Final Year Project Report

Project Title (Smart Energy Meter with IoT Connectivity)

Project Advisor: Muhammad Rehan Saleem

Submitted By:

Usman Khalid	**(15026020282)**
Uzair Nawaz	**(15026020283)**
Aqeel Alam	**(150026020285)**
Mohsin javaid	**(14024020072)**

Session

(2015-2019)

University of Management and Technology

C-II Johar Town Lahore Pakistan

Dedication

This project is dedicated to all the people who are working tirelessly to bring the fruits of technology to our county. This includes the teachers in universities and other academic institutes, the researchers who implement the latest technology to solve everyday problems and the students who work their best to learn the state f the art in their respective fields.

FINAL APPROVAL

- **Head of Department** ________________________
  Department of Computer Science
  School of Systems and Technology
  UMT Lahore

- **Director Final Year Projects**
- Department of Computer Science ________________________
  School of Systems and Technology
  UMT Lahore

- **Supervisor** ________________________
  School of Systems and Technology
  UMT Lahore

- **Co-Supervisor (if any)** ________________________

Acknowledgment

We are very thankful to our ultimate Lord and benefactor, ALLAH for his countless blessings that have always showered upon us. We find Him to be our strength in the times of weakness and our pride in the times of achievement.

We are highly grateful to our FYP Supervisor and the Lecturer in School of Systems and Technology, UMT, Lahore, Sir Muhammad Rehan Saleem Without his guidance and motivation, this feat would never have been possible. With his keen interest in this subject and timely guidance, we have not only been able to complete the research but also write the thesis with the highest level of motivation.

The most of my thanks go to our parents who have shaped us to be the persons we are today. Their prayers and support has never let us down in any stage of our lives.

Project Title   Smart Energy Meter with IoT Connectivity

Undertaken by :

1. **Usman Khalid (15026020282)**
2. **Uzair Nawaz (15026020283)**
3. **Aqeel Alam (150026020285)**
4. **Mohsin javaid (14024020072)**

Supervised by: Syed Mohsin Nizami

Starting Date: 3rd March 2019

Completion Date: 9th January 2020

Tools Used :

Current transformer module

Voltage transformer module

Power adapter

5 V Buck Converter

Mains terminal blocks

Node MCU Controller

Web Server

Plagairism Report

Supervisor’s Approval

We have carefully examined the documentation of the Final Year Project titled *“Smart Energy Meter with IoT Connectivity* “and I endorse that this documentation complies with the standards of an undergraduate level Final Year Project report.

Moreover, I have also checked for the plagiarism beyond Turnitin the documentation and I am convinced that this documentation has not been plagiarized.

Signature:___________________ Date:____________________

Abstract

This project involves the design and development of a smart energy meter with IoT connectivity. The meter has been designed as a low power prototype for a domestic energy meter. It directly measures the voltage and current of the utility electricity connection and calculates the electrical power and energy being utilized. It then uses an internet connection to send the measurements to an IoT server.

The server aggregates the data and performs analysis on it to derive different usage patterns and to analyze how much energy is being utilized at which part of the day and which day of the week. This analysis can be very useful in helping the user make informed decisions to cut back their electricity consumption.

REVISION CHART

V 0.001	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Meeting with the advisor and talking about the idea	23/05/2019
V 0.010	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Planning	13/6/2019
V 0.020	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Literature Review.	04/07/2019
V 0.030	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Voltage Measurement.	18/07/2019
V 0.040	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Current Measurement.	29/07/2019
V 0.050	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Calculation of power from measured values.	15/08/2019
V 0.060	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Development of Web server	12/09/2019
V 0.070	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Sending data to internet	12/10/2019
V 0.090	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Saving data on web server	04/11/2019

V 0.1	Usman Khalid Uzair Nawaz Aqeel Alam Mohsin javaid	Final testing and troubleshooting.	27/11/2019

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

1.1 Motivations

Most of the energy conservation efforts go towards the usage of smart and efficient devices and the installation of renewable energy sources like solar, wind, hydro and geothermal power generation.

These methods are very useful in reducing the energy footprint of a nation and can even help individuals to contribute to the overall reduction of energy use and carbon emissions. However, at individual level, there is a strong need for information about energy demand and usage trends that can help an individual make informed decisions about cutting back their energy usage about a particular time or a day.

Such information can help individuals and households to know better about their energy demand across various periods. By observing their energy usage trend with time of day and day of week, it is anticipated that they may be able to identify some key times when they can easily cut back their overall energy usage by a significant amount.

1.2 Project Overview

In this project, we propose to develop a model of such a system where a smart meter measures the voltage and current being consumed by the user and use it to calculate the total power being consumed, as well as the total energy consumed by the user until now. The consumed energy is recorded at regular intervals and analysis are performed on the trends of energy usage during every hour, day and month.

1.3 Problem Statement

Design and develop a smart energy meter that measures the voltage and current being consumed by a household and uses the measured values to calculate the power being consumed by the household. The meter must send the latest measurements along with a time tag to a dedicated web server through an internet connection.

The web server must keep the record of all data sent by the meter and categorize it on hourly, daily and weekly basis. A user must be able to view the data in any of the decided formats and observe their energy usage trends from that data.

1.4 Objectives

(Tentative)

With this project, we tentatively plan you achieve the following objectives.

- Measure the line voltage and current in a domestic electric circuit.
- Calculate the current real power consumption by the complete household.
- Keep record of the total energy used by the household over the last hour, day and month.
- Send all the collected and calculated data to a cloud-based server.
- Enable a user to view the energy and current power consumption records over their Smartphone.
- Design and implementation of hardware.
- Writing software for the sensors measurements and the device to send data to the server
- Writing software for server to accept and record data
- Using a python based cloud computing services to build a reliable and scalable system

2. DOMAIN ANALYSIS

2.1 Customer

This project can be implemented privately or any public place like:

- House holds
- Shops
- Schools
- Offices

After successful small scale testing and deployment it can also be implemented in larger establishments such as the following

- Factories
- Shopping malls
- Restaurants
- Universities

2.2 Stakeholders

Stakeholder	***Role in System***
Developer	*Developer will maintain all the backend things like developing the application, maintaining sever etc.*
User	*The user can view their electricity usage trends in time slots of hours, days, weeks and months. They can make important decisions to cut back their electricity usage at important times using this data.*

Table 1: List of stakeholders

2.3 Affected Groups with social or economic impact

After successful testing and deployment, this project will affect the following social groups.

- Home owners
- Shopkeepers
- Educational institutes

This project has the potential to bring the following social and economic benefits.

- Observe one's energy footprint
- Reduce energy consumption
- Cut back electricity consumption at strategic times
- Reduce the peak stress on the electricity distribution network

2.4 Dependencies/ External Systems

This project has the following dependencies.

- Electricity supply
- Internet connectivity
- Server space
- User must have a web browser

3. REQUIREMENTS ANALYSIS

3.1 Requirements

We will need following things to make this system.

- Node MCU
- ZMCT130C current sensor
- ZMPT101B voltage sensor
- Power adapter
- Mains terminal blocks
- Python based server (native HTTP server libraries)

REF#	***FUNCTIONS***	***CATEGORY***	***ATTRIBUTE***	***DETAILS***
R1	*To measure mains voltage*	*FUNCTIONAL*	*Measure voltage*	*The device measures the voltage of the mains supply at regular intervals*
R2	*To measure mains current*	*FUNCTIONAL*	*Measure current*	*The device measures the current being consumed by the household at regular intervals*
R3	*To calculate power*	*FUNCTIONAL*	*Calculate power*	*The meter calculates the real time electric power being consumed using the measured voltage and current.*

R4	*To calculate energy*	*FUNCTIONAL*	*Calculate energy*	*The device calculates the total electrical energy consumed by the user since the last update to the server*
R5	*To send data to the server*	*FUNCTIONAL*	*Send data to the server*	*The smart meter sends the calculated and measured data to the server at regular intervals.*
R6	*To record data*	*FUNCTIONAL*	*Record data*	*The server accepts and records the data received from the meter, along with a time stamp.*
R7	*To view hourly data*	*FUNCTIONAL*	*View hourly data*	*The user can view the hourly record of electricity consumption*
R8	*To view daily data*	*FUNCTIONAL*	*View daily data*	*The user can view the daily record of electricity consumption*
R9	*To view monthly data*	*FUNCTIONAL*	*View monthly data*	*The user can view the monthly record of electricity consumption*

Table 2: List of Requirements

REF#	ATTRIBUTE	DETAILS	CATEGORY
R1	SECURITY	USER CREDENTIALS MUST BE SECURE	MENDATORY
R2	PRIVACY	HIDE THE INFORMATION	MENDATORY
R3	AVAILABILITY	USER MUST BE ONLINE	MENDATORY
R4	USABILITY	MUST BE EFFICIENT	OPTIONAL
R5	CONSISTENCY	APPLICATION SHOULD HAVE STABILITY	MENDATORY
R6	RELIABILITY	APP SHOULD HAVE RELIABLE	MENDATORY

Table 3: List of Attributes

3.2 List of Actors

- Smart meter
- User

3.3 List of use cases

This smart meter is an intelligent device that performs several real time measurements and calculations. the following use cases are developed and implemented for the user and the smart meter.

- Case 1 (measure mains voltage): The meter measures the AC voltage on the mains line at regular intervals.
- Case 2 (measure mains current): The meter measures the AC current flowing in the mains line at regular intervals.
- Case 3 (calculate instantaneous power): The meter calculates the instantaneous electrical power being consumed by the household.
- Case 4 (view current power consumption): The user views the current electrical power consumption on the server home page.
- Case 5 (view daily power consumption): The user views the hourly electrical power consumption on the server for the last 24 hours.
- Case 6 (view weekly power consumption): The user views the daily electrical power consumption on the server for the last 7 days.
- Case 7 (view monthly power consumption): The user views the daily electrical power consumption on the server for the last 30 days.

3.4 System use case diagram

A sample use-case diagram is shown in the following diagram. It shows the possible action that the user of the system can take and the resulting reaction to each of them.

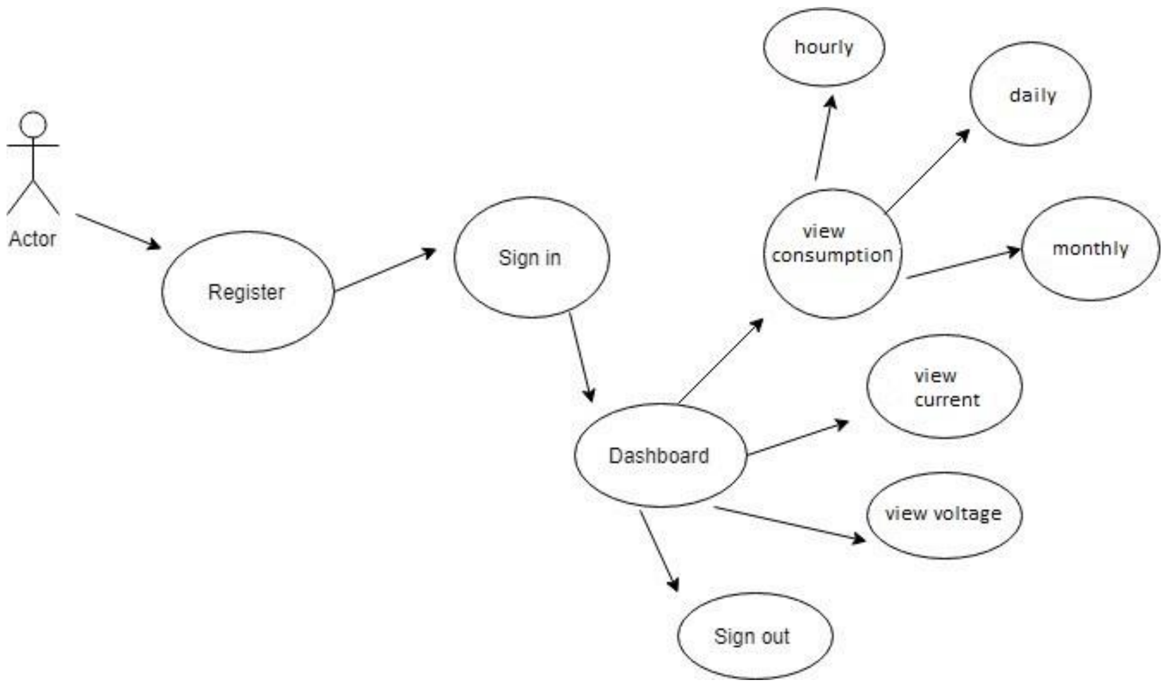

Figure 1: A sample use-case diagram

The use cases are described in the following section.

3.5 Use Cases

Use Case ID:	Case 1		
Use Case Name:	Measure mains voltage		
Created By:		**Last Updated By:**	
Date Created:	01-Jul-2019	**Last Revision Date:**	
Actors:	Smart meter		
Description:	The meter measures the AC voltage on the mains line at regular intervals.		
Trigger:	System timer		
Preconditions:	Mains connection is available and system is powered up.		
Post conditions:	Measure current and power.		
Normal Flow:	The meter measures the power consumption and submits to the server.		
Alternative Flows:	Power is not measured.		
Exceptions:	Must be calibrated previously to make accurate measurements.		
Includes:	None		
Frequency of Use:	Regular intervals		
Special Requirements:	None		
Assumptions:	None		
Notes and Issues:	None		

The meter measures the AC voltage on the mains line at regular intervals

Use Case ID:	Case 2		
Use Case Name:	Measure mains current		
Created By:		**Last Updated By:**	
Date Created:	14-Jul-2019	**Last Revision Date:**	
Actors:	Smart meter		
Description:	The meter measures the AC current flowing in the mains line at regular intervals.		
Trigger:	System timer		
Preconditions:	Mains connection is available and system is powered up.		
Post conditions:	Calculate power.		
Normal Flow:	The meter measures the power consumption and submits to the server.		
Alternative Flows:	Power is not measured.		
Exceptions:	Must be calibrated previously to make accurate measurements.		
Includes:	None		
Frequency of Use:	Regular intervals		
Special Requirements:	None		
Assumptions:	None		
Notes and Issues:	None		

The meter measures the AC current flowing in the mains line at regular intervals.

Use Case ID:	Case 3		
Use Case Name:	Calculate instantaneous power		
Created By:		**Last Updated By:**	
Date Created:	03-Aug-2019	**Last Revision Date:**	

Actors:	Smart meter
Description:	The meter calculates the instantaneous electrical power being consumed by the household.
Trigger:	System timer
Preconditions:	Voltage and current have been measured.
Post conditions:	Submit data to server.
Normal Flow:	The meter measures the power consumption and submits to the server.
Alternative Flows:	Power is not measured.
Exceptions:	Instantaneous power must be within nominal range of the prototype.
Includes:	None
Frequency of Use:	Regular intervals
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

The meter calculates the instantaneous electrical power being consumed by the household.

Use Case ID:	Case 4		
Use Case Name:	View current power consumption		
Created By:		**Last Updated By:**	
Date Created:	20-Aug-2019	**Last Revision Date:**	
Actors:	User		
Description:	The user can view the current power consumption of the household in near real time.		
Trigger:	User opens web page of server		
Preconditions:	Meter has submitted power measurements.		
Post conditions:	None		
Normal Flow:	The user can view their electricity consumption.		
Alternative Flows:	User cannot view their electricity consumption.		
Exceptions:	User must have a web browser and an internet connection		
Includes:	None		
Frequency of Use:	once		
Special Requirements:	None		
Assumptions:	None		
Notes and Issues:	None		

The user can view the current power consumption of the household in near real time.

Use Case ID:	Case 5		
Use Case Name:	View daily power consumption		
Created By:		**Last Updated By:**	
Date Created:	18-Sep-2019	**Last Revision Date:**	

Actors:	User
Description:	The user views the hourly electrical power consumption on the server for the last 24 hours.
Trigger:	User opens daily consumption web page.
Preconditions:	Meter has submitted power measurements.
Post conditions:	None
Normal Flow:	The user can view their electricity consumption.
Alternative Flows:	User cannot view their electricity consumption.
Exceptions:	User must have a web browser and an internet connection
Includes:	None
Frequency of Use:	once
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

The user views the hourly electrical power consumption on the server for the last 24 hours.

Use Case ID:	Case 6		
Use Case Name:	View weekly power consumption		
Created By:		**Last Updated By:**	
Date Created:	10-Oct-2019	**Last Revision Date:**	
Actors:	User		
Description:	The user views the daily electrical power consumption on the server for the last 7 days.		
Trigger:	User opens weekly consumption web page.		
Preconditions:	Meter has submitted power measurements.		
Post conditions:	None		
Normal Flow:	The user can view their electricity consumption.		
Alternative Flows:	User cannot view their electricity consumption.		
Exceptions:	User must have a web browser and an internet connection		
Includes:	None		
Frequency of Use:	Once		
Special Requirements:	None		
Assumptions:	None		
Notes and Issues:	None		

The user views the daily electrical power consumption on the server for the last 7 days.

Use Case ID:	Case 7		
Use Case Name:	View monthly power consumption		
Created By:		**Last Updated By:**	
Date Created:	25-Oct-2019	**Last Revision Date:**	

Actors:	User
Description:	The user views the daily electrical power consumption on the server for the last 30 days.
Trigger:	User opens monthly consumption web page.
Preconditions:	Meter has submitted power measurements.
Post conditions:	None
Normal Flow:	The user can view their electricity consumption.
Alternative Flows:	User cannot view their electricity consumption.
Exceptions:	User must have a web browser and an internet connection
Includes:	None
Frequency of Use:	Once
Special Requirements:	None
Assumptions:	None
Notes and Issues:	None

The user views the daily electrical power consumption on the server for the last 30 days.

4. DATA FLOW DIAGRAM

4.1 Data Flow Diagram Level

The data flow in this project can be categorized into two main parts. First is the submission of calculation and measruement results from the smart meter to the server. And the second is the user checking their electricity consumption in various formats by opening the server web page.

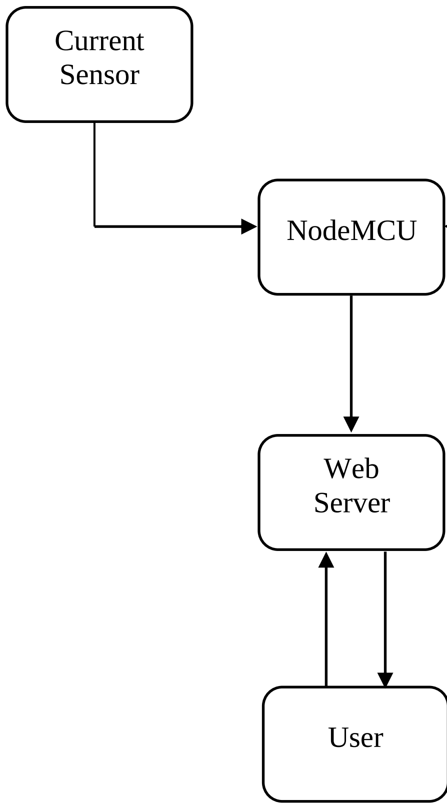

Figure 2: Data flow diagram

This diagram is showing the connectivity of Node MCU with both sides like hardware side with sensors and with the web server. User may interact via web server.

5. SYSTEM DESIGN

5.1 System Architecture Diagram

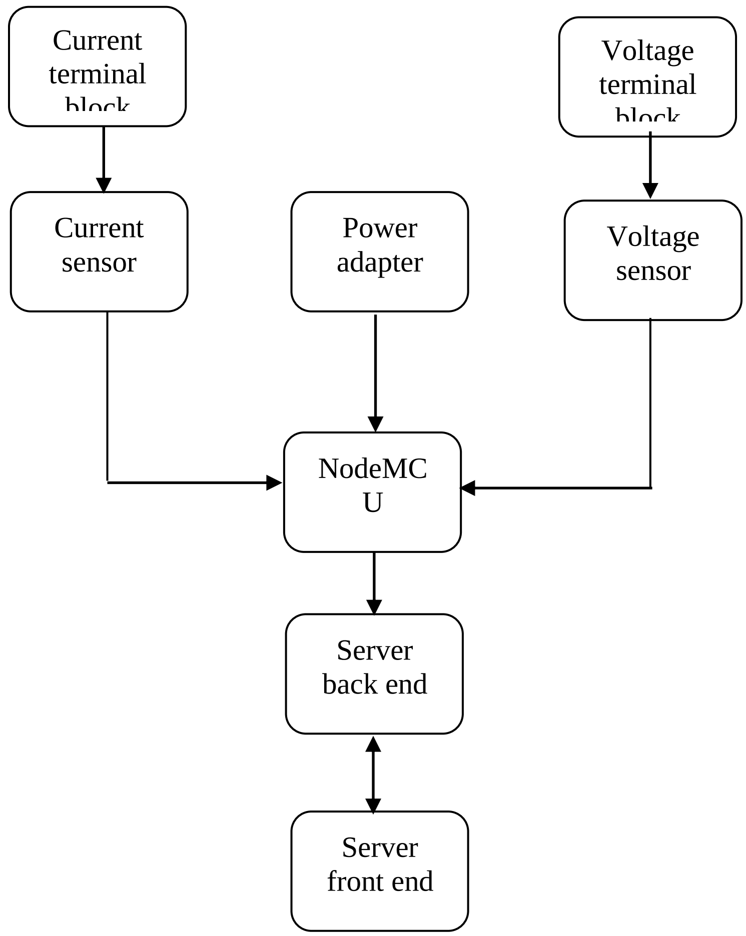

Figure 3: System architecture diagram

This is complete architecture diagram with configuration of sensors, Node MCU server front end and back end. Node MCU requires power, gets value from sensors, and gives signal on server.

5.2 Class Diagram

The following figure shows the class diagram of the smart meter system.

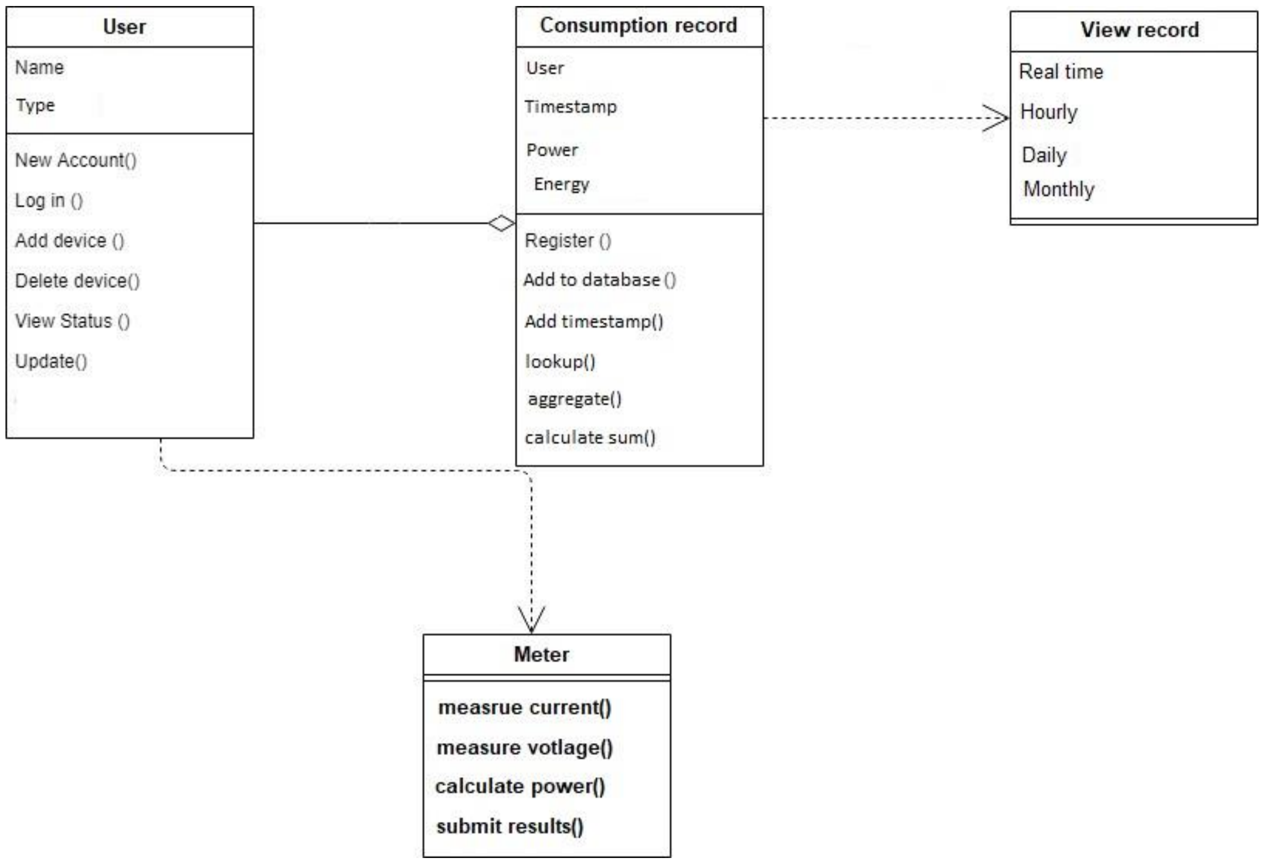

Figure 4: System class diagram

In this diagram User, consumption record, view record and meter functions are shown.

5.3 Sequence Diagrams

The following figure shows the sequence diagram of the data interactions in the smart meter system.

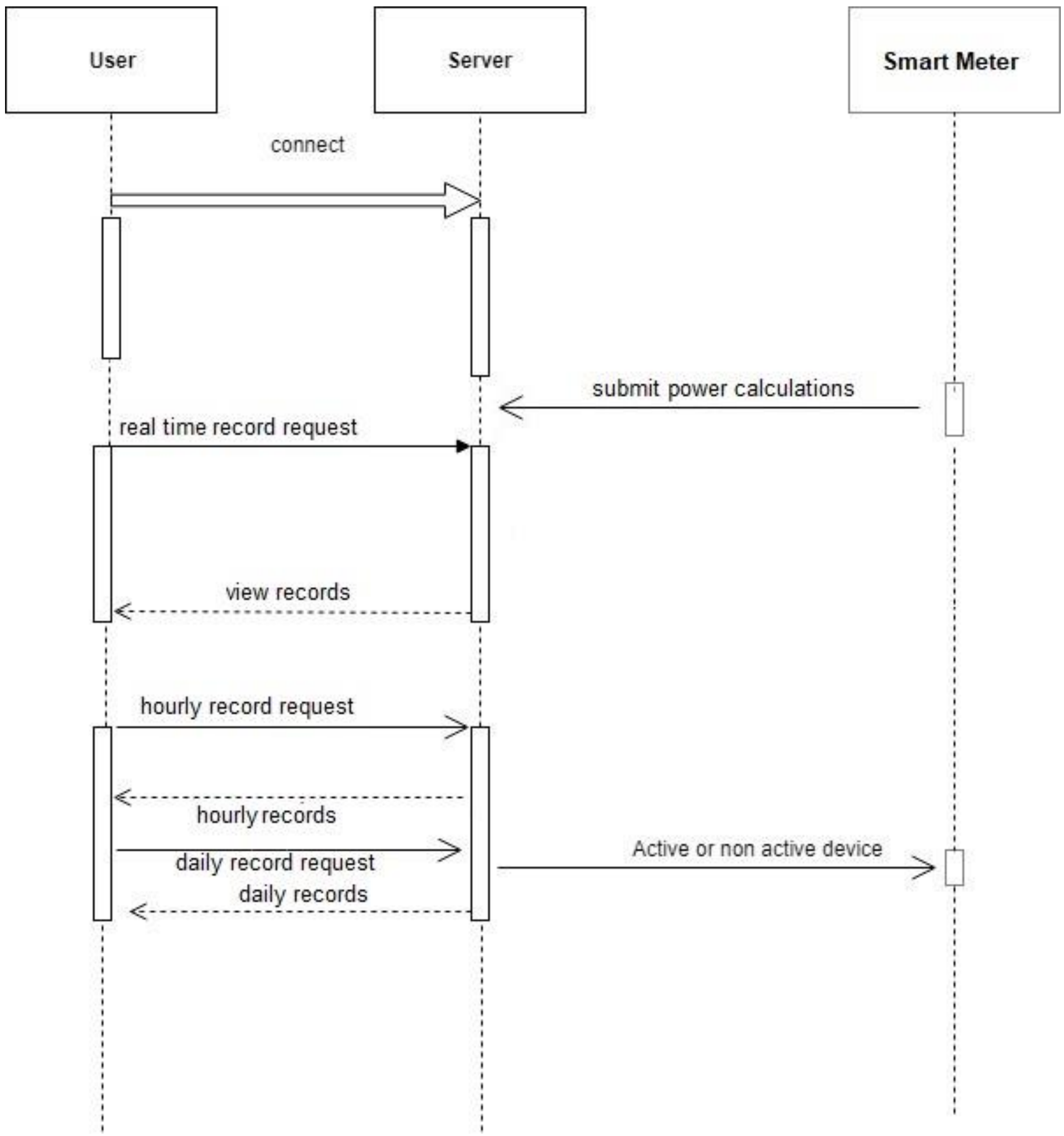

Figure 5: Sequence diagram

5.4 Collaboration Diagrams

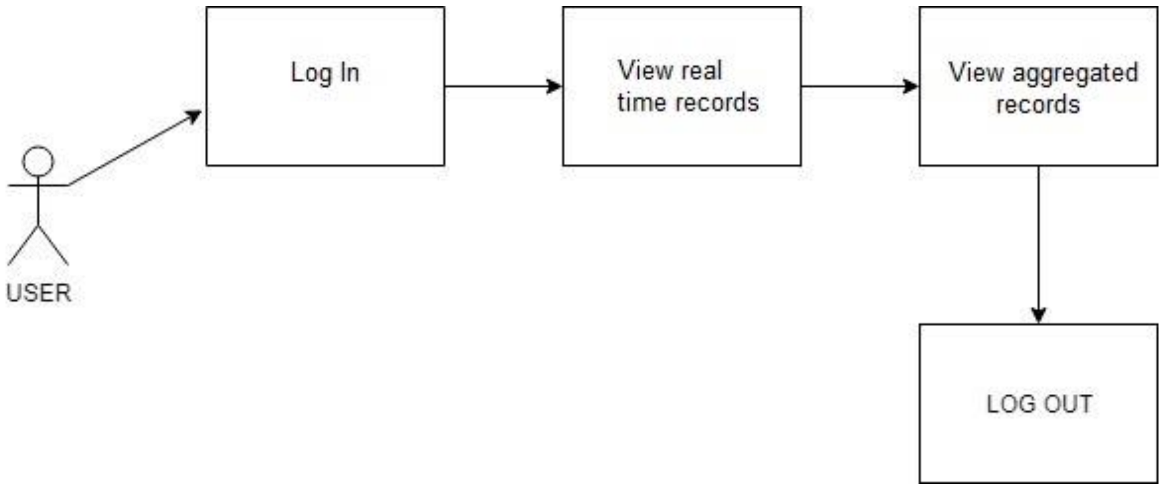

Figure 6: Collaboration diagram

5.5 Data Dictionary

Element Name	Type	Validation	Mandatory	Remarks
current	*float*	*No*	*Yes*	Contains the measured current.
voltage	*float*	*No*	*Yes*	Contains the measured voltage.
power	*float*	*NO*	*Yes*	Contains the instantaneous power calculated from the voltage and current.
timestamp	*Int array*	*No*	*Yes*	The date and time at the instant of calculation the power.
realtime	*int*	*No*	*Yes*	Running time.
ssid	*string*	*No*	*Yes*	For WiFi connectivity
password	*string*	*No*	*Yes*	For connecting to a WiFi network.

Table 4: Data dictionary

6. IMPLEMENTATION DETAILS

6.1 Development Setup

This project involves both hardware and software components, so we used both types of tools and technologies in the process of development and implementation of this project.

In the hardware side, we used the following components in the project. A brief functionality of each component and how it assists in the overall working of the project, is also given.

Node MCU: This is a controller with the capability of connecting to internet via WiFi. It was programmed in the Arduino IDE. We are using its GPIO pins to control the lights and the water pump.

ZMCT130C: This is a current sensor based on a current transformer and an op amp amplifier. It provides the voltage at the output that is proportional to the current passing through the transformer. Due to the design based on a transformer, it can only work with AC current, which is always the case in mains power supply. The controller, using its internal analog to digital converter, measures the output of the current sensor.

ZMPT101B: This is a voltage sensor based on a potential transformer and an op amp amplifier. It provides the voltage at the output, which is proportional to the voltage presented to the transformer. Due to the design based on a transformer, it can only work with AC voltage, which is always the case in mains power supply. The controller, using its internal analog to digital converter, measures the output of the voltage sensor.

Power adapter: As the most of the components in this meter are electronic components, they use low voltage and low current for their power supply requirements. To make the meter self-reliant, the same mains voltage that it is measuring powers all the internal components. For this purpose, a 12 V adapter has been provided in it. This takes the mains voltage as the input and provides the electronic components with a clean DC 12 volts.

5 V Buck Converter: The power to all the components is provided by 12 V, 2 A power supply. The controller and the relay board, however, operate at 5 V. the buck converter converts the 12 V from the power supply to 5 V useful for the low voltage devices. The reason for using buck converter instead of a linear voltage regulator like LM7805 is its better efficiency and less heat generation at the collective current level required by the low voltage components.

In the software side, there are two major parts. One is the controller at the front-end and the other is the server at the back-end. A brief explanation of the programming tools used for each are explained in the following.

Node MCU Controller: The front-end device controller has been programmed in the Arduino IDE. We have used the NodeMCU as a web client that connects to a web server through internet. The internet is accessed via WiFi, by making the NodeMCU connect to an access point that is connected to internet. The controller repeatedly sends the power calculations to the server, which include the current power consumption of the devices and the time stamp.

Web Server: The server is implemented in Python. It has two communication paths, namely, with the smart meter and with the user. It receives the power measurements from the meter and logs them in sorted data files. When it receives the view requests from the user, it aggregates the data from the files and sends the results back to the user via a webpage. The user can view the hourly data for the last 24 hours, daily data for the last week and the daily data for the last month. The data recorded in the data file is just data rows with time stamps. The server performs all the aggregation of the data and formatting for the presentation to the user is performed at runtime.

6.2 Deployment setup

The hardware part was designed and integrated according to the requirements of the project. It is implemented as a model meter that can be connected to the mains power supply of any appliance. For test purposes, we had 4 bulbs acting as a load for the meter. The controller must be provided with an internet connection and the whole set up must be provided with 220 V mains power connection.

The program for the controller was developed in the Arduino IDE using libraries for web client and WiFi connectivity. The same IDE was used to transfer the program to Node MCU using the USB cable.

The program for the server has been implemented on an online Python server. It uses a simple HTTP server library and some file operations to communicate with the controller and the user and store data in files locally. There are two different web pages at the server, one for the controller and the other for the user.

6.3 Algorithms

As for each part of the software, there were to major algorithms running in this project. One was at the server side and the other was at the controller. The controller repeatedly sends the power calculations to the server, which include the current power consumption of the devices and the time stamp. The server receives the power measurements from the meter and logs them in sorted data files. When it receives the view requests from the user, it aggregates the data from the files and sends the results back to the user via a webpage. The user can view the hourly data for the last 24 hours, daily data for the last week and the daily data for the last month. The data recorded in the data file is just data rows with time stamps. The server performs all the aggregation of the data and formatting for the presentation to the user is performed at runtime.

The algorithms for the controller and the server are represented with flow charts in the following.

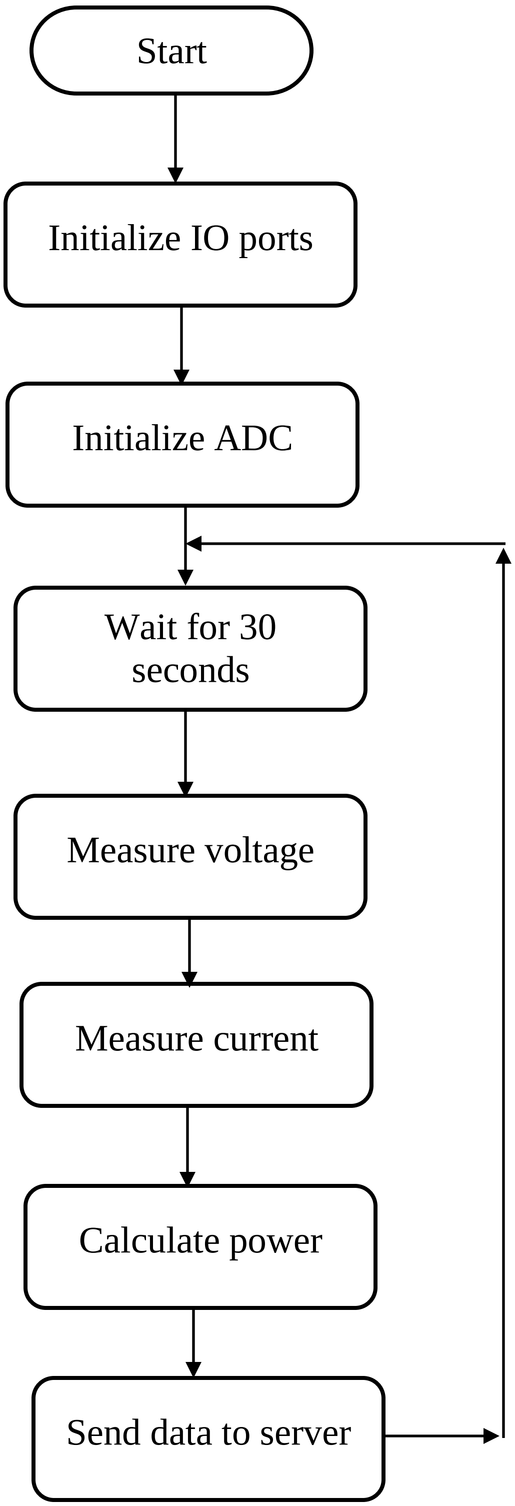

Figure 7: Algorithm for smart meter

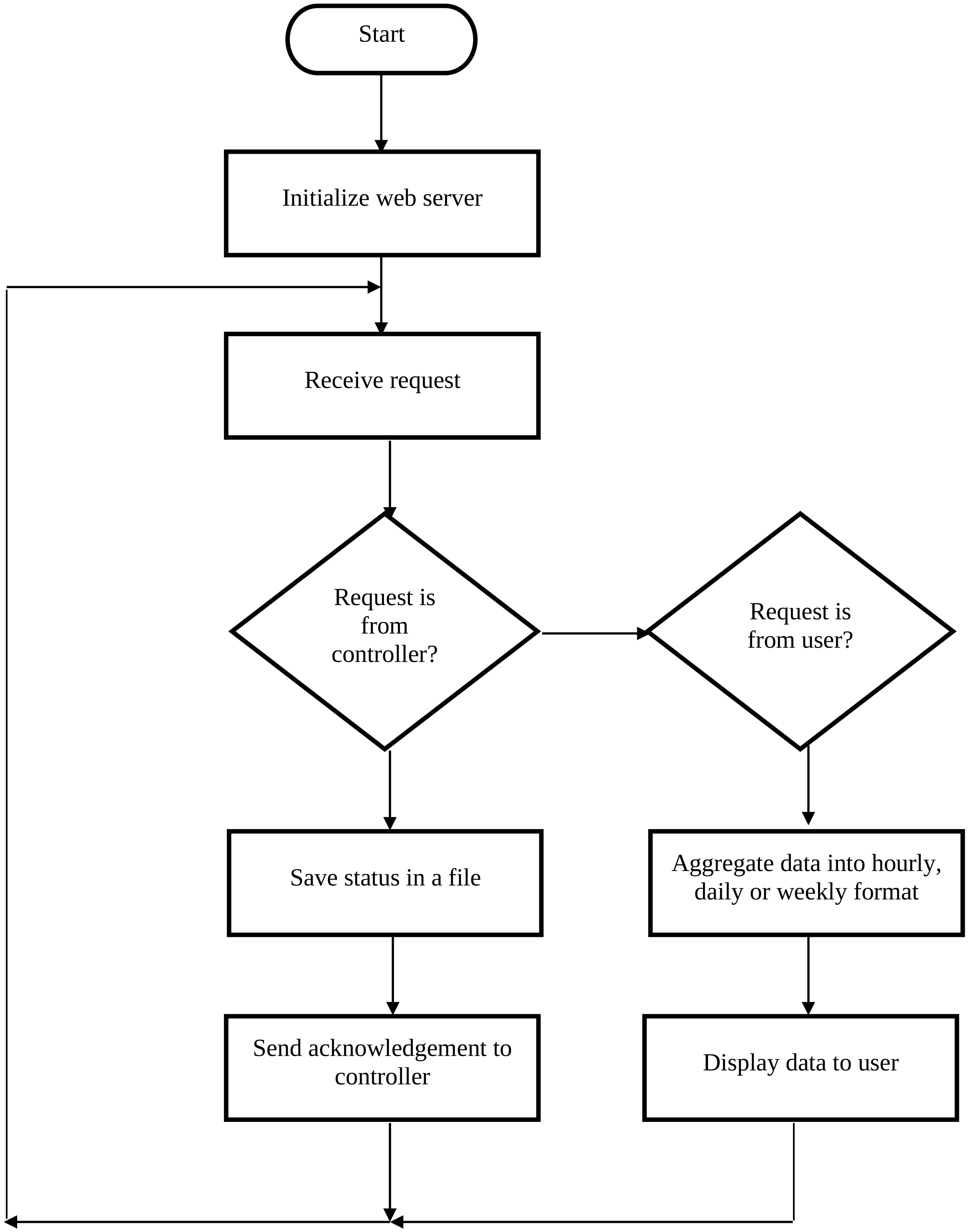

Figure 8: Algorithm for server

6.4 Constraints

The prototype built in the project is a modest sample of the actual product that may be later developed afterwards. So it is meant to operate in a limited way, just to demonstrate the concept.

Although, as a product, the prototype has all the functionalities that were decided as the requirements, it assumes some environmental setup and some constraints that must be observed during its operation. Some of these constraints are related to the facilities this prototype needs to operate correctly and others are related to how the user must interact with the user interface.

The following list provides some of the assumptions and the constraints for this project.

- There must be a working WiFi connection for the controller. The controller is programmed with the name and password of the WiFi network beforehand.
- The WiFi connection must have access to internet.
- There must be a 220 V mains supply for the controller. The controller contains the power supplies for the devices and the low voltage section, but it must be fed with 220 V mains supply via a power cord.
- The server program must be deployed on a Python server connected to internet. We are currently using a very light free server, but larger scale operation may need a dedicated server with possible hardening and protection against cyber-attacks.
- The user can connect to the server via any web browser. Expected access mechanisms are a mobile phone and a desktop computer, in the order of preference.
- Only 8 users can connect to the server at a time. For future development, the server can be extended to handle more users, more controllers and more devices per controller.

7. TESTING

7.1 Test Cases

<table>
<tr><td colspan="4">Test case ID: 1</td><td colspan="3">Test Designed By:</td></tr>
<tr><td colspan="4">Test Module Name: WiFiconnection</td><td colspan="3">Test Designed:06-Aug-2019</td></tr>
<tr><td colspan="4">Test Priority:-Normal</td><td colspan="3">Test Executed by:</td></tr>
<tr><td colspan="4">Test title: Connect to WiFi</td><td colspan="3">Test executed on : 06-Aug-2019</td></tr>
<tr><td colspan="4">Description: Test to check WiFi connectivity.</td><td colspan="3"></td></tr>
<tr><td colspan="7">Pre-Condition: The meter must be programmed with WiFi SSID and Password</td></tr>
<tr><td colspan="7">Dependencies</td></tr>
<tr><td>Step</td><td>Test step</td><td>Test Data</td><td>Expected Result</td><td>Actual Result</td><td>Status(Pass/Fail)</td><td>Notes</td></tr>
<tr><td>1</td><td>Turn on the smart meter with mains voltage.</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>Wait for 30 seconds for the meter to initialize.</td><td></td><td>Meter connects to WiFi and sends data to server.</td><td>Connected</td><td>Pass</td><td></td></tr>
<tr><td colspan="7">Post Condition:The meter must send the same data to the server.</td></tr>
</table>

Table 5: Test case 1

<table>
<tr><td colspan="4">Test case ID:2</td><td colspan="3">Test Designed By:</td></tr>
<tr><td colspan="4">Test Module Name: Measure voltage</td><td colspan="3">Test Designed:21-Aug-2019</td></tr>
<tr><td colspan="4">Test Priority:-Normal</td><td colspan="3">Test Executed by:</td></tr>
<tr><td colspan="4">Test title:Check voltage measurement</td><td colspan="3">Test executed on : 21-Aug-2019</td></tr>
<tr><td colspan="4">Description: Test to check accuracy of voltage measurement module.</td><td colspan="3"></td></tr>
<tr><td colspan="7">Pre-Condition: the meter must be provided with the mains voltage</td></tr>
<tr><td colspan="7">Dependencies</td></tr>
<tr><td>Step</td><td>Test step</td><td>Test Data</td><td>Expected Result</td><td>Actual Result</td><td>Status(Pass/Fail)</td><td>Notes</td></tr>
<tr><td>1</td><td>Turn on the smart meter with mains voltage.</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>Wait for voltage measurement</td><td></td><td>Meter measures the voltage accurately.</td><td>Connected</td><td>Pass</td><td></td></tr>
<tr><td colspan="7">Post Condition:The meter must send the same data to the server.</td></tr>
</table>

Table 6: Test case 2

Test case ID:3			**Test Designed By**:			
Test Module Name: Measure current			**Test Designed**:14-Sep-2019			
Test Priority:-Normal			**Test Executed by**:			
Test title:Check current measurement			**Test executed on** :14-Sep-2019			
Description: Test to check accuracy of current measurement module.						
Pre-Condition: the meter must be provided with the mains voltage and connected to a load						
Dependencies						
Step	**Test step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**	**Notes**
1	Turn on the smart meter with mains voltage.					
2	Wait for current measurement		Meter measures the current accurately.	Connected	Pass	
Post Condition: The meter must send the same data to the server.						

Table 7: Test case 3

<table>
<tr><td colspan="3">Test case ID:4</td><td colspan="4">Test Designed By:</td></tr>
<tr><td colspan="3">Test Module Name: Power calculation</td><td colspan="4">Test Designed:01-Oct-2019</td></tr>
<tr><td colspan="3">Test Priority:-Normal</td><td colspan="4">Test Executed by:</td></tr>
<tr><td colspan="3">Test title:Check power calculation</td><td colspan="4">Test executed on : 01-Oct-2019</td></tr>
<tr><td colspan="3">Description: Test to check accuracy of power calculation function</td><td colspan="4"></td></tr>
<tr><td colspan="7">Pre-Condition: the meter must be provided with the mains voltage and connected to a known load</td></tr>
<tr><td colspan="7">Dependencies</td></tr>
<tr><td>Step</td><td>Test step</td><td>Test Data</td><td>Expected Result</td><td>Actual Result</td><td>Status(Pass/Fail)</td><td>Notes</td></tr>
<tr><td>1</td><td>Turn on the smart meter with mains voltage.</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>Wait for power calculation result</td><td></td><td>Meter calculates power accurately.</td><td>Connected</td><td>Pass</td><td></td></tr>
<tr><td colspan="7">Post Condition:The meter must send the same data to the server.</td></tr>
</table>

Table 8: Test case 4

Test case ID:5			**Test Designed By**:			
Test Module Name: View data from server			**Test Designed**:22-Oct-2019			
Test Priority: High			**Test Executed by**:			
Test title:Verify server data			**Test executed on** : 22-Oct-2019			
Description: Test to check the functionality of the server						
Pre-Condition: The meter must have submitted some data to the server						
Dependencies						
Step	**Test step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**	**Notes**
1	Open the server page from a web browser.					
2	Go to one of the hourly, daily or weekly data display options.		Server correctly displays the recorded data.	Connected	Pass	
Post Condition:the server must keep accepting data from the smart meter						

Table 9: Test case 5

7.2 Code Snippet

```
#define BLYNK_PRINT Serial
#include <ArduinoOTA.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#define CLOUD  // comment out for local server

#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET 0  // GPIO0
Adafruit_SSD1306 display(OLED_RESET);


charmyhostname[] = "Energy-Meter-V2.0";
constintSensor_Pin = A0;
unsignedint Sensitivity = 185;   float Vpp = 0; // peak-peak voltage
floatVrms = 0; // rms voltage
floatIrms = 0; // rms current
floatSupply_Voltage = 233.0;           // reading from DMM
floatVcc = 5.0;         // ADC reference voltage // voltage at 5V pin
float power = 0;         // power in watt
floatWh =0 ;             // Energy in kWh
unsigned long last_time =0;
unsigned long current_time =0;
unsigned long interval = 100;
unsignedint calibration = 100;  // V2 slider calibrates this
```

```
unsignedint pF = 85;    // Power Factor default 95
floatbill_amount = 0;   // 30 day cost as present energy usage
unsignedintenergyTariff = 8.0; // Energy cost in INR per unit (kWh)

void setup()
{
display.begin();
WiFi.hostname(myhostname);
Serial.begin(115200);
Serial.println("\n Rebooted");
WiFi.mode(WIFI_STA);
  #ifdef CLOUD
Blynk.begin(auth, ssid, pass);
  #else
Blynk.begin(auth, ssid, pass, server);
  #endif
while (Blynk.connect() == false) {}
ArduinoOTA.setHostname(myhostname);
ArduinoOTA.begin();
timer.setInterval(2000L, getACS712); // get data every 2s
}

void loop()
{

displaydata();
Blynk.run();
ArduinoOTA.handle();
timer.run();

}
```

7.3 Traceability Matrices

7.3.1 RID vs UCID (requirements vs use cases)

UCID/RID	R 1	R 2	R 3	R 4	R 5
UC 1	✓	✓			
UC 2		✓			
UC 3	✓	✓			
UC 4	✓	✓			
UC 5	✓	✓	✓		✓

7.3.2 Test Cases (RID vs TID)

RID/TID	TID 1	TID 2	TID 3	TID 4	TID 5
RID 1	✓	✓	✓	✓	
RID2	✓	✓	✓	✓	
RID3	✓	✓	✓	✓	
RID4	✓	✓	✓	✓	
RID5					✓

7.3.3 Coverage (UCID vs TID)

UCID/TID	TID 1	TID 2	TID 3	TID 4	TID 5
UC 1	✓				✓
UC 2		✓			✓
UC 3		✓	✓		✓
UC 4				✓	✓
UC 5	✓	✓	✓	✓	

8. RESULTS/OUTPUT/STATISTICS

8.1 %completion

As per the traceability matrix that shows the relations between the requirements and use cases, 85% of the undertaking objectives have been finished. The remaining necessities are either minor or were agreed upon with the supervisor to be left from the usage because of their superfluity or being out of space of this current undertaking's degree.

8.2 %accuracy

All the qualitative objectives set in the proposal phase are met in the implementation and development process. The accuracy of the results of the major test cases is as shown in the following.

98% accuracy has been achieved in measuring the voltage and current by the respective sensors.

97% accuracy has been achieved in calculating power from the measured voltage and current values.

98% accuracy has been achieved in viewing saved results from the server.

8.3 %correctness

Most of the targets set in the proposal stage are met in the implementation and testing phases. In spite of the fact that there are some nonfunctional prerequisites that still need executions, the principle target of the undertaking is completely finished. With respect to promote advancements, the nonfunctional prerequisites can be actualized to accomplish an increasingly competent product.

9. CONCLUSION

This smart meter has been installed and tested using several different electrical loads. The meters measures the power being consumed by each load and sends the data to the server. We have also tested the meter for endurance and reliability for several days. It can run continuously and provide electricity measurements 24/7 without any interruptions, if it is provided with a continuous source of power and an internet connection.

The user can view their power consumption on the server using any web browser. The data is presented in a simple and effective way in three different time settings. The server keeps the record of all data permanently so the user can view their past records any time.

The assorted data about the electricity consumption significantly helps the user to identify important times in daily and weekly frames at which they can cut their electricity usage to optimally reduce their energy footprint and electricity bill.

10. FUTURE WORK

This project has a lot of potential for improvement and implementation on a commercial and larger scale because it uses easy simple methodology and cloud based architecture. Some of the suggestions for future development upon this concept are as follows.

- Include some daily and weekly limits on electricity usage.
- Send notification to the user's phoneabout hourly and daily usage.
- Make a central database for several users to compare energy usage. Reward the person with most savings.

11. BIBLIOGRAPHY

11.1 Books

1) Getting started with internet of things by CunoPfister
2) Electrical Energy Meters –Principles and Applications by GossenMetrawatt
3) Electricity Metering in Easy Steps: An outline book on smart energy meters for everyone by Dr. ShashikantBakre
4) Building the internet of things by Maciej
5) The internet of things by Samuel Greengard

11.2 Research papers

- Advanced Metering Infrastructure Based on Smart Meters in Smart Grid by TrongNghia Le, Wen-Long Chin, Dang Khoa Truong and Tran Hiep Nguyen, Published: June 29th 2016

12. APPENDIX

12.1 Pre-requisites

Must have the knowledge of IOT and its hardware.The mathematical basis of measuring and calculating electrical power in AC circuits is also required.