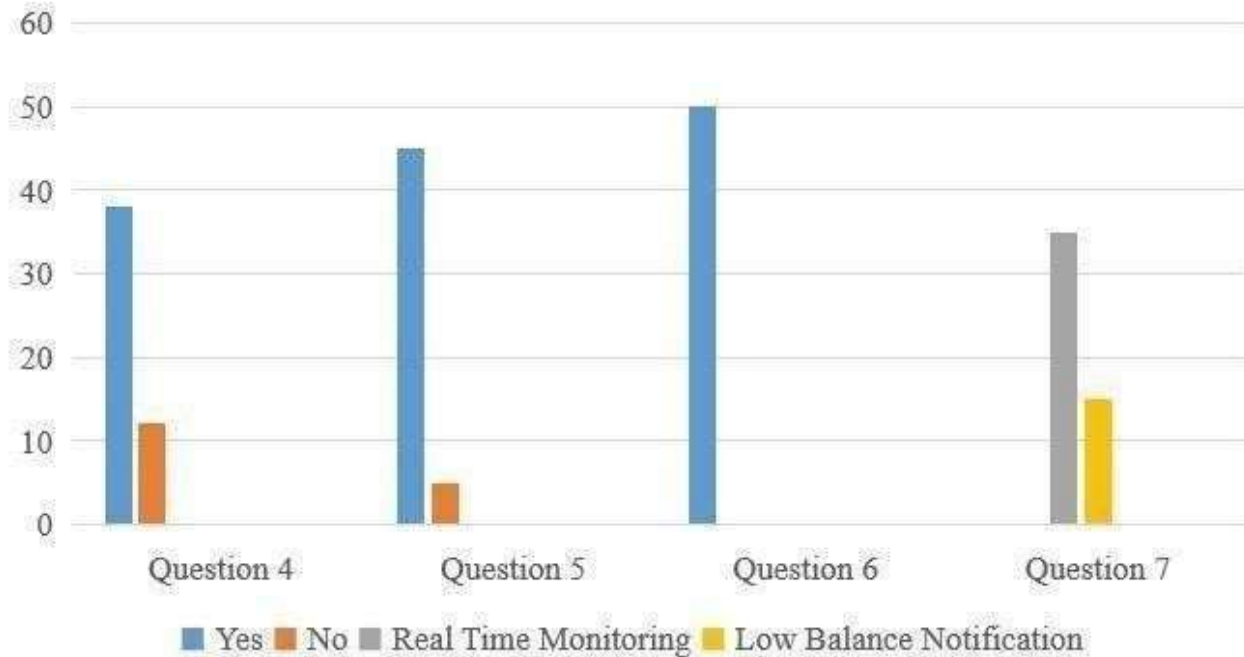


Figure 6: Bar Graph showing checklist during Observation

4.4 SECONDARY DATA COLLECTION

Secondary data collection involved the data obtained from the review of written documents and online content such as articles and journals. Secondary data was collected for the purpose of helping with the design of the project prototype. From the field the data collected suggested to have the following features

4.4.1 ENERGY SUB-METERS

In the proposed Electricity Sub-Metering System, the energy sub-meter serves as the primary sensing unit. It continuously monitors the tenant's energy consumption and sends corresponding digital pulse signals to the control unit for real-time processing and energy calculation.

Based on the system requirements, the following specifications were gathered for the sensing/input unit:

i. High Accuracy and Reliability:

The sub meter conforms to Class 1 accuracy, making it suitable for accurate energy billing in both residential and commercial tenant settings. This ensures that consumption data is precise and trustworthy for automated billing and monitoring.

ii. Pulse-Based Measurement for Digital Integration:

The sub-meter outputs pulses per kWh, where each pulse represents energy. This makes it ideal for digital systems, as the pulse output can be read via GPIO interrupts on a microcontroller.

iii. Voltage and Current Compatibility:

Rated for 220V AC and 5(60) A, the meter is compatible with standard tenant energy consumption levels and protects against overload conditions.

iv. Real-Time Data Acquisition:

The pulse signal allows continuous real-time tracking of energy usage, enabling features like low credit alerts, usage visualization, and remote monitoring.

4.4.2 CONTROL UNIT

The control unit is the central part of the system responsible for processing input signals, managing data flow, and coordinating the activities of other modules such as the sensing unit, display, keypad, and relay. It acts as the brain of the system, interpreting instructions and executing operations in real time.

Based on the design of the Electricity Sub-Metering System, the following data were collected regarding the control unit:

i. Real-Time Performance:

Since the system handles multiple user interactions, payment validations, and energy monitoring events within short intervals, the control unit must support real-time operations to ensure timely responses, accurate record-keeping, and smooth system performance.

ii. Programmability:

To support updates, improvements, or feature changes based on system requirements, the control unit must be reprogrammable. This allows the firmware or logic to be modified without replacing the hardware, ensuring flexibility and long-term adaptability of the system.

4.4.3 DISPLAY UNIT

The display unit is an essential output component in the Electricity Sub-Metering System. It serves as the user interface, presenting important information to the tenant or operator such as energy usage, available credit, payment status, error messages, and system instructions.

It enhances system usability by providing real-time visual feedback and guiding the user during interactions with the system.

i. Alphanumeric Display Capability:

ii. Interface Compatibility:

- a) To ensure smooth communication with the control unit, the display should support standard communication protocols such as:
- b) I²C (Inter-Integrated Circuit)
- c) SPI (Serial Peripheral Interface)
- d) Parallel Communication (if required for higher speed or larger screens)

4.4.4 RELAY MODULE

The relay module serves as a crucial control component in the Electricity Sub-Metering System. It acts as a switching mechanism, enabling or disabling the flow of electricity to individual tenant circuits based on the system's logic. The relay operates under commands from the control unit, depending on parameters such as energy credit status, overload conditions, or remote instructions.

Relay module must:

- a) Automatically disconnects power when a tenant's energy credit is depleted.
- b) Restores power once payment is confirmed or credit is reloaded.
- c) Provides a manual override option for administrative control (optional).
- d) Supports load shedding and fault isolation when needed.

4.4.5 CENTRALIZED DATABASE

A centralized database is used to store, retrieve, and manage system data. It serves as the main storage hub for all essential information related to users, energy usage, payment records, meter readings, and system activity logs.

In the proposed Electricity Sub-Metering System, the centralized database plays a key role in ensuring organized data management, enabling fast queries, secure access, and reliable record-keeping.

Based on the system design, the following data requirements were identified for the centralized database:

i. Multi-User Access:

Since the system may involve administrators, tenants, and automated processes accessing the database simultaneously, it must support concurrent multi-user access without performance issues.

ii. Scalable Storage Capacity:

To handle records of over 50 tenants and store detailed transaction histories, usage logs, and system events, the database must have scalable storage to accommodate growing data over time.

iii. Security and Privacy:

To protect sensitive user data (such as tenant identity, consumption history, and payment records), the database must support user authentication, role-based access control, and audit logging to track any changes made.

iv. Backup and Recovery:

The system must include backup and recovery mechanisms to ensure that data is protected from loss or corruption and that services can resume quickly in case of system failure or unexpected shutdowns.

4.4 CHAPTER SUMMARY

This chapter has discussed the data collection. It contains a summary of the content collected through a variety of data collection methods. The collected data has been clearly presented in tables and graphs. The data collected will be used in the design of the proposed system. The next chapter explains about data analysis and proposed circuit design.

CHAPTER FIVE

DATA ANALYSIS

5.0 INTRODUCTION

The system is designed for rental houses which ensures that each tenant's electricity usage is monitored, billed, and controlled individually. If a tenant's prepaid electricity balance runs out, the system automatically disconnects power to that specific unit while keeping others operational.

The system also integrates real-time monitoring via a mobile application, enabling users to check their electricity consumption and make payments through mobile transactions to recharge their accounts. Data analysis in this system involves collecting, processing, and interpreting electrical parameters to provide accurate billing and efficient electricity management.

5.1 ENERGY SUB-METER UNIT

A sub-meter is an energy measurement device used to monitor and record the amount of electricity consumed by an individual user or tenant within a shared building or premises. Unlike the main utility meter, which measures total building consumption, the sub-meter provides tenant-specific energy usage data.

5.1.1 SUB-METER COMPARISON

To function effectively within the Electricity Sub-Metering System, the sub-meter must:

- i. Comply with Class 1 accuracy to ensure precise energy measurement.
- ii. Operate at a standard voltage rating of 220 V AC and handle current up to 60 A.
- iii. Provide a pulse output of pulses per kilowatt-hour, enabling real-time energy tracking.
- iv. Be compatible with microcontroller input for accurate pulse counting and data processing.
- v. Offer reliable performance under normal residential conditions with appropriate safety features.

5.1 SUB-METER COMPARISON

Feature	Analog (Electromechanical) Meter	Smart Energy Meter	Digital Sub-Meter with Pulse Output
Technology	Rotating disc	Fully digital with IoT/GSM	Digital with pulse or serial output
Accuracy Class	Class 2 (less precise)	Class 1 or better	Class 1 (IEC 6205321 compliant)
Data Output	No digital output	Wi-Fi/GSM/Lora WAN	Pulse output (e.g., 1600 pulses/kWh)
Integration with MCU	Not possible	Requires custom API or gateway	Easy GPIO-based pulse counting
Energy Calculation	Manual (meter reading)	Automatic and cloudbased	Automatic (pulse × constant)
Cost	Low	High	Moderate

SELECTED SUB-METER TYPE: DIGITAL SUB-METER WITH PULSE OUTPUT

Based on the comparison table 5.1, the Digital Sub-Meter with Pulse Output was selected for the following reasons:

- a) High Accuracy: Compliant with Class 1 (IEC 62053-21), suitable for billing purposes and trusted energy monitoring.
- b) Microcontroller-Friendly: Outputs digital pulses that can be easily read by the control unit (e.g., ESP32) using simple interrupt or counter-based logic.
- c) Real-Time Monitoring: Enables accurate and continuous tracking of energy usage for each tenant
- d) Low Cost and Simplicity: Easier to integrate and maintain compared to cloud-based smart meters.
- e) Scalability: Multiple sub-meters can be managed in a multi-tenant system with minimal wiring and complexity.

ENERGY CALCULATION

The meter generates 1600 pulses per kilowatt-hour (kWh). Each pulse corresponds to 0.000625 kWh. This pulse output allows for digital counting and energy calculation in the microcontroller.

Formula:

$$\text{Energy (kWh)} = \text{Number of Pulses} \times 0.000625$$

Energy (kWh)=Number of Pulses×0.000625 Example:

If the meter registers 800 pulses:

$$\text{Energy} = 800 \times 0.000625 = 0.5 \text{ kWh}$$

$$\text{Energy}=800 \times 0.000625=0.5 \text{ kWh}$$



Figure 7: Sub meter

5.2 DISPLAY UNIT

The display unit is a user-interface component used to present information visually to the user, including system instructions, current energy credit, transaction status, and warnings (e.g., low balance or disconnection notices). It allows users to interact with the system more easily by providing real-time feedback and simple text-based guidance.

In this system, the display device communicates with the control unit using the Inter-Integrated Circuit (I²C) protocol, which minimizes wiring complexity while maintaining reliable data exchange.

5.2.1 COLLECTED DATA

The following key requirements were identified from the design needs of the display unit:

- i. Should support the display of alphanumeric characters, including instructions, numerical data, and system status.
- ii. Should support standard communication interfaces (specifically I²C) for easy and stable integration with the microcontroller.

5.2.2 REQUIREMENTS OF THE DISPLAY UNIT

To function effectively within the Electricity Sub-Metering System, the display unit must:

- i. Be capable of displaying both letters and numbers clearly.
- ii. Be large enough to display multiple lines of information simultaneously for better user interaction.
- iii. Be compatible with microcontroller interfaces, especially I²C or parallel communication modes.
- iv. Provide adjustable brightness and contrast to support visibility in various lighting environments.

5.2.3 DISPLAY MODULE COMPARISON

Several display types were evaluated to determine the most suitable option for this system. The comparison was based on size, voltage compatibility, resolution, display features, and communication interface.

TABLE 5.2.3: SPECIFICATIONS OF DIFFERENT DISPLAY MODULES

Specification	16×2 LCD	20×4 LCD	7-Segment Display	OLED Display
Display Size	16 characters × 2 rows	20 characters × 4 rows	Varies (1–8 digits)	Varies
Operating Voltage	4.7V to 5.0V	5.0V	Varies	3.3V / 5.0V
Contrast Control	Adjustable	Adjustable	Not Applicable	Built-in
Display Features	Alphanumeric	Alphanumeric	Numeric only	Alphanumeric + Graphics
Resolution	5×8 pixel per character	5×8 pixel per character	Segment-based	Varies

Communication	Parallel, I ² C	Parallel, I ² C	GPIO only Low	I ² C, SPI
---------------	----------------------------	----------------------------	---------------	-----------------------

SELECTED DISPLAY: 20×4 LCD

Based on the above comparison, the 20×4 LCD was selected as the preferred display module due to the following reasons:

- Provides more display space (4 lines × 20 characters) than the 16×2 LCD, allowing simultaneous display of messages, balance, status, and instructions.
- Fully supports alphanumeric output with clear character resolution.
- Compatible with I²C interface, reducing wiring to just 4 connections: VCC, GND, SDA, and SCL.
- Operates reliably at 5V, matching standard logic and power supply levels used in the system.
- Cost-effective and widely supported in microcontroller environments.



Figure 8: LCD Display 20 X 4

5.3 CONTROL UNIT

The control unit functions as the central processor of the Electricity Sub-Metering System. It is built around a microcontroller that receives input signals (such as pulse counts from the energy meter or commands from the keypad), processes them based on programmed instructions, and generates corresponding output actions such as updating the display, controlling the relay, or storing data in memory.

The control unit creates a logical link between the input devices (e.g., sub-meter, keypad) and output modules (e.g., display, relay), ensuring efficient and reliable system operations.

5.3.1 COLLECTED DATA

Based on the functional needs of the system, the control unit must fulfill the following requirements:

- i. Should include integrated communication interfaces to support data exchange between modules (e.g., Wi-Fi, UART, SPI).
- ii. Should support a real-time operating system (RTOS) to handle concurrent tasks like energy monitoring, relay control, display updates, and user inputs efficiently.
- iii. Must be reprogrammable to allow updates, modifications, or system upgrades as needed.

5.3.2 REQUIREMENTS OF THE MICROCONTROLLER

To meet system demands, the microcontroller selected for the control unit must:

- i. Allow firmware updates, ensuring that the system can be improved or patched in the future.
- ii. Include integrated Wi-Fi for remote communication, monitoring, and control.
- iii. Operate with a suitable clock frequency and sufficient RAM to support multitasking in a real-time environment.

5.3.3 MICROCONTROLLER COMPARISON

A comparison was conducted between commonly used microcontrollers to evaluate their suitability for the sub-metering application. The comparison considered critical factors such as operating voltage, memory, communication features, and processing speed.

TABLE 5.3.3: SPECIFICATIONS OF DIFFERENT MICROCONTROLLER MODULES

Specifications	PIC16F877A	ATmega328P	Raspberry Pi	ESP32
Operating Voltage	5V	5V	5V via USB-C	3.3V to 5V

Flash Memory	14 KB	32 KB	MicroSD Card	4 MB
Clock Speed	20 MHz	16 MHz	1.5 GHz	80 MHz (dual-core)
Communication Interfaces	UART, SPI, I ² C	UART, SPI, I ² C	UART, SPI, I ² C	UART, SPI, I ² C, Wi-Fi
RAM	368 Bytes	2 KB	1.8 GB	520 KB SRAM

CONCLUSION: ESP32 MICROCONTROLLER SELECTION

Based on the comparison, the ESP32 microcontroller was selected due to the following reasons:

- a. Higher execution speed for fast response and real-time operations.
- b. Low power consumption, ideal for embedded and always-on systems.
- c. Greater memory capacity to support RTOS and multitasking.
- d. Built-in Wi-Fi for wireless communication without extra modules.
- e. High compatibility with peripherals like relays, displays, meters, and GSM modules.

5.3.4 ESP32 MICRO-CONTROLLER DESCRIPTION

ESP32 has powerful on-board processing and storage capabilities that allow it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

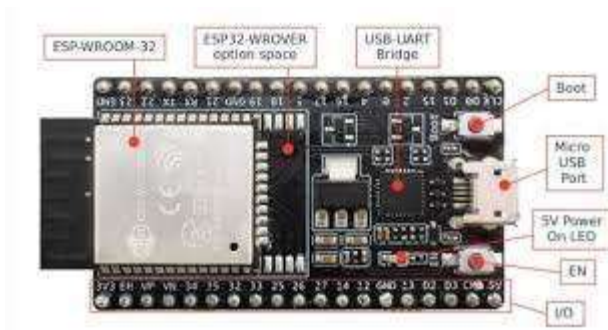


Figure 9: ESP32

5.4 RELAY MODULE

The relay module is a critical component in the Electricity Sub-Metering System. It serves as an electrical switch that enables or disables the power supply to a tenant's circuit based on signals from the control unit. The relay responds to system logic such as credit depletion, overload, or remote disconnection by physically opening or closing the circuit.

5.4.1 COLLECTED DATA

Based on the control requirements of the system, the following data were identified for the relay module:

- a) Must support a current rating of 10A, which is sufficient for residential or small commercial energy loads.
- b) Should have a switching response time ≤ 10 milliseconds for reliable and timely operation.
- c) Must provide opto-isolation between the control signals and the high-voltage switching side for system safety.
- d) Should include visual status indication (LED) to show ON/OFF relay status.
- e) Must be compatible with low-voltage (3.3V or 5V) digital signals from the control unit.

5.4.2 REQUIREMENTS OF THE RELAY MODULE

To operate reliably and safely in the proposed system, the relay module must meet the following:

- a) Rated Load Capacity: 10A at 220V AC
- b) Control Signal: Accepts 3.3V or 5V logic level
- c) Isolation: Opto-isolated control input to protect the microcontroller

- d) Durability: Mechanically rated for thousands of switching cycles
- e) Form Factor: Compact and DIN-rail mountable (if enclosed in panel)

TABLE 5.4 THE RELAY MODULE

specification	Electromechanical Relay (EMR)	Solid-State Relay (SSR)	Magnetic Contactor
Switching Capacity	Up to 30A	Up to 10A	20 A to 100 A (3-phase)
Switching Time	5	< 1 ms	20–30 ms
Isolation Type	Mechanical and electrical	Optical (opto-isolated)	Magnetic coil isolation
Control Voltage	3.3V / 5V	3.3V / 5V	Requires external driver

CONCLUSION: SOLID-STATE RELAY (SSR):

- a) Adequate for typical residential loads (lights, chargers, sockets, etc.).
- b) Compatible with control signals from the microcontroller (ESP32).
- c) Provides safe disconnection during credit expiration or overload.
- d) Affordable and easy to maintain.
- e) Integrates well with existing metering and display systems.

5.4.3 RELAY DESCRIPTION

A relay is an electromechanical switch that allows a low-power control signal from a microcontroller (such as a digital output pin) to switch a high-power electrical load. In the Electricity Sub-Metering System, the relay is used to control the supply of electricity to each tenant’s power line.

The relay operates by using a small electrical current to energize a coil, which in turn mechanically closes or opens an internal contact, allowing or interrupting the flow of 220V AC power. This switching function enables the system to automatically connect or disconnect a tenant's electrical supply based on their energy credit status, system safety conditions, or remote commands.



Figure 10: Relay switch

CHAPTER SIX

DESIGN AND SIMULATION

6.0 INTRODUCTION

To validate the hardware logic of our electricity sub metering system, we performed a detailed circuit simulation using Proteus Design Suite. The objective was to verify core functionalities such as pulse detection, relay control, voltage regulation, and the system's automatic response when prepaid units are exhausted.

6.1 SIMULATION OBJECTIVES

- i. Validate pulse input detection using GPIO interrupts.
- ii. Test relay switching logic when units are available or depleted.
- iii. Simulate regulated power supply to ESP32 (via AMS1117).
- iv. Verify transistor-driven relay operation with flyback protection.
- v. Confirm system behavior under typical load consumption scenarios.

6.2 CIRCUIT COMPONENTS USED IN SIMULATION

In our Proteus-based circuit simulation, we carefully selected components that represent the core hardware functionality of the Electricity sub metering system. Although the ESP32 microcontroller is not directly available in Proteus, we used a suitable equivalent (Arduino UNO/Nano) to emulate its GPIO behavior and logic control.

Below is a description of each component and its role in the simulation:

a) **ESP32 (Simulated as Arduino UNO/Nano):**

Served as the central processing unit, responsible for detecting pulse signals (representing energy usage) and making relay control decisions based on available energy balance.

b) **PULSE Signal Generator:**

Generated periodic digital pulses to simulate the output of an actual energy meter. The frequency was calibrated to mimic real consumption patterns, using the standard of 1600 pulses per kilowatt-hour

c) **2N2222 NPN Transistor:**

Functioned as an electronic switch to drive the relay coil. It was controlled by a digital output from the simulated ESP32 and allowed for safe current switching without overloading the microcontroller pin.

d) **1N4007 Diode:**

Used as a flyback diode across the relay coil to suppress voltage spikes (back-EMF) when the relay is de-energized, ensuring protection for the switching transistor and overall circuit stability.

5V SPDT Relay:

Simulated the actual load control mechanism. It was toggled ON or OFF depending on the tenant's available energy units, enforcing disconnection when units dropped to zero.

f) **AMS1117 (3.3V Voltage Regulator):**

Used to step down the 5V supply to a stable 3.3V required for the ESP32 module. This ensured safe and reliable operation of the controller under simulated conditions.

g) **Virtual LEDs:**

Represented the status of the load connected through the relay. They provided a clear visual indication of whether the power was active (relay ON) or disconnected (relay OFF).

h) **Multimeter and Oscilloscope Tools:**

Employed to measure voltage levels, monitor pulse signals, and verify real-time changes during simulation. These instruments helped validate switching times and signal integrity throughout the test.

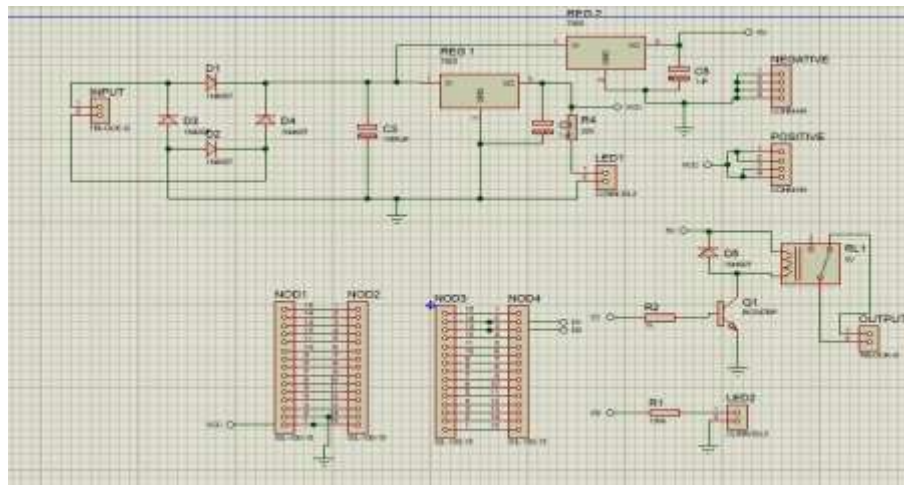


Figure 11: PCB design systematic

6.3 SIMULATION SETUP

i. **Pulse Input:**

A square wave was generated to simulate meter output pulses connected to an interrupt-enabled GPIO pin.

ii. **Relay Logic:**

When unit balance > 0 , GPIO output remained high to activate the transistor and keep the relay closed (load connected).

When unit balance = 0, the GPIO pin went LOW, turning off the transistor and opening the relay (disconnecting the load).

iii. **Power Regulation:**

A 5V DC input was passed through an AMS1117 regulator to supply the ESP32 module with 3.3V. Bypass capacitors were added for stability.

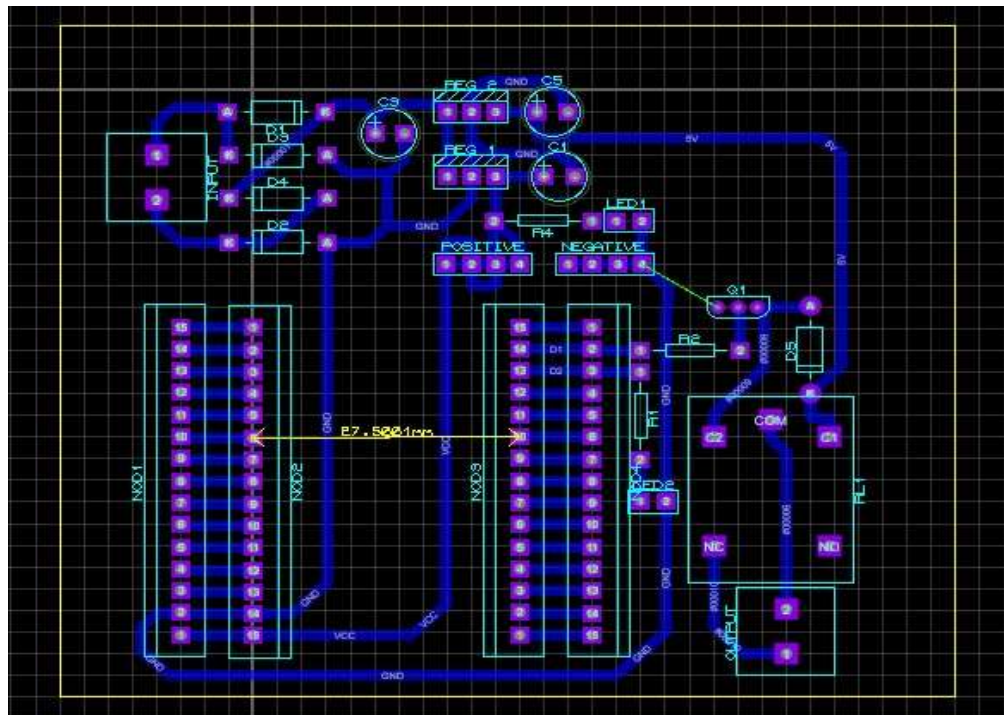


Figure 12: PCB layout

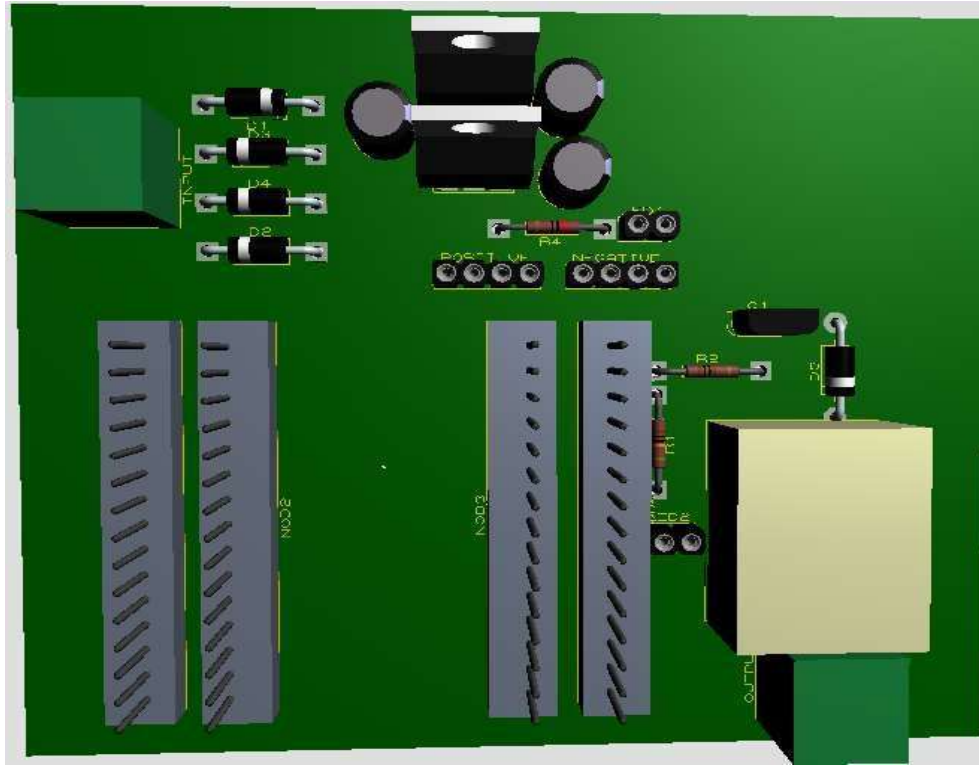


Figure 13: PCB 3D

6.3 TEST SCENARIO AND RESULTS

To verify the reliability and performance of the Electricity sub metering system, a series of functional simulations were conducted using Proteus. The following test scenarios represent typical operational and edge-case conditions, along with their observed outcomes:

1. Normal Consumption

- i. Condition: Pulse signal is active and energy units are above zero.
- ii. Expected Behavior: Relay remains energized, allowing the load to operate.
- iii. Simulated Output: Relay stayed ON and system functioned as expected

Result: Pass

2. Zero Balance

- i. Condition: Energy balance reaches zero.
- ii. Expected Behavior: Relay deactivates automatically, disconnecting the load.
- iii. Simulated Output: Relay turned OFF upon unit depletion Result: Pass

3. Recharge Simulated

- i. Condition: Units restored through simulated token recharge.
- ii. Expected Behavior: Relay reactivates and power is restored to the load.
- iii. Simulated Output: Relay resumed operation after balance update Result: Pass

4. Pulse Interruption (Idle State)

- i. Condition: No pulses detected from the energy meter.
- ii. Expected Behavior: System remains idle and maintains the last known state.
- iii. Simulated Output: System remained stable with no unintended switching Result: Pass

CHAPTER SEVEN

PROTOTYPE IMPLEMENTATION AND TESTING

7.1 INTRODUCTION

This prototype showcases an electricity sub metering system built around three ESP32 microcontrollers two acting as sub-metering nodes for individual tenants, and one serving as the main controller unit.

The system is designed to:

- i. Monitor real-time electricity usage for two separate tenants.
- ii. Calculate and display both individual and total energy consumption.
- iii. Track and update available energy units for each tenant.
- iv. Control electrical loads through relays, enabling or disabling power based on energy availability or predefined conditions.

With wireless communication between the nodes and the main controller, this system offers a scalable and intelligent solution for prepaid metering and energy management in shared residential or commercial setups. Within this chapter we have the **hardware part** and **software part**.

7.2 HARDWARE PART

7.2.1 HARDWARE SPECIFICATION

The prototype utilizes the following components for implementing the electricity sub metering system:

System Components

- i. ESP32-WROOM-32 (x3): Acts as the core microcontroller for both tenant sub-meters and the main control unit.
- ii. AMS1117 3.3V Regulators (x3): Provide stable 3.3V power supply to each ESP32 module.
- iii. 5V SPDT Relays (x2): Used to switch electrical loads for each tenant based on available energy credits.
- iv. 2N2222 NPN Transistors & 1N4007 Diodes: Ensure safe and efficient relay operation with flyback protection.
- v. LCD Display (x1): Displays real-time energy consumption and status updates at the central controller.
- vi. Terminal Blocks, Custom PCB, and Passive Components: Facilitate robust wiring, signal conditioning, and circuit integration.

7.2.2 CIRCUIT SUMMARY

Each tenant's ESP32 sub-meter is responsible for counting energy meter pulses, which represent electricity consumption (typically 1600 pulses per kilowatt-hour). These pulse counts are transmitted to the main controller ESP32 using a communication protocol such as Serial, MQTT, or ESP-NOW.

The main controller performs the following key functions:

- i. Calculates real-time energy consumption for each tenant.
- ii. Deducts usage from the available prepaid energy units.
- iii. Controls load relays to disconnect power when a tenant's balance reaches zero.

This architecture enables accurate, tenant-specific sub-metering with centralized control and monitoring.

7.2.3 OPERATION FLOW

- i. Energy Monitoring:

Each tenant ESP32 sub-meter continuously detects and counts energy pulses (1600 pulses = 1 kWh).

- ii. Data Transmission:

The pulse count is sent periodically to the central ESP32 controller via a reliable communication protocol.

- iii. Consumption Calculation:

The controller computes energy consumption using the formula:

$$\text{kWh} = \text{pulse_count} / 1600$$

Balance Update:

The system subtracts real-time consumption from each tenant's available energy balance.

Load Control:

If a tenant's balance reaches zero, the controller automatically disables the respective load by deactivating its relay.

- vi. User Feedback:

An LCD display provides real-time feedback, showing current usage and remaining energy units for both tenants.

7.2.4 SYSTEM WIRING

Tenant ESP32 Nodes

- a) Pulse Counting: Each ESP32 sub-meter is connected to an energy meter's pulse output via a GPIO interrupt pin to count energy usage accurately (1600 pulses = 1 kWh).
- b) Load Control (Relay Switching): Each tenant node controls a 5V SPDT relay using a GPIO pin. I.
 - The relay is driven via a 2N2222 NPN transistor,
 - II. Protected by a 1N4007 flyback diode,
 - III. Used to switch the tenant's electrical load on or off based on energy balance.
- c) Power Supply: Powered by a 5V DC source, regulated to 3.3V using the AMS1117 voltage regulator to supply the ESP32 safely.

Main Controller ESP32

- a) Data Reception: Receives pulse count data from both tenant ESP32 nodes via Serial, MQTT, or ESP-NOW communication.
- b) Energy Processing & Monitoring: Calculates energy usage and updates balances in real-time for each tenant.
- c) Display Interface: Interfaces with a LCD display to show current usage and remaining units for both tenants.



Figure 14: Prototype



Figure 15: Prototype connections



Figure 16: first sub meter

7.2.5 SYSTEM BEHAVIOR

Upon powering the system, the main controller ESP32 initializes and begins monitoring operations. The system functions as follows:

a) Initialization:

The controller sets up communication channels, initializes the OLED display, and prepares internal counters for energy tracking.

b) Real-Time Monitoring:

Each **tenant ESP32 node** begins counting pulses from its respective energy meter (1600 pulses = 1 kWh) using GPIO interrupts to ensure precise measurements.

c) Periodic Processing (Every Minute):

i. The main controller receives pulse counts from both tenant nodes.

ii. It calculates energy consumption per tenant using:

$$\text{kWh} = \text{pulse_count} / 1600$$

iii. Prepaid energy units are updated by subtracting the consumed amount. iv. The LCD display is refreshed to show current consumption and remaining units

d) Load Control Logic:
If a tenant's energy balance reaches zero:

i. The tenant's local ESP32 triggers its relay to disconnect the load, ensuring no further consumption occurs.

ii. The relay remains open until new units are loaded.

This behavior enables a fully automated and tenant-specific energy management system with real-time feedback and reliable load control.

7.2.5 PROTOTYPE SUMMARY:

This prototype successfully demonstrates an IoT-enabled smart electricity sub-metering system tailored for multi-tenant environments. It features:

- i. Accurate pulse-based energy monitoring.
- ii. Relay-controlled load disconnection when prepaid units are exhausted.
- iii. Centralized real-time consumption tracking and display using an ESP32 main controller.
- iv. The system is scalable, cost-effective, and ideal for deployment in residential, commercial, or shared utility settings.

7.3 SOFTWARE PART

7.3.1. INTRODUCTION

The backend of this system is powered by a structured MySQL database, interfaced with IoT-enabled ESP32 microcontrollers. The database plays a critical role in data logging, energy accounting, token validation, and real-time alert management making the system scalable, secure, and adaptable for multitenant environments.

7.3.2. DATABASE DESIGN

We've developed a clean, normalized database schema named `energy_meter`, which integrates the core functionalities required for prepaid sub-metering. Let me highlight the most essential tables:

- i. `Users`: Stores tenant and admin credentials, user roles, and account status.
- ii. `Meters`: Keeps real-time data for each meter: current units, total consumption, pulse counts, and operational status.
- iii. `meter_readings` and `submeter_readings`: Log individual pulse inputs from each ESP32, used to calculate energy in kilowatt-hours.
- iv. `Tokens`: Stores prepaid energy codes with linked units, monetary value, and expiration for secure token validation.
- v. `Transactions`: Records all consumption and recharge events for traceability and billing.
- vi. `Alerts`: Captures system events such as low balance, tampering detection, or communication failures.

- vii. Settings: Allows system-wide configurations like energy pricing, unit-per-pulse ratio, maintenance mode, and time zone.

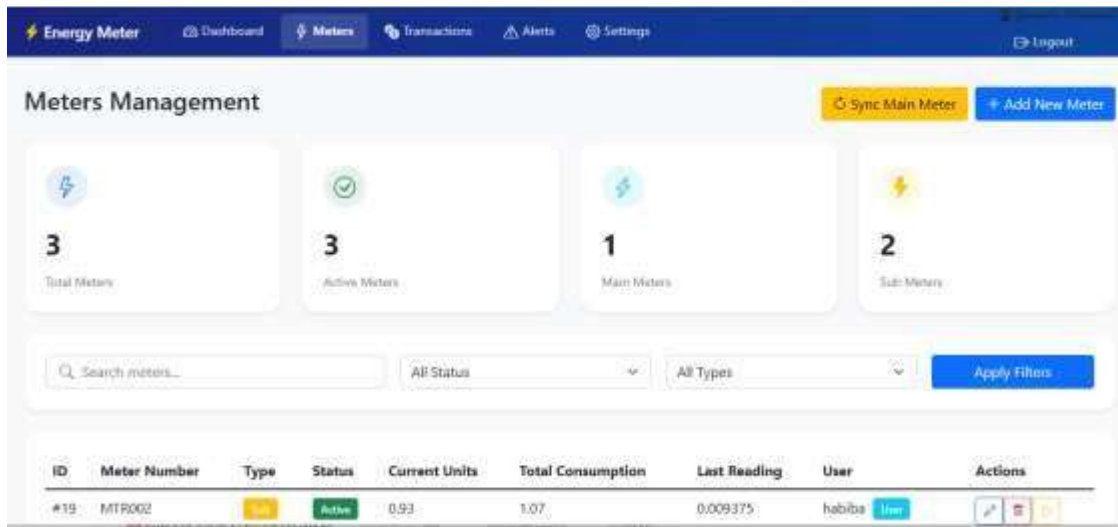


Figure 17: admin interface

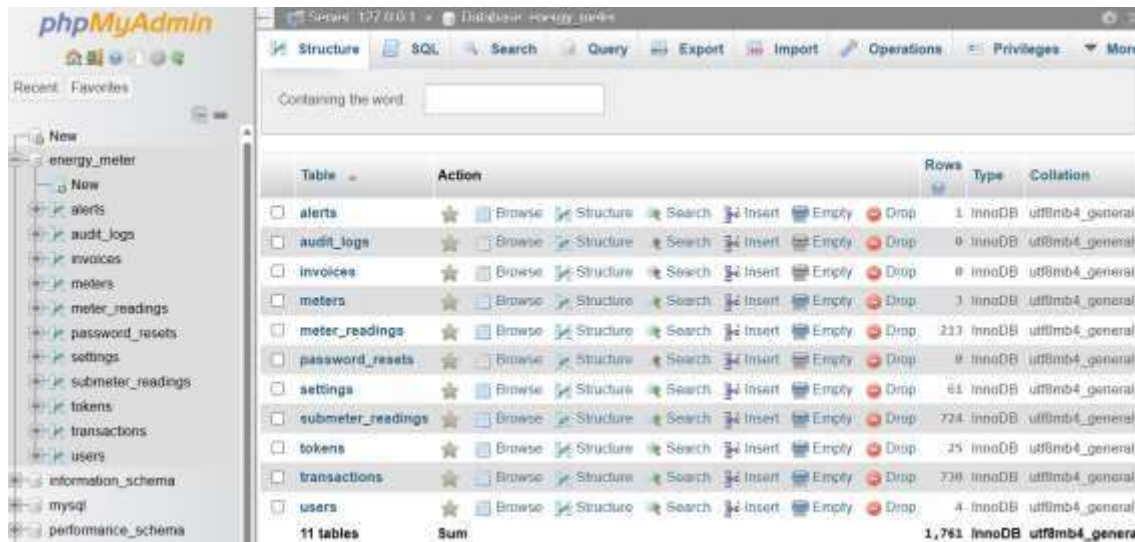


Figure 18: Database structure

7.3.3 SOFTWARE FUNCTIONALITY OVERVIEW

Here's how the software ties everything together:

- i. **Real-Time Monitoring:**
ESP32 nodes count energy pulses (1600 per kWh) and periodically send this data to the main controller, which then updates the database.
- ii. **Prepaid Logic:**
The system validates recharge tokens through the tokens table. Once accepted, units are credited to the user's meter. Usage is deducted in real-time based on pulses received.
- iii. **Automated Load Control:**
When a tenant's available units drop to zero, the tenant's local ESP32 automatically opens the relay cutting off power to prevent overuse.
- iv. **Alert System:**
Critical events such as tampering or connection loss generate alerts, logged with severity levels and timestamps, ensuring system accountability.
- v. **Billing and Accountability:**
With every transaction whether purchase or consumption logged under transactions, administrators can generate detailed reports and invoices with full transparency.
- vi. **Security and Integrity:**
User credentials are encrypted using secure hashing (bcrypt). The database is governed by foreign key constraints for data consistency.

7.3.4 SUMMARY OF SOFTWARE BENEFITS

- i. **Fairness:** Ensures accurate billing and eliminates disputes over shared utility bills.
- ii. **Transparency:** Provides both tenants and admins with clear data on consumption and remaining units.
- iii. **Accountability:** Encourages responsible energy use by linking consumption to real-time balance.
- iv. **Sustainability:** Helps reduce wastage by actively involving users in their consumption patterns
- v.

7.4 PROTOTYPE TESTING AND RESULTS

The prototype reading, we have seen in pulses counting when we are using electric iron for seeing the results in electricity usage.

TABLE 7.4 PROTOTYPE RESULTS

Reading id	Meter id	Energy consumption (kWh)	Reading time	pulses	Timestamp
523	18	0.009375	2025-07-02 14:50:51	15	2025-07-02 04:50:51
525	18	0.009375	2025-07-02 15:10:10	15	2025-07-02 05:10:10
527	18	0.009375	2025-07-02 15:13:43	15	2025-07-02 05:13:43
528	18	0.009375	2025-07-02 15:13:51	15	2025-07-02 05:13:51
529	18	0.009375	2025-07-02 15:13:58	15	2025-07-02 05:13:58

7.5 COMPARISON: PROTOTYPE AND EXISTING SUB-METER READINGS

TABLE 7.5:

Feature	Prototype	Existing Commercial Sub-Meter System
Energy Consumption Logged	Constant value: 0.009375 kWh repeatedly	Incremental values
Pulse Count	Fixed: 15 pulses on every reading	Increasing total pulses
Reading Trigger	Every few seconds or minutes, regardless of change	On change or set interval (e.g., every 5 minutes or per 1 kWh)

CHAPTER EIGHT

CONCLUSION AND RECOMMENDATIONS

The electricity sub metering system for energy consumption has successfully addresses the critical challenges associated with energy consumption in shared rental properties. This system introduces an innovative, efficient, and transparent approach to energy management. The implementation of real-time monitoring, accurate billing algorithms, and an intuitive user interface ensures fair cost allocation and enhances tenant satisfaction.

The system not only improves energy efficiency by promoting accountability but also fosters sustainability by encouraging tenants to monitor and reduce their consumption. Its ability to support up to 50 tenants, provide real-time notifications, and integrate with existing electrical manual interaction with meter while feeding the token where by user must use a keypad for token feeding, due to that some of meters were installed at a certain height make difficulty to the users .Hence the proposed system has more advantages such as its use auto feeding of token to the meter, does not involves the use to keypad,CIU ,high efficiency and transparent to users.

Furthermore, the inclusion of field-testing and refinement ensures the system's reliability and practicality in real-world scenarios. Overall, this project offers a comprehensive, modern solution to energy management, contributing to equitable energy usage and fostering harmony among tenants while supporting sustainable practices.

REFERENCE

- [1] Prathik , M. , Anitha , K. , Anitha , V. , Smart energy meter surveillance using IoT . Proceedings of the International Conference on Power, Energy, Control and Transmission Systems, ICPECTS , pp. 186 – 189 , 2018 .
- [2] Alaudin , A.H.B. , Zan , M.M.M. , Mahmud , A.R. , Yaaya , C. K. H. C. K., Yusof, M.I., Yussoff, Y.M., Real-time residential energy monitoring device using internet of things . ICSET 2018 - 2018 IEEE 8th International Conference on System Engineering and Technology, Proceedings , pp. 97–101, 2019
- [3] Mohammad Hossein Yaghmaee Design and Implementation of an Internet of Things Based Smart Energy Metering” 6th IEEE. International Conference on Smart Energy Grid Engineering 2018.
- [4] Himanshu kpatel “arduino based smart energy meter” 2nd Int'l Conf.on Electrical Engineering and Information &Communication Technology (ICEEICT) 2018.
- [5] Chaouch , H. , Bayraktar , A.S. , Çeken , C. , Energy Management in Smart Buildings by Using M2M Communication . International Istanbul Smart Grids and Cities Congress and Fair, ICSG - Proceedings , pp. 31–35, 2019
- [6] Panahi, F.H., Moshirvaziri, S., Mohammedi, Y., Panahi, F.H., Ohtsuki, T., Smart Energy Harvesting for Internet of Things. Proceedings - 2018 Smart Grid Conference, SGC, **2018**
- [7] Landi, c.: Dipt. Di Ing.dell’Inf, SecondaUniv di Napoli,Aversa,Italy; Merola p. ”ARM-based energy management system using smart meter and Web server”,IEEE instrumentation and measurement technology conference pp.1-5 may 2011.

- [8] B. S. Koay, S. S. Cheah, Y. H. Sng, P. H. Chong, P. Shum, Y. C. Tong, X. Y. Wang, Y. X. Zuo and H. W. Kuek, "Design and implementation of Bluetooth energy meter", IEEE Proceedings of the 4th International Joint Conference of the ICICS, vol. 3, pp. 1474-1477, Dec,2003.
- [9] N. Fathima, A. Ahammed, R. Banu, B.D. Parameshachari, and N.M. Naik, "Optimized neighbor discovery in Internet of Things (IoT)," In Proc. of International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), pp. 1-5, 2017
- [10] Choi, C.S., Jeong, J.D., Lee, I.W., Park, W.K., LoRa based renewable energy monitoring system with open IoT platform. International Conference on Electronics, Information and Communication, ICEIC, pp. 1 – 2, 2018

APPENDICES

APPENDIX A

Table A: Project conceptualization schedule

	PROJECT DURATION PER WEEK																	
ACTIVITY	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Project title selection	■	■	■	■	■	■												
Title defending and redefending				■		■												
First min presentation								■										
Literature review and data collection	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
Min two presentation													■					
Project report writing and proofreading									■	■	■	■	■	■	■			
Project report submission															■			
First semester project presentation																		■

APPENDIX B

Table B: Data Collected Through Questionnaires

S/N	Question	s/Responses in Tsh	Number of Respondents	Percentage
1	How do you currently pay for your electricity?	Prepaid	35	70%
		Post paid	15	30%
2	Do you face difficulties in accessing your electricity bill or meter readings?	Yes	40	80%
		No	10	20%
3	What is your average monthly electricity cost?	Below 15000/=	10	20%
		20,000-50,000	30	60%
		Above 50,000/=	10	20%
4	Have you experienced electricity cutoff due to billing issues?	Yes	38	76%

		No	12	24%
5	Would you prefer a real-time monitoring system for electricity usage?	Yes	45	90%
		No	5	10%
6	Would you be comfortable using a mobile app or SMS service to monitor your electricity usage and receive notifications?	Yes	50	100%
7	What features would you like in a smart metering system?	Real-time monitoring	35	70%
		Low balance notifications	15	30%
8		Less than	40	80%

	How much extra would you be willing to pay for a smart metering system?	15000/=month		
		15000/=month	10	20%

APENDEX C.

Checklist Used During Observations.

S/N	Observation Area	Criteria to Observe	Checked (√/X)	Comments /Notes
1	Meter Accessibility	Meter is easily accessible to tenants.		
		Meter location is safe and secure.		
		Meter is readable without external assistance (e.g., ladder, torchlight).		
2	Real-Time Monitoring	Tenants regularly check their meter readings		
		Tenants are aware of their current consumption		
		Tenants receive updates or notifications about consumption levels.		
3	Disconnection Impact	Tenants are notified before disconnection.		
		Disconnections occur frequently		
		Delays in power restoration after credit top-up		
4	Energy Usage Behaviour	Tenants actively manage energy usage to avoid high consumption.		
		Tenants top up prepaid credit proactively (before disconnection).		

		face challenges understanding the billing system.		
5	System Maintenance	Meters are well-maintained and functional		
		Tenants report issues with meters promptly.		
		Repairs or maintenance are carried out in a timely manner.		
6	Technology Integration	System provides real-time usage feedback (e.g., apps, notifications).		

SOURCE CODES

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Jul 02, 2025 at 12:21 AM
-- Server version: 10.4.32-MariaDB
-- PHP Version: 8.0.30
```

```
SET          SQL_MODE          =
"NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101          SET
@OLD_CHARACTER_SET_CLIENT=@@C
HARACTER_SET_CLIENT */;
```

```
/*!40101          SET
@OLD_CHARACTER_SET_RESULTS=@@
CHARACTER_SET_RESULTS */;
```

```
/*!40101          SET
@OLD_COLLATION_CONNECTION=@@C
```

```

OLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
-- Database: `energy_meter`

-- Table structure for table `alerts`
CREATE TABLE `alerts` (
  `alert_id` int(11) NOT NULL,
  `meter_id` int(11) NOT NULL,
  `alert_type`
enum('tampering','low_units','connection_lost','other')
NOT NULL,
  `alert_message` text NOT NULL,
  `severity` enum('low','medium','high','critical')
DEFAULT 'medium',
  `status`
enum('active','acknowledged','resolved')
DEFAULT 'active',
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `resolved_at` timestamp NULL DEFAULT
NULL,
  `resolved_by` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

-- Dumping data for table `alerts`
INSERT INTO `alerts` (`alert_id`, `meter_id`,
`alert_type`, `alert_message`, `severity`, `status`, `created_at`,
`resolved_at`, `resolved_by`)
VALUES
(1, 11, 'tampering', 'Test: Meter case opened -
possible tampering detected', 'high', 'resolved',
'2025-06-27 12:09:06', '2025-06-27 13:12:47', 4);

-- Table structure for table `audit_logs`
CREATE TABLE `audit_logs` (
  `log_id` int(11) NOT NULL,
  `user_id` int(11) DEFAULT NULL,
  `action` varchar(255) NOT NULL,
  `details` text DEFAULT NULL,

```

```

`ip_address` varchar(45) DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT
current_timestamp()
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

```

```

-- Table structure for table `invoices`
CREATE TABLE `invoices` (
  `invoice_id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `meter_id` int(11) NOT NULL,
  `amount` decimal(10,2) NOT NULL,
  `units` decimal(10,2) NOT NULL,
  `status` enum('pending','paid','cancelled')
DEFAULT 'pending',
  `due_date` date NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

```

```

-- Table structure for table `meters`
CREATE TABLE `meters` (
  `meter_id` int(11) NOT NULL,
  `user_id` int(11) DEFAULT NULL,
  `meter_number` varchar(50) NOT NULL,
  `type` enum('main','sub') NOT NULL
DEFAULT 'sub',
  `location` varchar(255) DEFAULT NULL,
  `current_units` decimal(10,2) DEFAULT 0.00,
  `total_consumption` decimal(10,2) DEFAULT 0.00,
  `last_reading` decimal(10,6) DEFAULT NULL,
  `status` enum('active','inactive','maintenance') DEFAULT
'active',
  `last_update` timestamp NOT NULL DEFAULT
current_timestamp() ON UPDATE
current_timestamp(),
  `created_at` timestamp NOT NULL DEFAULT
current_timestamp(), `pulse_count`
int(11) DEFAULT 0

```

```

) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
-- Dumping data for table `meters`
INSERT INTO `meters` (`meter_id`, `user_id`,
`meter_number`, `type`, `location`, `current_units`,
`total_consumption`, `last_reading`, `status`,
`last_update`, `created_at`, `pulse_count`) VALUES
(11, NULL, 'MAIN001', 'main', NULL, 5.00, 0.00,
0.000000, 'active', '2025-07-01 22:15:53',
'2025-06-18 23:36:10', 0),
(17, 17, 'MTR002', 'sub', 'ROOM90', 5.00, 0.00,
NULL, 'active', '2025-07-01 22:11:18', '2025-0701
22:11:18', 0);

-- Table structure for table `meter_readings`
CREATE TABLE `meter_readings` (
`reading_id` int(11) NOT NULL,
`meter_id` int(11) DEFAULT NULL,
`pulses` int(11) NOT NULL,
`energy_consumption` decimal(10,6) NOT
NULL,
`reading_time` timestamp NOT NULL DEFAULT
current_timestamp(),
`timestamp` datetime DEFAULT
current_timestamp()
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
-- Table structure for table `password_resets`
CREATE TABLE `password_resets` (
`id` int(11) NOT NULL,
`user_id` int(11) NOT NULL,
`token` varchar(255) NOT NULL,
`expires_at` timestamp NOT NULL DEFAULT current_timestamp()
ON UPDATE
current_timestamp(),
`created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

```

```

-- Table structure for table `settings`
CREATE TABLE `settings` (
  `setting_id` int(11) NOT NULL,
  `setting_key` varchar(50) NOT NULL,
  `setting_value` text DEFAULT NULL,
  `setting_description` text DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT
current_timestamp(),
  `updated_at` timestamp NOT NULL
DEFAULT current_timestamp() ON UPDATE
current_timestamp()
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
-- Dumping data for table `settings`
INSERT INTO `settings` (`setting_id`, `setting_key`,
`setting_value`,
`setting_description`, `created_at`, `updated_at`)
VALUES
(1, 'system_name', 'Energy Meter Management
System', 'Name of the system', '2025-06-18
19:39:48', '2025-06-18 19:39:48'),
(2, 'company_name', 'Energy Meter System',
'Company name', '2025-06-18 19:39:48', '2025-
06-18 19:39:48'),
(3, 'currency', 'USD', 'Currency for transactions', '2025-06-18
19:39:48', '2025-06-18 19:39:48'),
(4, 'unit_price', '0.15', 'Price per unit of energy',
'2025-06-18 19:39:48', '2025-06-18 19:39:48'),
(5, 'maintenance_mode', 'false', 'System
maintenance mode', '2025-06-18 19:39:48',
'2025-06-18 19:39:48'),
(6,
'api_key',
'f8407d45c145ede1c2fla1c9367cd0dd', 'API key
for external access', '2025-06-18 19:39:48',
'2025-06-18 19:39:48'),
(7, 'notification_email', 'admin@example.com', 'Email
for system notifications', '2025-06-18
19:39:48', '2025-06-18 19:39:48'),
(8, 'timezone', 'UTC', 'System timezone', '2025-
06-18 19:39:49', '2025-06-18 19:39:49'),

```

```

(9, 'date_format', 'Y-m-d H:i:s', 'Date format for
display', '2025-06-18 19:39:49', '2025-06-18
19:39:49'),
(10, 'units_per_pulse', '0.001', 'Energy units per
pulse', '2025-06-18 19:39:49', '2025-06-18
19:39:49'),
(11, 'last_data_update', '2025-07-02 01:11:19',
'Last time data was updated', '2025-06-18
20:54:09', '2025-07-01 22:11:19'), (13,
'token_expiry_hours', '24', 'Number of hours
before a generated token expires', '2025-06-23
09:43:12', '2025-06-23 09:43:12'),
(14, 'minimum_purchase_units', '1', 'Minimum
number of units that can be purchased', '2025-
0623 09:43:12', '2025-06-23 09:43:12'),
(17, 'contact_email', 'support@example.com', 'Support
contact email', '2025-06-23 09:43:14',
'2025-06-23 09:43:14');

```

```

-- Table structure for table `submeter_readings`
CREATE TABLE `submeter_readings` (
  `id` int(11) NOT NULL,
  `main_meter_id` int(11) DEFAULT NULL,
  `submeter_id` int(11) DEFAULT NULL,
  `energy_consumption` decimal(10,6) NOT
NULL,
  `timestamp` timestamp NOT NULL DEFAULT
current_timestamp(), `is_updated`
tinyint(1) DEFAULT 0
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4

```

```

COLLATE=utf8mb4_general_ci;
-- Table structure for table `tokens`
CREATE TABLE `tokens` (
  `token_id` int(11) NOT NULL,
  `token` varchar(50) NOT NULL,
  `user_id` int(11) DEFAULT NULL,
  `meter_id` int(11) DEFAULT NULL,
  `units` decimal(10,2) NOT NULL,

```

```

`amount` decimal(10,2) NOT NULL,
`status`      enum('pending','used','expired')  DEFAULT
'pending',
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`expires_at` timestamp NULL DEFAULT NULL
)          ENGINE=InnoDB          DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;
-- Dumping data for table `tokens`
--

```

```

INSERT INTO `tokens` (`token_id`, `token`,
`user_id`, `meter_id`, `units`, `amount`, `status`,
`created_at`, `expires_at`) VALUES
(4, 'R0011000100007757371', 4, NULL, 10.00,
1000.00, "", '2025-06-18 19:45:43', '2025-06-19
19:45:43'),
(5, 'R0022000200089505247', 4, NULL, 20.00,
2000.00, "", '2025-06-18 19:51:54', '2025-06-19
19:51:54'),
(6, 'R0011000100047468891', 4, NULL, 10.00,
1000.00, "", '2025-06-18 20:51:41', '2025-06-19
20:51:41'),
(7, 'R0011000100078154647', 4, NULL, 10.00,
1000.00, "", '2025-06-18 20:59:24', '2025-06-19
20:59:24'),
(8, 'R0011000100075514666', 4, NULL, 10.00,
1000.00, "", '2025-06-18 21:11:23', '2025-06-19
21:11:23'),
(9, 'R0011000100013581105', 4, NULL, 10.00,
1000.00, "", '2025-06-18 21:20:31', '2025-06-19
21:20:31'),
(10, 'R0032000200048445417', 4, NULL, 20.00,
2000.00, "", '2025-06-18 21:51:55', '2025-06-19
21:51:55'),
(11, 'R0031200120079507623', 4, NULL, 12.00,
1200.00, "", '2025-06-18 22:29:33', '2025-06-19
22:29:33'),
(12, 'R004100001000093519393', 4, NULL,
100.00, 10000.00, "", '2025-06-18 22:31:49',
'2025-06-19 22:31:49'),

```

(13, 'R0031000100082615053', 4, NULL, 10.00,
1000.00, ", '2025-06-18 22:32:52', '2025-06-19
22:32:52'),
(14, 'R0041400140064018703', 4, NULL, 14.00,
1400.00, ", '2025-06-19 09:20:38', '2025-06-20
09:20:38'),
(15, 'R0040100010007910367', 4, NULL, 1.00, 100.00, ",
'2025-06-19 11:23:23', '2025-06-20
11:23:23'),
(16, 'R0041000100069662360', 4, NULL, 10.00,
1000.00, ", '2025-06-19 11:24:51', '2025-06-20
11:24:51'),
(17, 'R0040100010040763402', 4, NULL, 1.00,
100.00, ", '2025-06-19 11:26:07', '2025-06-20
11:26:07'),
(18, 'R0041000100001859465', 4, NULL, 10.00,
1000.00, ", '2025-06-19 16:17:18', '2025-06-20
16:17:18'),
(19, 'R0021000100030133764', 4, NULL, 10.00,
1000.00, ", '2025-06-23 09:54:31', '2025-06-24
09:54:31'),
(20, 'R0050100010066219509', 4, NULL, 1.00,
100.00, ", '2025-06-24 10:00:46', '2025-06-25
10:00:46'),
(21, 'R0011000100027393994', 4, NULL, 10.00,
1000.00, 'pending', '2025-06-28 11:42:42', '2025-
06-29 11:42:42'),
(22, 'R0051000100061203198', 4, NULL, 10.00,
1000.00, ", '2025-06-28 11:43:17', '2025-06-29 11:43:17'),
(23, 'R0031000100070891703', 4, NULL, 10.00,
1000.00, ", '2025-06-28 12:14:34', '2025-06-29
12:14:34'),
(24, 'R0010100010061868342', 4, NULL, 1.00,
100.00, ", '2025-07-01 22:08:56', '2025-07-02
22:08:56'),
(25, 'R0010001000153694753', 4, NULL, 0.01, 1.00, ",
'2025-07-01 22:15:15', '2025-07-02
22:15:15');

--

```

-- Table structure for table `transactions`
--
CREATE TABLE `transactions` (
  `transaction_id` int(11) NOT NULL,
  `meter_id` int(11) DEFAULT NULL,
  `user_id` int(11) DEFAULT NULL,
  `token_id` int(11) DEFAULT NULL, `transaction_type`
enum('purchase','consumption') NOT NULL,
  `units` decimal(10,2) NOT NULL,
  `amount` decimal(10,2) NOT NULL,
  `status` enum('pending','completed','failed')
DEFAULT 'pending',
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

```

```

--
-- Dumping data for table `transactions`
--
INSERT INTO `transactions` (`transaction_id`, `meter_id`,
`user_id`, `token_id`,
`transaction_type`, `units`, `amount`, `status`, `created_at`)
VALUES
(2519, 17, 17, NULL, "", 5.00, 0.00, 'completed',
'2025-07-01 22:11:18');

```

```

-- Table structure for table `users`
--
CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(100) NOT NULL,
  `name` varchar(100) NOT NULL,
  `meter_number` varchar(50) DEFAULT
NULL,
  `role` enum('admin','user') NOT NULL
DEFAULT 'user',
  `status` enum('active','inactive','suspended')
DEFAULT 'active',

```

```

`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4
COLLATE=utf8mb4_general_ci;

```

```

--
-- Dumping data for table `users`
--
INSERT INTO `users` (`user_id`, `username`,
`password`, `email`, `name`, `meter_number`,
`role`, `status`, `created_at`, `updated_at`)
VALUES
(4, 'doctor',
'$2y$10$Yn.R9.C93P2cIquEB6fgH.j.nWowz/y5
TrZYOJYgP4eCDI/KeVLCa',
'hosea@gmail.com', 'HOSEA', 'M20256126',
'admin', 'active', '2025-06-18 19:42:28', '2025-06-
18 19:42:28'),
(17, 'hosea12@gmail.com',
'$2y$10$.VaVbNcd469UTBECu0V08eNpKm3t
eRRCeD8/W2eyDavUofMwjcWv.',
'hosea12@gmail.com', 'kimaro2', 'MTR002',
'user', 'active', '2025-07-01 22:11:18', '2025-07-
01 22:11:18');

```

```

--
-- Indexes for dumped tables

```

```

--
-- Indexes for table `alerts`

```

```

--
ALTER TABLE `alerts`
ADD PRIMARY KEY (`alert_id`),
ADD KEY `fk_alerts_resolved_by`
(`resolved_by`),
ADD KEY `idx_alerts_meter_id` (`meter_id`),
ADD KEY `idx_alerts_status` (`status`),

```

```
ADD KEY `idx_alerts_created_at`  
(`created_at`);
```

```
--  
-- Indexes for table `audit_logs`
```

```
--  
ALTER TABLE `audit_logs`  
ADD PRIMARY KEY (`log_id`),  
ADD KEY `user_id` (`user_id`);
```

```
--  
-- Indexes for table `invoices`
```

```
--  
ALTER TABLE `invoices`  
ADD PRIMARY KEY (`invoice_id`),  
ADD KEY `user_id` (`user_id`),  
ADD KEY `meter_id` (`meter_id`);
```

```
--  
-- Indexes for table `meters`
```

```
--  
ALTER TABLE `meters`  
ADD PRIMARY KEY (`meter_id`),  
ADD UNIQUE KEY `meter_number`  
(`meter_number`),  
ADD KEY `user_id` (`user_id`);
```

```
--  
-- Indexes for table `meter_readings`
```

```
--  
ALTER TABLE `meter_readings`  
ADD PRIMARY KEY (`reading_id`),  
ADD KEY `meter_id` (`meter_id`);
```

```
--  
-- Indexes for table `password_resets`
```

```
--  
ALTER TABLE `password_resets`
```

```

ADD PRIMARY KEY (`id`),
ADD KEY `user_id` (`user_id`);

--
-- Indexes for table `settings`
--
ALTER TABLE `settings`
  ADD PRIMARY KEY (`setting_id`),
  ADD UNIQUE KEY `setting_key`
  (`setting_key`);

--
-- Indexes for table `submeter_readings`
--
ALTER TABLE `submeter_readings`
  ADD PRIMARY KEY (`id`),
  ADD KEY `main_meter_id`
  (`main_meter_id`),
  ADD KEY `submeter_id` (`submeter_id`);

--
-- Indexes for table `tokens`
--
ALTER TABLE `tokens`
  ADD PRIMARY KEY (`token_id`),
  ADD UNIQUE KEY `token` (`token`),
  ADD KEY `user_id` (`user_id`),
  ADD KEY `meter_id` (`meter_id`);

--
-- Indexes for table `transactions`
--
ALTER TABLE `transactions`
  ADD PRIMARY KEY (`transaction_id`),
  ADD KEY `meter_id` (`meter_id`),
  ADD KEY `user_id` (`user_id`),
  ADD KEY `token_id` (`token_id`);

--
-- Indexes for table `users`
--

```

```

ALTER TABLE `users`
  ADD PRIMARY KEY (`user_id`),
  ADD UNIQUE KEY `username` (`username`),
  ADD UNIQUE KEY `email` (`email`),
  ADD UNIQUE KEY `meter_number`
(`meter_number`);
--
-- AUTO_INCREMENT for dumped tables
--
--
-- AUTO_INCREMENT for table `alerts`
--
ALTER TABLE `alerts`
  MODIFY `alert_id` int(11) NOT NULL
  AUTO_INCREMENT,
  AUTO_INCREMENT=9;
--
-- AUTO_INCREMENT for table `audit_logs`
--
ALTER TABLE `audit_logs`
  MODIFY `log_id` int(11) NOT NULL
  AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `invoices`
--
ALTER TABLE `invoices`
  MODIFY `invoice_id` int(11) NOT NULL
  AUTO_INCREMENT;
--
-- AUTO_INCREMENT for table `meters`
--
ALTER TABLE `meters`
  MODIFY `meter_id` int(11) NOT NULL
  AUTO_INCREMENT,
  AUTO_INCREMENT=18;
--
-- AUTO_INCREMENT for table
`meter_readings`
--
ALTER TABLE `meter_readings`
  MODIFY `reading_id` int(11) NOT NULL

```

```

AUTO_INCREMENT,
AUTO_INCREMENT=14;
--
--      AUTO_INCREMENT      for      table
`password_resets`
--
ALTER TABLE `password_resets`
  MODIFY `id` int(11) NOT NULL
  AUTO_INCREMENT;

--
-- AUTO_INCREMENT for table `settings`
--
ALTER TABLE `settings`
  MODIFY `setting_id` int(11) NOT NULL
  AUTO_INCREMENT,
  AUTO_INCREMENT=54;

--
--      AUTO_INCREMENT      for      table
`submeter_readings`
--
ALTER TABLE `submeter_readings`
  MODIFY `id` int(11) NOT NULL
  AUTO_INCREMENT,
  AUTO_INCREMENT=14;

--
-- AUTO_INCREMENT for table `tokens`
--
ALTER TABLE `tokens`
  MODIFY `token_id` int(11) NOT NULL
  AUTO_INCREMENT,
  AUTO_INCREMENT=26;

--
-- AUTO_INCREMENT for table `transactions`
--
ALTER TABLE `transactions`
  MODIFY `transaction_id` int(11) NOT NULL
  AUTO_INCREMENT,
  AUTO_INCREMENT=2521;

--
-- AUTO_INCREMENT for table `users`

```

```

--
ALTER TABLE `users`
  MODIFY `user_id` int(11) NOT NULL
  AUTO_INCREMENT,
  AUTO_INCREMENT=18;

-- Constraints for table `alerts`
--
ALTER TABLE `alerts`
  ADD CONSTRAINT `fk_alerts_meter_id` FOREIGN
  KEY (`meter_id`) REFERENCES
  `meters` (`meter_id`) ON DELETE CASCADE,
  ADD CONSTRAINT `fk_alerts_resolved_by` FOREIGN
  KEY (`resolved_by`) REFERENCES
  `users` (`user_id`) ON DELETE SET NULL;
--
-- Constraints for table `audit_logs`
--
ALTER TABLE `audit_logs`
  ADD CONSTRAINT `audit_logs_ibfk_1`
  FOREIGN KEY (`user_id`) REFERENCES
  `users` (`user_id`);
--
-- Constraints for table `invoices`
--
ALTER TABLE `invoices`
  ADD CONSTRAINT `invoices_ibfk_1` FOREIGN KEY
  (`user_id`) REFERENCES
  `users` (`user_id`),
  ADD CONSTRAINT `invoices_ibfk_2` FOREIGN KEY
  (`meter_id`) REFERENCES
  `meters` (`meter_id`);

--
-- Constraints for table `meters`
--
ALTER TABLE `meters`
  ADD CONSTRAINT `meters_ibfk_1` FOREIGN KEY
  (`user_id`) REFERENCES
  `users` (`user_id`) ON DELETE SET NULL;

```

```

--
-- Constraints for table `meter_readings`
--
ALTER TABLE `meter_readings`
  ADD CONSTRAINT `meter_readings_ibfk_1` FOREIGN
  KEY (`meter_id`) REFERENCES
  `meters` (`meter_id`) ON DELETE CASCADE;

-- Constraints for table `password_resets`

ALTER TABLE `password_resets`
  ADD
  CONSTRAINT
  `password_resets_ibfk_1` FOREIGN KEY
  (`user_id`) REFERENCES `users` (`user_id`);

-- Constraints for table `submeter_readings`

ALTER TABLE `submeter_readings`
  ADD
  CONSTRAINT
  `submeter_readings_ibfk_1` FOREIGN KEY
  (`main_meter_id`) REFERENCES `meters`
  (`meter_id`) ON DELETE CASCADE,
  ADD
  CONSTRAINT
  `submeter_readings_ibfk_2` FOREIGN KEY (`submeter_id`)
  REFERENCES `meters`
  (`meter_id`) ON DELETE CASCADE;
-- Constraints for table `tokens`
ALTER TABLE `tokens`
  ADD CONSTRAINT `tokens_ibfk_1`
  FOREIGN KEY (`user_id`) REFERENCES
  `users` (`user_id`) ON DELETE SET NULL,
  ADD CONSTRAINT `tokens_ibfk_2` FOREIGN KEY
  (`meter_id`) REFERENCES
  `meters` (`meter_id`) ON DELETE SET NULL;

-- Constraints for table `transactions`
ALTER TABLE `transactions`
  ADD CONSTRAINT `transactions_ibfk_1` FOREIGN
  KEY (`meter_id`) REFERENCES

```

```
`meters` (`meter_id`) ON DELETE CASCADE,    ADD CONSTRAINT `transactions_ibfk_2`  
FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`) ON DELETE SET NULL,  
    ADD CONSTRAINT `transactions_ibfk_3` FOREIGN  
KEY (`token_id`) REFERENCES  
`tokens` (`token_id`) ON DELETE SET NULL;  
COMMIT;
```

```
/*!40101                SET  
CHARACTER_SET_CLIENT=@OLD_CHAR  
ACTER_SET_CLIENT */;
```

```
/*!40101                SET  
CHARACTER_SET_RESULTS=@OLD_CHA  
RACTER_SET_RESULTS */;
```

```
/*!40101                SET  
COLLATION_CONNECTION=@OLD_COLL  
ATION_CONNECTION */
```