REMOTE ENERGY MONITORING AND PROFILING SYSTEM

By

NC Rafia Malik
NC Sanam Rashid
NC Anam Ajmal
PC Sulaiman Sadiq

Project Supervisor

Dr. Adnan Rashdi

Submitted to the Faculty of Electrical Engineering, Military College of Signals National University of Sciences and Technology, Rawalpindi in partial fulfillment for the requirements of a B.E. Degree in Telecom Engineering
JULY 2011

ABSTRACT

Traditional meter reading for electricity consumption and billing is done by human operators from house to house and building to building. This requires huge number of labor operators and long working hours to achieve complete area data reading and billing. Human operator billing is prone to reading error as sometimes the houses electric power meter is placed in a location where it is not easily accessible. Labor billing job is sometimes also restricted and slowed down by bad weather conditions. The increased development of residential housing and commercial building in a developing country, such as Pakistan, requires more human operators and longer working hours to complete the meter reading task. In order to achieve efficient meter reading, reduce billing error and operation costs, Automatic Meter Reading (AMR) system plays an important role to address the above mentioned problems. AMR is an effective means of data collection that allows substantial saving through the reduction of meter re-read, greater accuracy, allows frequent reading, improved billing and customer service, more timely energy profiles and consumption trends updates, and better deployment of human resource.

In this project, GSM technology will be used to implement an AMR system. The GSM Energy Profiling System (GEPS) takes advantage of the available GSM infrastructure's nationwide coverage and the Short Messaging Service (SMS) to transmit energy reading, from the digital electric meters, to the supplier (server) and receive alerts at the consumer (User) end. An interface has been developed at the server end to receive the readings transmitted by the consumers and update the records in the billing/consumption database. The data collected is also uploaded to the web server and illustrative energy consumption profiles of all users are maintained. An additional feature explored is the Global

Positioning System (GPS) to indicate the location of consumers (traffic profile). This would be beneficial if used in collaboration with sensor circuits to indicate meter theft.

If commercially employed, this is a comprehensive system which accurately maintains and illustrates energy usage data via an efficient monitoring and profiling system. The system also has the flexibility to include extensive control by the supplier company.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To Almighty Allah, for Whose greatness we do not have enough words,

To our parents and friends, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible

ACKNOWLEDGEMENTS

There is no success without the will of Allah. We are grateful to Allah, Who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality.

We are also grateful to our parents, family and well-wishers for their admirable support. We would like to thank our supervisor Dr. Adnan Rashdi, for his help and motivation throughout the course of our project. Without his help we would have not been able to accomplish anything. He provided us with the opportunity to polish our technical skills and guided us into this area of learning.

We are also indebted to IESCO, especially CEO Mr. Javed Parvez, for believing in our project and extending their full support and unconditional help. Their faith in our project was a constant drive towards achieving our aims and even exceeding them.

We are thoroughly obliged to Director Finance PESCO Anwar-ul-Haq and Maam Sana Javed for their courteous help in leading us to developing a practical approach to this project. We extend our gratitude to Dr. Shoaib and Sir Safwat Irteza, for being a source of inspiration and innovation in our project selection phase.

We offer our regards and blessings to the University Administration especially our Lab Staffs, and Akhtar for bearing with us throughout the course of our project and supporting us during the completion of this work.

Words will fail us in order to thank Rabia Malik, Sir Bilal Bashir and Salman Ali Abbassi for their selfless help and unconditional support throughout our project. Without their patronage and guidance, this project would not have been possible.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SIM	Subscriber Identity Module
GSM	Global System for Mobiles
GPS	Global Positioning System
EPS	Energy Profiling System
USB	Universal Serial Bus
GEPS	GSM Energy Profiling System
SMS	Short Messaging Service
AMR	Automated Meter Reading
HV	High Voltage
SOC	System on Chip
MIU	Meter Interface Unit
TTL	Transistor Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter
TX	Transmitter
RX	Receiver
ISR	Interrupt Service Routine
IE	Interrupt Enable
TI	Transfer Interrupt
RI	Received Interrupt

1. Introduction

1.1. Overview

Pakistan has been facing an unprecedented energy crisis since the last few years. It is a developing country and its need for energy is extremely high and growing; industrial sectors are expanding, but they are experiencing an energy shortage. We need to focus on conserving energy rather than relying solely on exploring alternate energy sources, which requires curbing the extravagant use of energy. Given the severity of this crisis, Energy supply companies are being forced to seek resolutions. The first step to conservation is by inculcating a sense of awareness in consumers about their consumption on frequent basis. If the consumer can conveniently check his bill-so-far, he would automatically resort to curtailing his energy usage. In order for the electric supply companies to limit the energy usage given the current crisis in Pakistan, the first step towards control is through an efficient monitoring system.

In this project, we present the development of a GSM Energy Profiling System, illustrated in Figure 1. The system consists of digital energy meters, installed in every consumer unit, and an energy monitoring system at the electric supplier side. The digital electric meter used is a single phase A283 Meter, developed by MicroTech Limited, Pakistan with an appended transmission module, which takes the meter reading and utilizes the GSM network to transmit the energy usage reading using Short Message Service (SMS) back to the energy supplier wirelessly. At the supplier end, a monitoring system is used to manage all received SMS meter readings, compute the billing cost, update the database and maintain an energy consumption profile for each

user. Various alerts and control can also be generated by the supplier. A working prototype of the complete system has been developed to demonstrate the effectiveness and efficiency of Automatic Meter Reading, billing and notification through the use of GSM network.

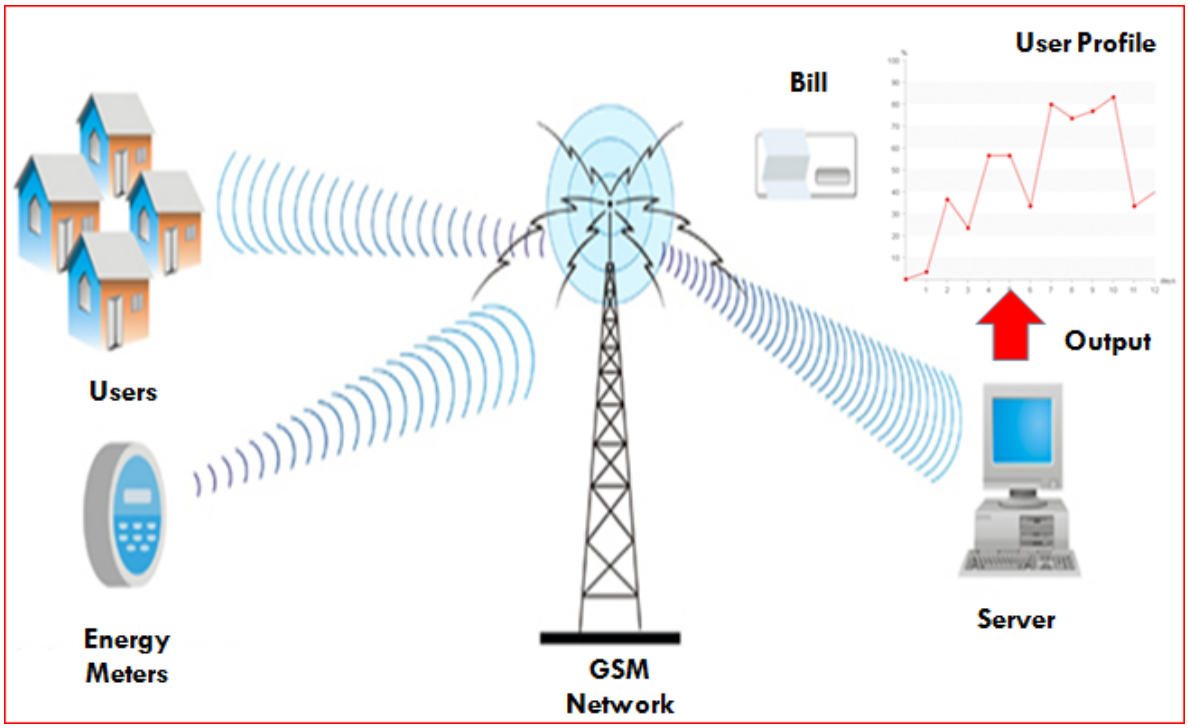

Figure 1: GSM Energy Profiling System

1.2. Problems

Our country has been facing the electricity crisis for quite some time now, which has a major impact on the country's prosperity. The industrial growth, coupled with high losses (due to electricity theft), has resulted in a major demand-supply imbalance resulting in an energy shortage with a devastating domino effect on Pakistan's economy. Energy shortages create blackouts, and without energy to run industries and commerce, companies either shut down or production is reduced greatly. In addition, shipping dates

are drawn out, work forces are reduced and companies' competitive effectiveness in world markets is severely limited. A way out of the energy crisis is to bridge the gap between supply and demand.

1.3. Statement

The Project has been developed to induce transparency in the current meter reading system, by facilitating real time energy consumption monitoring. Automation would lead to an efficient energy metering system by removing human errors. The project also allows the supplier company to implement remote control to some extent. A major feature is the inclusion of a user consumption profiling system, accessible to users and the energy supply company. By incorporating control coupled with profiling, the project aims at creating some degree of awareness among users, encouraging them towards conservation of energy.

1.4. Approach

The basic approach was to implement an automatic meter reading system. In order to devise such a system, the idea was to make use of the existing energy meters, such that the developed module could be affixed to the already installed meters, instead of replacing them with the expensive smart meters that would have resulted in large replacement cost. These modules were designed to be capable of transmitting energy readings, by making use of the existing GSM infrastructure. The selection of GSM network as the transmission medium has been due to the many advantages that come along. It provides an economic transmission means, along with security via data encryption. Moreover, since the GSM infrastructure is widespread, it saves installation cost of a new transmission system infrastructure. The readings from all meters in an area are then to be received at a server and stored in a database. To fulfill this purpose, a

Windows application was required. It was developed to extract the consumption data from the received text messages and record them in the database system. Database is to store the raw data, which needs to be processed into legible information. The information is in the form of illustrative user consumption profiles, along with energy usage history and billing information. The interactive format of the information makes aberrant behavior easier to detect. The profiling system is accessible to the user which inculcates awareness amongst the consumers and is conducive to energy conservation. Since the supplier company can also view the usage information in real time, it leads to the development of an efficient and transparent monitoring system, which lays down the foundation of the concept of remote control. This remote control may be in the form of warnings to the consumer that would alert the user of any abnormal usage patterns, or in dire need, may also be used in an authoritative way by the supplier company to remotely shutdown power to any consumer.

1.5. Objectives

The project was developed with the drive of taking a step towards the amelioration of the current energy crisis in Pakistan. Various objectives were set and hence after, were successfully achieved. A brief overview of the project aims is as follows:

a) To develop a real time system to monitor the energy consumption at short regular intervals of time, as opposed to the conventional monthly meter reading.
b) To maintain an illustrative profiling system portraying the energy usage and associated billing information.
c) To induce transparency and efficiency in the process of meter reading by eliminating the need for personnel to record the meter readings.

d) An easily accessible profiling system leads to two way advantages; to the supplier to easily detect any abnormal trend in energy consumption, and to the consumer by instilling awareness and thereafter, paving the way towards energy conservation.

e) To remove the possibility of meter theft by installing meter trackers using the Global Positioning System, indicating the location of all live meters.

f) Recovering lost meter readings, owing to unfavorable circumstances (bad weather, network congestion, etc), by providing the supplier with discretion to demand forced feedback from the consumer-polling.

g) To enable the supplier to deploy some level of remote control in the form of alert messages to the consumer in case of aberrant behavior, or remote shutdown, when needed.

2. Literature Review

To survive in today's business environment, companies have to be innovative, either for the need to match the increasing competitions among electric utilities, or for meeting the ever-increasing demand from the end-users. The need to utilize electricity distribution, consumption, and to develop more efficient load management techniques have become increasingly more important on the day-to-day basis. One potential solution to all of those issues is the development of automatic meter reading systems, or an Energy Profiling System (EPS).

Like the name suggests, EPS refers to the collection of data from electronic meters or other devices and then automatically transmitting the collected data via communication links without any human intervention. A general EPS system mainly consists of three parts - the meter interface module, the data concentrator system and a central computer system.

The earliest form of AMR systems developed were fixed carrier, analogue systems that were first put to trial in the 1950s using the ripple control signalling method. These systems were originally used for residential and commercial load control and tariff switching that helped utilities to offset peak demand to maintain their services. Back then the systems used conventional high voltage (HV) transformers to couple communication signals between 30 Hz to 50 kHz to and from the HV power lines. The ripple control systems deployed, used carrier frequencies that were kept between 30 Hz to 1 kHz with their data transmission rates at a bandwidth of 10 Hz. All of the earlier systems were only capable of providing one-way communication.

New systems were not developed until the early 1980s. These systems offered a slightly higher data transmission rate. Then investigations were made in the mid 1980s by several utility corporations to analyze the characteristic properties of the electric grid as a medium for communication. Signaling frequencies in the range of 5 - 500 kHz were looked at. Main areas of investigation were the signal-to-noise levels affected by the power channel, as well as the attenuation of the signal by the transmission grid. As a result of the extent of the researches, bi-directional communication was developed in the late 1980s and the early 1990s, where the present systems came onto the market during this time frame. The main difference in the newer system was the use of much higher frequencies (the wideband frequency range, often in the MHz range) and a substantial reduction of the signal levels. Through this development, two-way communication became realistic.

Today, advanced routing protocols are being implemented into AMR systems to make them adaptive to network changes and to achieve better management of data transmissions. The anticipated future development will mainly comprise of looking into the use of frequencies in the GHz range, providing a much higher bandwidth and an even higher data throughput, possibly in the order of mega-bit/second speed range.

The design of an energy profile system is a wide spread task which may be approached with various methods employing multifarious technologies. [4] Using these technologies data can be send from transmitting end to the receiving end. In this project, GSM technology would be used for transmitting the meter reading from one point to other

point. The different types of technologies are described below. Out of which handheld technology is used rarely.

2.1. Handheld

In handheld ESP, a meter reader carries a handheld computer with a built-in or attached receiver/transceiver (radio frequency or touch) to collect meter readings from an AMR capable meter. This is sometimes referred to as "walk-by" meter reading since the meter reader walks by the locations where meters are installed, as they go through their meter reading route. Handheld computers may also be used to manually enter readings without the use of EPS technology. This technology does not make optimum use of the AMR capable meters, as manual meter reading personnel are required.

2.2. Touch Based

With touch based ESP, a meter reader carries a handheld computer or data collection device with a wand or probe. The device automatically collects the readings from a meter by touching or placing the read probe in close proximity to a reading coil enclosed in the touchpad. When a button is pressed, the probe sends an interrogate signal to the touch module to collect the meter reading. The software in the device matches the serial number to one in the route database, and saves the meter reading for later download to a billing or data collection computer. It introduces transparency in the metering system but does not eliminate the need of meter readers.

2.3. Mobile

Mobile or "Drive-by" meter reading is where a reading device is installed in a vehicle. The meter reader drives the vehicle while the reading device automatically collects the meter readings. With mobile meter reading, the reader does not normally have to read the meters in any particular route order, but just drives the service area until all meters are read. Components often consist of a laptop or proprietary computer, software, RF receiver or transceiver, and external vehicle antennas. Due to the short range of mobility, it requires a team for collection of meter readings.

2.4. Fixed Network

Fixed Network EPS is a method where a network is permanently installed to capture meter readings. This method can consist of a series of antennas, towers, collectors, repeaters, or other permanently installed infrastructure to collect transmissions of meter readings from AMR capable meters and get the data to a central computer without a person in the field to collect it.

There are several types of network topologies in use to get the meter data back to a central computer. A star network is the most common, where a meter transmits its data to a central collector or repeater. Some systems use only collectors which receive and store data for processing. Others also use a repeater which forwards a reading from a more remote area back to a main collector without actually storing it. A repeater may be forwarded by RF signal or sometimes is converted to a wired network such as telephone or IP network to get the data back to a collector. Some manufacturers are developing

mesh networks where meters themselves act as repeaters passing the data to nearby meters until it makes it to a main collector. A mesh network may save the infrastructure of many collection points, but is more data intensive on the meters. One issue with mesh networks is that battery operated ones may need more power for the increased frequency of transmitting.

2.5. Radio Frequency Network

Radio frequency based EPS can take many forms. The more common ones are Handheld, Mobile, and Fixed network. There are both two-way RF systems and one-way RF systems in use that use both licensed and unlicensed RF bands. In a two-way or "wake up" system, a radio transceiver normally sends a signal to a particular transmitter serial number, telling it to wake up from a resting state and transmit its data. The Meter attached transceiver and the reading transceiver both send and receive radio signals and data. In a one-way “bubble-up” or continuous broadcast type system, the transmitter broadcasts readings continuously every few seconds. This means the reading device can be a receiver only, and the Meter Reading device a transmitter only.

Data goes one way, from the meter AMR transmitter to the meter reading receiver. There are also hybrid systems that combine one-way and two-way technologies, using one-way communication for reading and two way communication for programming functions. RF based meter reading usually eliminates the need for the meter reader to enter the property or home, or to locate and open an underground meter pit. The utility saves money by

increased speed of reading, has lower liability from entering private property, and has less chance of missing reads because of being locked out from meter access.

2.6. Power Line Communication

AMR is a method where electronic data is transmitted over power lines back to the substation, then relayed to a central computer in the utility's main office. This would be considered a type of fixed network system the network being the distribution network which the utility has built and maintains to deliver electric power. Such systems are primarily used for electric meter reading. Some providers have interfaced gas and water meters to feed into a PLC type system. Besides the advantage of making use of existing technology, this technology has an inherent disadvantage of interference and noise in the PLC, which deems it unreliable. Moreover, in order to improve its performance by reducing interference, leads to employment of expensive equipment. [5]

2.7. Wireless Fidelity (Wi-Fi)

Today many meters are designed to transmit using Wi-Fi even if a Wi-Fi network is not available, and they are read using a drive-by local Wi-Fi hand held receiver. Narrow-banded signal has a much greater range than Wi-Fi, so the number of receivers required for the project is far fewer than the number of Wi-Fi access points covering the same area. These special receiver stations then take in the narrow-band signal and report their data via Wi-Fi. Compared to narrow-band burst telemetry, Wi-Fi technology uses far too

much power for long-term battery-powered operation. Thus Wi-Fi is the efficient mean of communication in AMR technologies, which allows communication between the central data base and the end users. A major disadvantage in this technology is the limited range of Wi-Fi, hence reduced coverage area, which restrains its use when it comes to commercial implementation. Not being a widespread technology, it requires the installation of access points to cover the designated areas.

2.8. Proposed Solution

The proposed design is based on the Fixed Network EPS, where the existing GSM infrastructure has been utilized as the transmission network and digitized pulses from the energy meter are used as the method for data acquisition.

GSM is an efficient and mature technology as compared to other meter reading collection systems. Being equipped with the facility of remotely accessing the meter readings, it eliminates the need for employing meter reading personnel along with reducing the human error involved. Also, GSM provides a reliable means for transmission of these readings by using data encryption. GSM infrastructure is widespread and low cost, resulting in greater coverage area along with economic transmission means.

The pulses generated by an energy meter relate to the energy consumed and since they are signified by an easily detectable voltage change, this ensures accuracy of the data acquired by the meter. The pulses generated by the meter can directly be input to the

microcontroller without any voltage level conversion. Also since the pulses are already digitized, no analog to digital converter circuitry is required.

2.8.1. Earlier Systems using GSM

Earlier systems used GSM along with incorporating other technologies like RF or Ethernet to channel messages containing meter readings through SMS gateways. [3] This involved laying down the infrastructure for a Local Area Network (LAN), which was both expensive and not an immediate solution. Apart from this, by defining gateways, the reliability of the system is inhibited since if the gateway goes down, the meter readings from all the meters connected to that gateway would fail to reach the server.

2.8.2. Earlier systems using pulse detection

Previously systems employing pulse detection as data acquisition method have been developed using other technologies like PLC as the transmission system. [1] Earlier systems focused mainly on the data acquisition and monitoring part and did not form a proper profiling system. The proposed solution however is to develop a comprehensive and ubiquitous profiling system, accessible to the user and the consumer with illustrative details.

Former systems also made use of fixed time intervals [1] which becomes a problem incase the number of consumers increases as this may cause congestion in the GSM network. On the other hand, the proposed solution sets the time intervals based on the number of users to cater for. Lesser the number of consumers, shorter the time interval and vice versa.

3. Design and Development

Concept of our Project

The idea of this project is based upon the ageless technology of GSM network to aid the development of a Remote Energy Monitoring System. The literature review consisted of many research papers and projects but we finally zoned in on "Automatic Power Meter Reading Using GSM Network" by H.G. Rodney Tan, IEEE, C.H.Lee and V.H. Mok, IEEE published in "The 8th International Power Engineering Conference (IPEC 2007)".[5]

Dedicated meters are installed at the user end. For prototype development, the existing digital electrical meters were used and aim was to to interface the meters to transmission module via a microcontroller circuit. The microcontroller takes the reading from the meter and then formulates a message including the reading and user ID. This message is then transmitted to the server over the GSM network.

A GPS module is also incorporated at this end. The microcontroller is interfaced with the GPS module such that it receives data from GPS module, extracts the location of the meter, and transmits this location to the server end. The location can be sent in the message containing the meter reading, once per day, or as per requirement.[6]

The microcontroller is programmed to receive and extract commands from the server in form of SMS, so as to incorporate control in the prototype.

At the Server end, data received via SMS would be recorded in a database and updated to a website. The website can be accessed by the electricity power supplier as the server system, and the user (with his unique login account) as the client system.

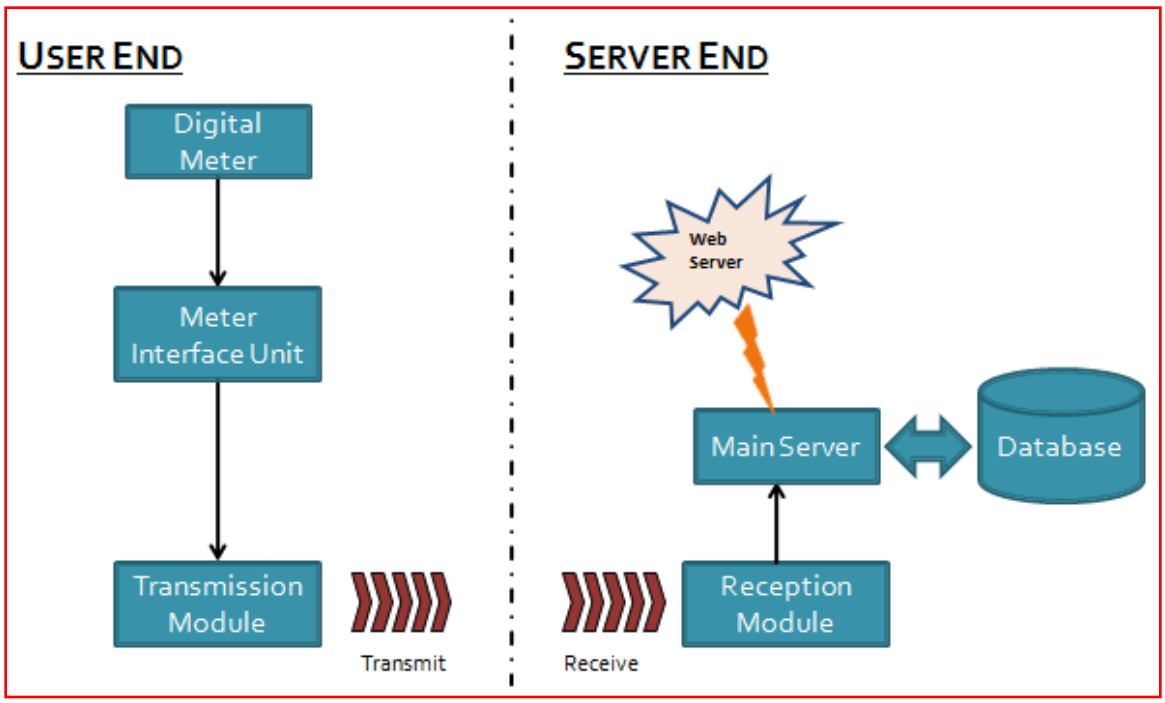

Figure 2: Modular Illustration of the Prototype

Modular Division

We have made progress in individual modules of our project, depicted in Figure 2, and explained as follows,

- Energy Reading from Energy Meter.
- Meter Interface Unit
- Transmission system.
- Receiving System at the server.
- Database Management System
- Profiling System
- Additional Feature-GPS.

3.1. Energy Reading from Energy Meter

The meters being used are single phase digital energy meters A283, manufactured by Microtech Limited, Pakistan. We did reverse engineering on the working of the Teridian microcontroller 71M6521DE of the meter. The reading is recorded in the form of pulses generated by the controller. Using this approach, the processed reading from the Teridian SOC is attained. There are 3200 pulses generated by the meter per kWh of energy consumed. We tested the pulse readings against the reading displayed by the energy meter through a test load of 1000Watts, as shown in Figure 3.

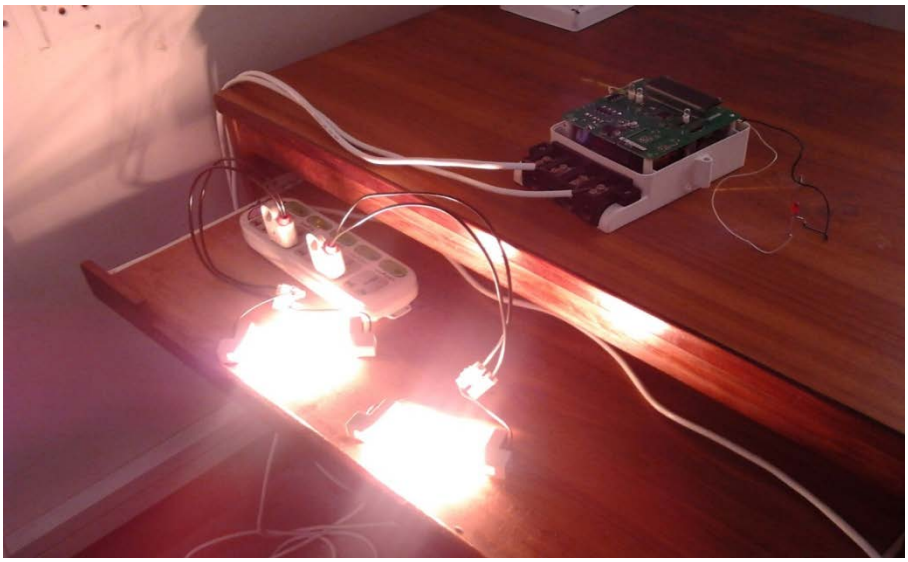

Figure 3: Energy Reading from the meter

3.2. Meter Interface Unit

The transmission system is composed of the meter interface unit and the SMS sending unit. The meter interface unit is the component that is attached to the energy meters installed at the consumer end. It is responsible for getting readings and then processing them into a proper format, fit to be sent as SMS to the server end. It is then includes the

SMS sending/receiving portion which sends the message with necessary fields over the GSM network, and receives incoming messages.

Initially the Atmel controller ATMEGA16L was used and basic circuits were tested with that controller. The AVR controller was over-priced for the project needs so the very basic controller Atmel 89C52 was put into use.

The meter interface unit is directly attached to the energy meters installed at the user end. It gets readings from the meter and then processes it into a format fit to be sent through SMS, which is the job of the transmission component. The meter interface unit was developed by performing a combination of tasks, which are listed and explained below.

3.2.1. Pulse Detector

The meter reading collection is based on the acquisition and counting of pulses, transmitted by energy meters. Electrical energy meters produces pulses as energy is being consumed. The frequency with which these pulses are generated is proportional to the amount of energy being consumed. Hence, greater the energy being consumed, greater will be the frequency of these pulses being generated.

In order to get reading from the meters, the first task was to detect and count these pulses. Therefore, the basic meter reading acquirement step was the development of a basic pulse detector circuit. For each pulse received by the energy meter at a specific pin of the microcontroller, an LED blinked. This confirmed that the pulses generated by the meter could actually be used by the microcontroller to do energy calculations.

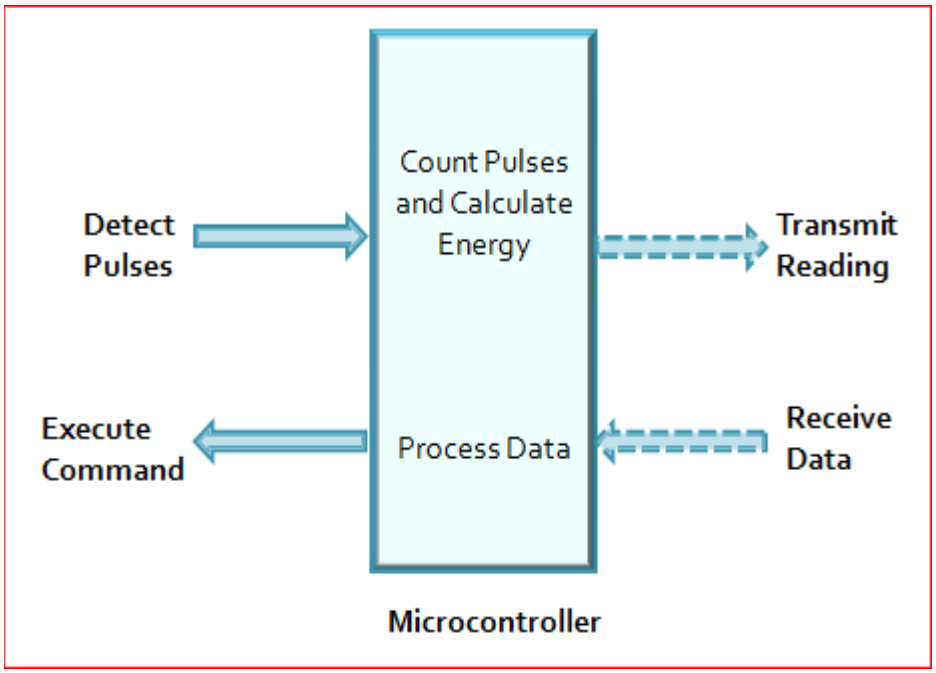

Figure 4: Illustration of Microcontroller Basic Functioning

3.2.2. Pulse Counter

Next step comprised of the summation of the number of occurrences of the pulses detected, and computation of the energy value accordingly. The functioning of the microcontroller has been shown in Figure 4. So it was required to develop to count the pulses received by the controller. For the testing phase, 3200 pulses per kWh were simulated with 20 pulses per increment in energy. That is, for actual calculations the units of energy were to be incremented whenever a total of 3200 pulses were detected. Whereas, for the prototype, the figure of 3200 was replaced by 20, to make testing feasible.

3.2.3. Serial Communication

The energy calculated by counting the pulses is then transmitted over the GSM network to the remote server via a GSM modem. The GSM modem is serially interfaced with the microcontroller via a level converter circuit. The level converter circuit is incorporated to

shift the voltage level from TTL at the microcontroller to RS-232 at the GSM serial interface and vice versa. The RS-232 serial port connector pinouts and its configuration is given in Figure 5. The message includes the energy reading and the SIM number, which signifies the user identity. The number of times a message is sent in a day from one user's end, or the interval between two messages can be set, as per requirement.

Serial communication takes place between the microcontroller and the GSM modem for transmission and reception of SMS. It carries out transfer of information in the form of serial transmission of bits, which maybe done through synchronous transfer or asynchronous transfer. The rate of data transfer in serial data communication is stated in bps (bits per second), and another widely used terminology for bps is baud rate. It is modem terminology and is defined as the number of signal changes per second. In modems, there are occasions when a single change of signal transfers several bits of data.

Initially the prototype was interfaced with Wavecom GSM Modem via RS-232 DB-9 interface, such that the microcontroller's UART was connected to the DB-9's TX and RX through a level converter circuit to transform the logic levels as required. As the project progressed, for an optimized prototype SIM548c module was used which is a combination of GSM and GPS. The use of SIM548c obviated the use of level converter as well, as it used TTL logic, compatible with the microcontroller.

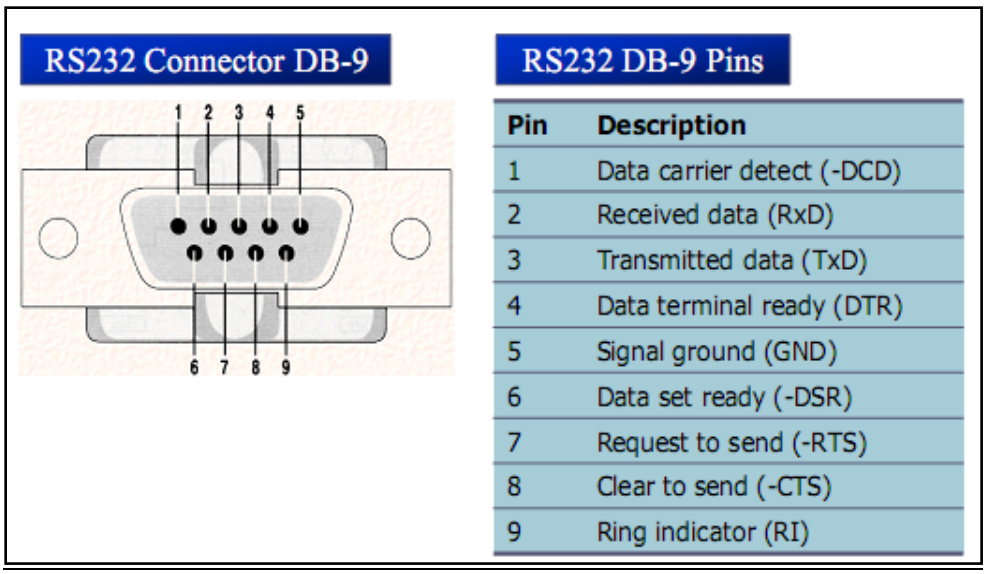

Figure 5: RS232 pin configuration and description

The program developed to send SMS via GSM network, first initializes the serial port. The serial port settings are made, as given in Table 1.

Table 1: Serial Port Settings

TMOD	0x20	Sets to Mode 2
TH1	0xFA	Sets the baud rate to 4800
SCON	0x50	Serial mode 1, 8-Bit data, 1 Stop Bit, 1 Start Bit, Receiving on

Serial Interrupts

An interrupt is an external or internal event that interrupts the microcontroller to inform it that a device needs its service. Upon receiving an interrupt signal, the microcontroller interrupts whatever it is doing and serves the device. The program which is associated with the interrupt is called the interrupt service routine (ISR). Serial communication has a

single interrupt, termed as Interrupt 4, which belongs to both receive and transfer. The interrupts must be enabled by software in order for the microcontroller to respond to them. There is a register called IE (interrupt enable) that is responsible for enabling (unmasking) and disabling (masking) the interrupts.

The data to be transmitted or received is placed in the internal buffer of the microcontroller, SBUF. TI (transfer interrupt) is raised when the last bit of the framed data, the stop bit, is transferred; indicating that the SBUF register is ready to transfer the next byte. RI (received interrupt) is raised indicating that the received bytes need to be picked up before they are lost by new incoming serial data. If the interrupt bit in the IE register is enabled, when RI or TI is raised the 8051 gets interrupted and jumps to memory location 0023H to execute the ISR. In that ISR, TI and RI flags need to be examined to see which one caused the interrupt, and respond accordingly.

3.2.4. Sending Reading via SMS

After setting serial port, the GSM module is configured and checked via the AT commands, the basic AT commands are given in Table 2. The mode of the modem is set to text mode before any transmissions are done, to enable message sending and reading and avoid any errors. It is then followed by the AT+CMGS command which is used to transmit the energy reading. The serial string first goes to SBUF. When the TI flag is set, the reading is then transmitted to the GSM modem serially after the AT+CMGS command, followed by 0x1A, which sends the message. The meter reading is sent after a definite interval, as required, such that after a certain number of units have been consumed, the transmission module, shown in Figure 6, will inform the server end of the energy consumed till then.

Table 2: AT Command Set used to configure GSM Modem

AT	Used to check if the module is available
AT+CMGF	Sets the Modem to Text mode
AT+CMGR	Used to read message
AT+CMGS	Used to send message in Text format
AT+CMGD	Used to delete message

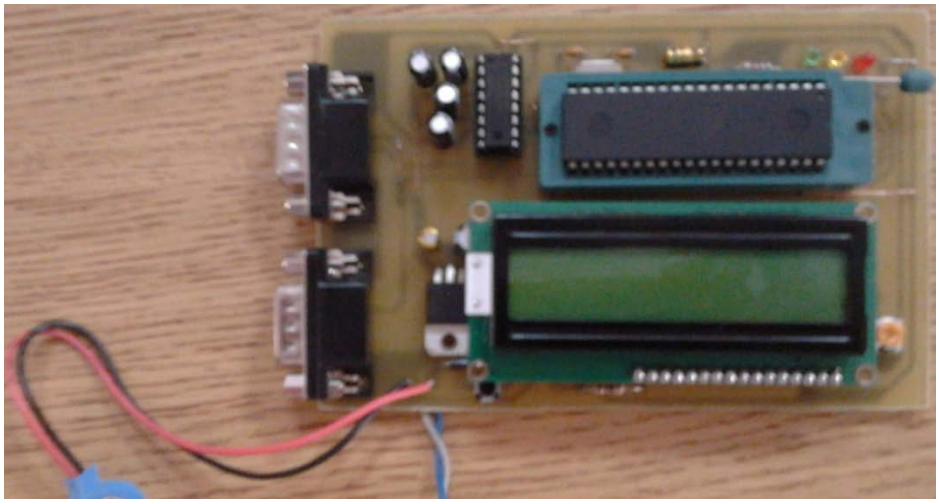

Figure 6: Transmission Module Test PCB

3.2.5. Receiving SMS at the User End

The meter interface unit is programmed to receive SMS sent from the server end, extract relevant information from it, and then act accordingly. This portion lays foundation for incorporating control in the system.

The messages that are received are stored in the SIM memory as and when they arrive. They are then received serially by the microcontroller by using the AT+CMGR command. Before using this command, interrupt 4 is enabled. After the AT+CMGR

command is executed, the serial interrupt pin becomes active for each bit of the message being received on the RX pin of the microcontroller, and results in the execution of the ISR, which comprises of the receive function. The whole message is first stored in the internal buffer SBUF of the microcontroller, which has to be transferred to a self-defined buffer. This buffer is then scanned and the relevant portion of the message is then extracted.

In order to utilize this system for control, the microcontroller can be programmed to interpret the extracted message, by comparing it to some already defined commands. If the microcontroller finds a match to a command, it will act accordingly to fulfill that command by making use of different I/O pins.

3.2.6. Remote Shutdown of Energy Meter

A major feature of control has been incorporated in the system by enabling the server with the facility to both remotely shut the meter down, and power it on. Having this feature, the supply company can cut off the energy connection of any user and reconnect it with equal ease, remotely, without the need to send someone do the task manually.

It can be utilized when the mains supply to a household needs to be cut off, in case the occupants leave the house, and till the arrival of new occupants. It can be put to use when the supply needs to be disconnected due to non-payment of bills. Moreover, if the supply company has a policy of placing restraints on the consumption, to rein the luxurious use as a measure for energy conservation, the defaulters can be penalized by cutting off their supply for a predetermined duration of time.

To include this feature, the microcontroller has been programmed to check for messages intended to shut down or power up the meter. A relay circuit is interfaced with

microcontroller, where the relay is present between the mains supply and the energy meter. The relay is in the normally close configuration, with its enable pin connected to **?** pin of the microcontroller, to switch it open and close. The messages received are compared against the commands of 'On' and 'Off'. If 'Off' is found in the message, the microcontroller sends a low signal on corresponding I/O pin, which switches the relay open, hence, breaking the connection and shutting the meter off. If the message contains 'On', the microcontroller sends a high signal on the same pin to switch the relay close, hence connecting the mains supply to the energy meter, and turning the meter on.

3.2.7. Indication of Alerts

The server end is equipped with the facility to send alerts in the form of messages to the MIU, which is programmed to lit LEDs in correspondence to the message being received. For starters, three levels have been set to point to the MIU whether the level of energy usage is low, moderate or high.

- The SMS '10' indicates the minimum energy consumed, indicated with a green LED.
- The SMS '20' indicates a medium energy consumed, indicated with an orange LED.
- The SMS '30' indicates the maximum energy consumed, indicated with a red LED.

When an SMS is received by the microcontroller from the server, it reads through the contents of the message and lights the respective LED. This is just a check for users to see if they are treading into the 'danger' level of energy consumption. This feature can be enhanced as per the application desired by electricity supply companies.

3.2.8. Polling

An added feature at the server end is that of Polling, which enables the server to poll the transmission end of the prototype to send the reading of the meter, at any time. Making use of both synchronous as well as asynchronous modes to acquire readings from the consumer end infuses flexibility in the system and adds to the control exercised by the server.

The server has to send a message of 'UPD' (Update Reading) to the consumer. When the MIU at the user end receives such kind of a message, and compares the extracted content, it'll interpret it as the update reading message, i.e. the polling alert, and would respond by transmitting the meter reading till that moment.

This proves beneficial in case the server end missed any reading, which might happen due a number of unforeseen reasons. Moreover the administrator can poll any user end for its current energy reading as and when required.

3.2.9. Startup Message

The meter interface unit has been provided an alternate power source in case the supply from mains is cut off, or if the meter has been shut down. This is to ensure that the meter interface unit still continues working and to avoid any reset, as that would result in loss of previous record of consumed units in the meter interface unit. To make the system fool proof even if the alternate power source is unavailable to the MIU, the microcontroller is programmed to indicate to the server that the MIU has been reset and the reading being sent includes only the newly consumed units.

In normal operation, whenever the server end receives any reading, it compares that reading to the one previously received from the same user, and finds the difference to

find the number of units consumed after the previous reading sent. This difference is then added to the total consumption being recorded for that very user. But if the MIU is reset, the received reading has to be added to the total consumption as it is, because it only gives the number of units consumed after the previous reading and is not cumulative of previously consumed units. This is where the startup message becomes useful.

The first reading sent by the consumer end contains an ‘S’ before the reading(e.g. S60), which serves as an indication to the server end that the MIU has been reset and the reading needs to be added directly. This startup message adds further reliability to the prototype.

3.3. Transmission Module

The transmission system utilizes the existing GSM network. GSM technology incorporates encryption hence the messages transmitted are secure. Dedicated SIMs or numbers would be utilized for the energy reading transmission since lack of service or network congestion cannot be afforded. Various users send timed messages containing their respective readings so that the server receives one message at a time and SMS is delivered without delay.

3.4. Receiving System at the Server

A USB GSM modem is connected to the server system. The GSM modem receives the energy reading messages from the various users. We developed a windows form application in Visual Studio 2010 C#.Net. We wrote a code such that an unread SMS received by the Modem is saved as a text file in the server System’s hard drive. This is not essential for our project but we are keeping it as a record for checking the messages

received against the readings updated in the database. The content of the message, that is the energy reading, and the sender ID and the time the message was sent is all saved in the database.

3.5. Windows Application

In order to receive messages from the GSM modem connected at the server through the USB interface, a windows form application, shown in Figure 7, was developed in C#.Net. The application was developed with two purposes

- Receive messages from the GSM modem
- Sift through the message and extract the consumer number and energy reading
- Upload the consumption readings into the database
- Detect usage exceeding threshold
- Detect location change of meter
- Alert the consumer of excess energy usage and/or meter theft via text message

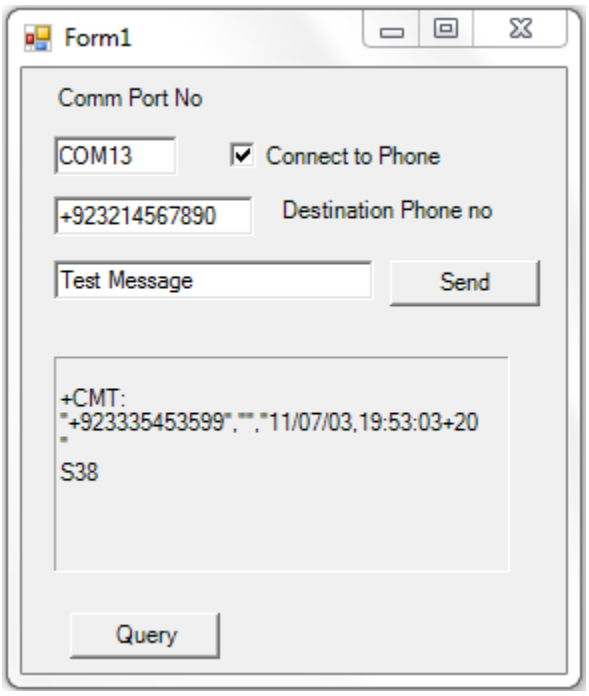

Figure 7: Application Form at the Server End

The USB GSM modem is connected on the virtual communication port, COM13. Once the COM port is entered in the text box, the “Connect to Phone” check box is checked. This tests the communication channel with the modem, and if the connection has been established, “Modem Initialized OK” message is displayed in the label box. Whenever a new SMS is received, an interrupt is generated and the message is read from the port and displayed in the label box. As shown above, the message is received from the number “+923335453599”, sent on “11/07/03 19:53:03”, and the message body contains the text “S20”. The letter S indicates a startup message, which is received when a modem at the consumer’s end is first switched on and the pulses are counted starting from zero, hence the received reading is directly inserted into the consumption table with SIM number as the identification. Subsequent readings are received without the prefix “S” and would be processed before submitting to the consumption table in the MySQL database.

The form application can also be used to send an SMS. For this purpose the receiving entity's number is input into the Destination Phone number text box. The message contents to be sent are written in the text box displaying "Test Message" by default. When the "Send" button is clicked, the message is transmitted. In the example shown above, the message transmitted is "Test Message" to the number "+923214567890".

3.5.1. Receive SMS from the GSM modem

First a serial port is defined which is associated with the COM port that the USB GSM modem is connected to. Whenever a new message is received on the serial port an event occurs. A SerialPortDataEventHandler reads the data incoming on the serial port and continuously stores it in the buffer. When the complete SMS has been received the contents of the buffer are written into a variable for further manipulation. All this preprocessing is saved as a datalink library file which is included in the main project code file.

3.5.2. Sift through the message and extract the consumer number and energy reading

First the received SMS is compared to a regular expression defined according to a fixed format in which an SMS is received. The successful matches are stored in a group which are then separated into the individual variables senderNumber, entryTime and units(message content).

```
Regex we = new Regex("^\\+CMT: \"(?<sender>\\+[0-9]*)\",\"\",\"(?<time>[0-
9,:/]*)\\+[0-9]*\"$|^\\S(?<units>[0-9]*)$");
```

If the received message is a startup message meaning that the meter starts counting from zero, the reading will be extracted and uploaded to the consumption table as it is.

But if the message is not a startup message the last uploaded reading is retrieved and subtracted from the received reading to calculate the energy consumption for the time interval between the two messages. To retrieve the latest reading from the database the select query is used to select all of the entries in the units field corresponding to the senderNumber.

Select units from consumption WHERE SIM = 'sim';

The reading of our interest is the one with the latest timestamp which is then stored in a variable for further manipulation using MySqlDataReader.

3.5.3. Upload the consumption readings into the database

An ASP.NET web service which is placed on the web server is invoked to submit these values in the corresponding fields of the consumption table. In the web service a connection with the MySQL database is established and opened. Next a query is executed which inserts the values into the consumption table

Insert into consumption(SIM, entryTime, units) VALUES('{0}', '{1}', '{2}');", sim, entryTime, units

3.5.4. Detect usage exceeding threshold

After the value of units consumed is extracted from the SMS it is compared to a predefined threshold. If the reading exceeds the set limit of energy usage, an alert is generated in the form of an SMS sent to the consumer. According to the type of consumer, commercial or residential, the defined threshold may be adjusted.

3.5.5. Detect location change of meter

For the SMS received containing GPS information another regular expression is defined according to the fixed format in which the SMS is received. The successful matches are stored in a group which are then separated into the individual variables senderNumber, latitude, longitude. These values are then passed onto the web service in which the previously stored values for latitude and longitude in the users table are retrieved using the select query and compared to the received readings in the SMS. In case they don't match a message is sent to the consumer indicating meter theft.

3.6. Database Management System

The database is maintained using MySQL server. Two tables have been maintained in the database. One is the real-time consumption table which logs all energy readings received. And other is the users table which keeps record of all the subscribers. The users table is created initially for all users, and the user's SIM number is used as his unique identification in the database. The users table can only be edited by the administrator while the consumption table is edited with each new received SMS. The content of the received SMS is added as a new row in the consumptions table.

3.7. Web Development

The MySQL database is used to store data in passive form. In order to attain useful and illustrative information from it, a website was developed. The purpose of the website is to generate the energy profiles of users, to view user records and to depict energy consumption and billing information. The website may be accessed from any location by the consumer or the administrator, which makes the monitoring and profiling system remote in true sense. Both the user and administrator, after login, have access to different

types of information, as is required. The content development was done in php scripting language, which gets, inserts or updates data into the MySQL database through SQL queries.

The Login Page

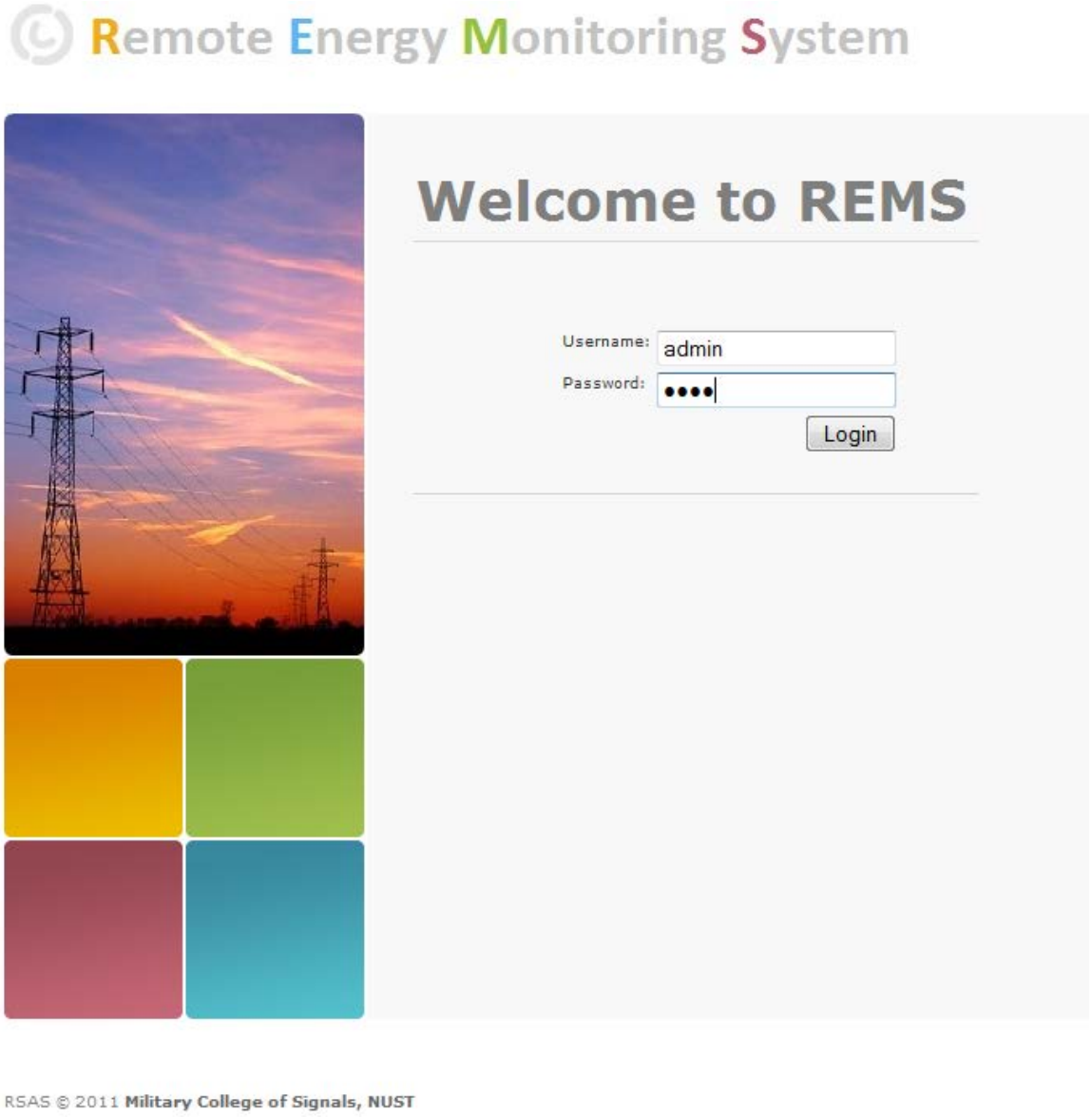

Figure 8: Login Page

The administrator logs into the website through the "**admin**" account, while the users log in through their assigned user names, for example "**user**".

When the username and password is entered, a SELECT query is run at the back end. All data from the “users” table is retrieved, and the entered username and password is matched. If either username and/or password is incorrect, login attempt is rendered unsuccessful and an error is returned.

The initial connection with the database is established using the **mysql_connect** command in PHP.

```
mysql_connect("$sql_host","$sql_username","$sql_pass")
```

The hostname, username and password for the database are specified in the above command. Initially the PHP web pages were tested using local host. Later the website was uploaded at **http://rems.mcs.edu.pk.**

3.7.1. Consumer interfaces

Consumer Home

Figure 9: Consumer page, viewed on login

The homepage, shown in Figure 9, displays general information about the logged in user. Information such as Today's usage, Usage since Last Payment, Bill since Last Payment are updated on refreshing the page. For instance If any energy consumption reading is received by the server while the user is logged in, his consumption history is updated, and the Today's usage shows the added consumption. The Usage and Bill since last payment fields are also updated accordingly.

From the homepage, the user has the option for navigating to his billing info, consumption history or can logout. The logout icon  may also be used.

Billing Info

This shows the graph for the users bill according to each consumption message received by the user along with the date on which the consumption was done. The total bill is also shown.

The bill is calculated according to different rates as per units consumed. For the first 100 units the rate is assumed at Rs. 2.5 per unit. Between 100 and 300 units, the increased rate is Rs. 5 per unit. And above 300 units, the fixed rate of Rs. 7 per unit applies. The usage units are multiplied with the corresponding rate to get the bill. These values of bill for various number of units become the data points for the billing graph.

Consumption History

The consumption history, shown in Figure 10, stored in the consumption table for the logged in user is retrieved using the SELECT query. The consumption data is displayed in tabular form. For the user interested in viewing his consumption history in graphical form, for a specified period of time, a **submit** type form is created in HTML.

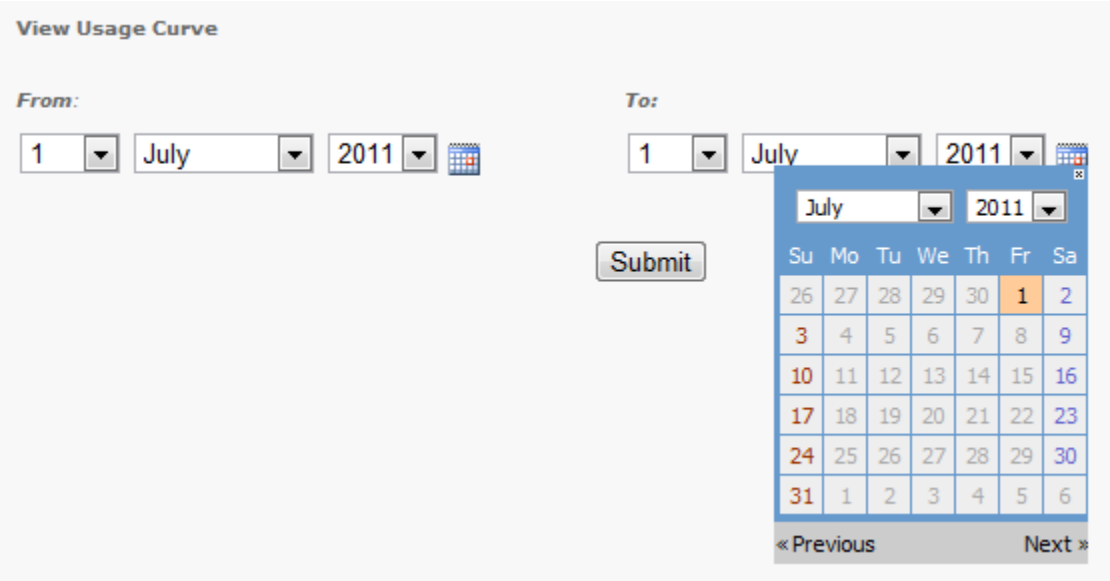

Figure 10: Consumption History

The user specifies the time duration and clicks the **submit** button. A select query is used to attain the data from the consumption table for the specified values of entryTime.

"SELECT * FROM billing.consumption WHERE SIM = "'.$sim.'" AND entryTime between "'.$llim['0'].'" and "'.$ulim['0'].'" ORDER BY entryTime"

The user may also select the date using the calendar, shown by clicking the calendar icon  .

The data is shown as a consumption graph after the **submit** button is clicked.

The Logout Page

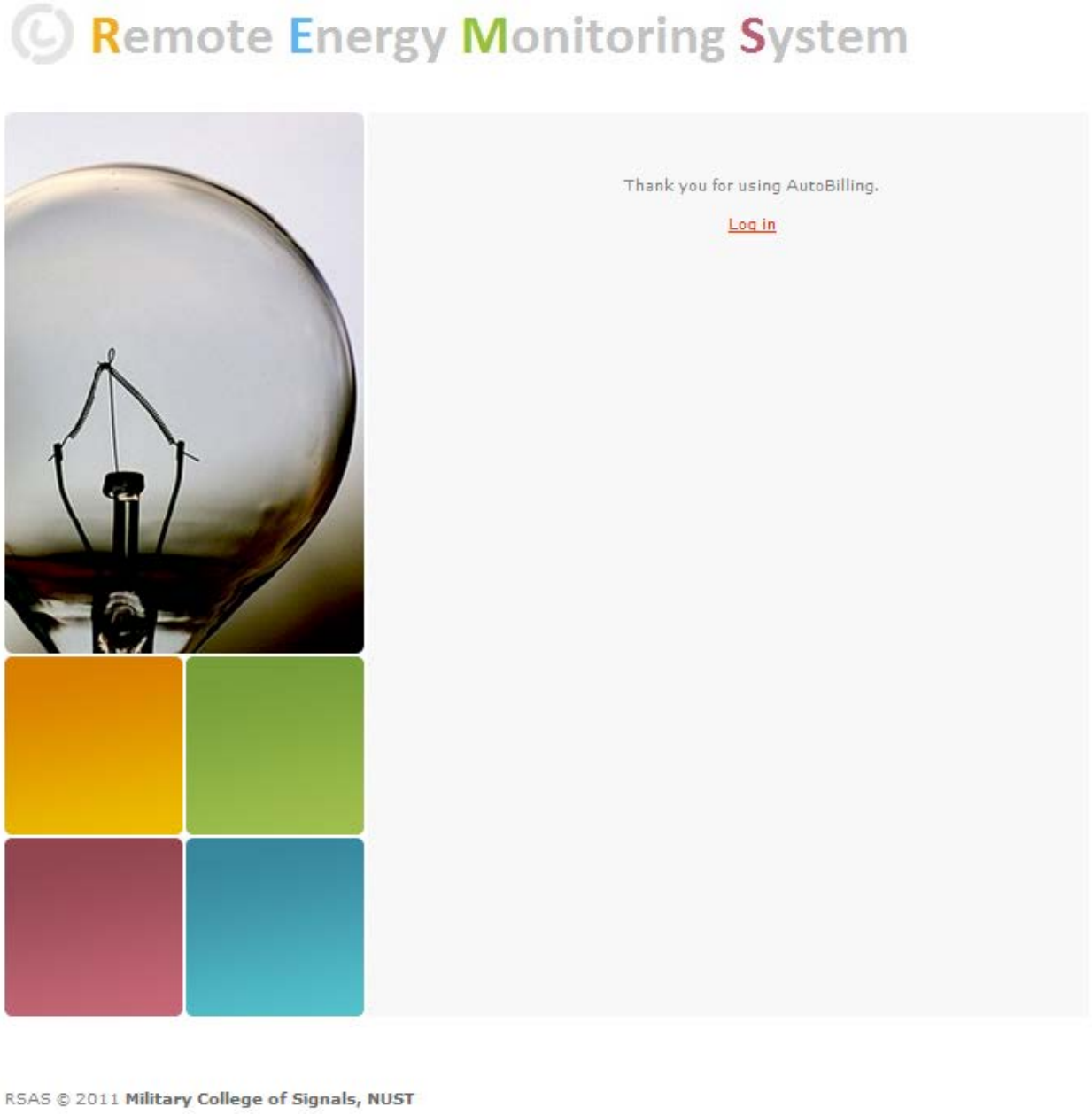

Figure 11: Logout Page

When the administrator or consumer logs out, the session is destroyed and a page, as shown in Figure 11, is displayed. If the user clicks the back button after logging out, he is not directed to the last viewed page, rather he is redirected to the login page. The user would have to login again to navigate through the website.

3.7.2. Supplier interfaces

Administrator Home

Figure 12: Home Page of Administrator

The administrator has the option to add a new user, view the existing users, manage the users or view energy consumption for all or individual users.

View Users

This page shows the records for all users and displays the total number of records in the "**users**" table. The administrator also has the option of searching for a user record, illustrated in Figure 13.

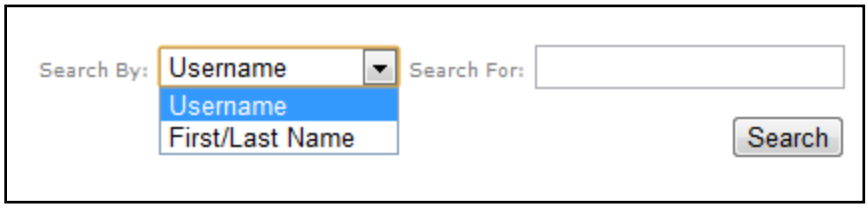

Figure 13: Option to search for user record

The search may be carried out according to the user name or by First/Last Name. A MySQL **select** query is run at the back end to retrieve the corresponding user record with the username or the First/Last Name as the selection factor.

"SELECT * FROM billing.users WHERE $search_by LIKE '%".$search_data."%'"

Add User

The add user shows an HTML **submit** type form, given in Figure 14, in which the administrator can fill out the details of a new user.

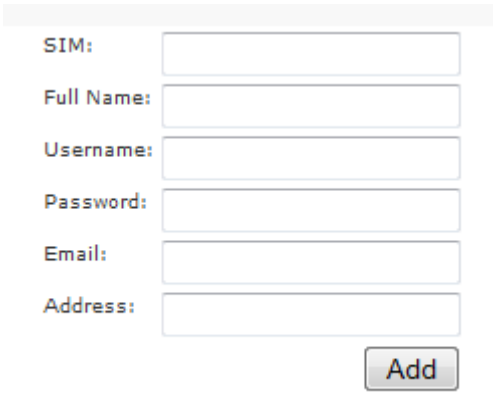

Figure 14: Add User Form

The data entered is stored as **POST** type variables into the corresponding fields in the "users" table in the MySQL database. An "INSERT" type query is used at the back end to communicate with the database.

"INSERT INTO `billing`.`users`
(`SIM`,`fullname`,`username`,`password`,`email`,`address`,`joinDate`,`lastLogin`,`lastPayment`) VALUES
('$SIM','$fullname','$username','$pass','$email','$address',CURDATE(),NULL,NULL)"

Manage

The manage page enables the administrator to view, delete or update the user records.

I. View

When the administrator clicks the "view" hyperlink, a new page, as shown in Figure 15, is opened showing the information of the selected user.

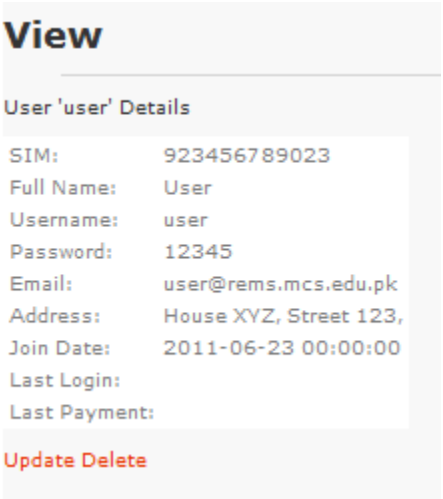

Figure 15: View User Details

A "select" query runs at the backend of this link, which selects all the fields from the "users" table and **echo** command is used in php to display the retrieved information. The

user has the option of deleting or updating the user information from the view page as well apart from the manage page.

II. Update

On selecting the “update” hyperlink, a form is opened, as given in Figure 16. The fields are being displayed using the php **echo** command and the data appearing in grey color represents the in-editable fields, that is these fields are “disabled”. The Update button is the submit type button which takes all the text entered in the form, and updates the MySQL database accordingly.

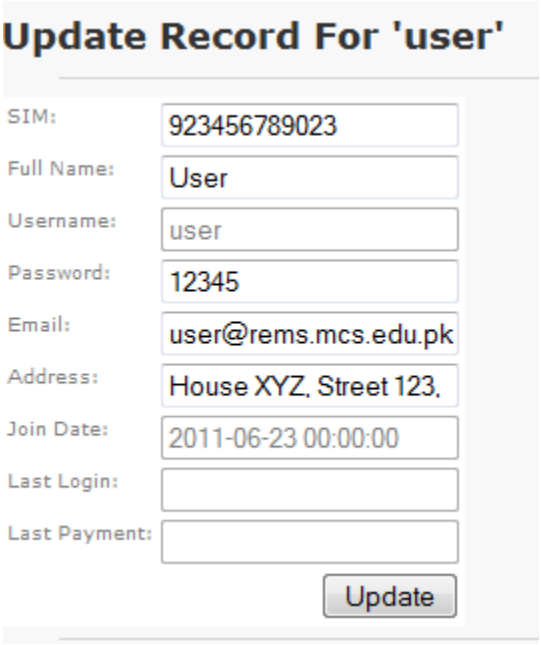

Figure 16: Update Record Form

The administrator can only alter the user name, address, email ID and password. All other fields are disabled for updating since the Join Date, Last Login or Last Payment are fields which are either fixed e.g. Join Date or automatically generated e.g. Last Login. The

MySQL query here is the “UPDATE” query and not the “Insert” query since an existing record is being updated.

"UPDATE billing.users SET password='".$password."', email='".$email."'
,address='".$address."' , fullname='".$fullname."' WHERE SIM='".$SIM."'"

When the update button is clicked, a confirmation message is displayed assuring that the record has been updated.

III. Delete

The “delete” link is used to delete the user from the database.

View Consumption

The view consumption page is the actual implementation of a profiling system. The administrator can view the yearly total consumption of all users, or can even view intricate consumption details for any particular user in a specified month. This enables the administrator to become aware of any abnormalities in the trend of energy consumption and preventive or control measures might be taken thereafter.

Two **submit** type HTML forms have been created here on a single page. When the administrator wants to view the consumption for a single user for a particular month, he specifies the month and SIM number and clicks the submit query buttons, in the form shown in Figure 17.

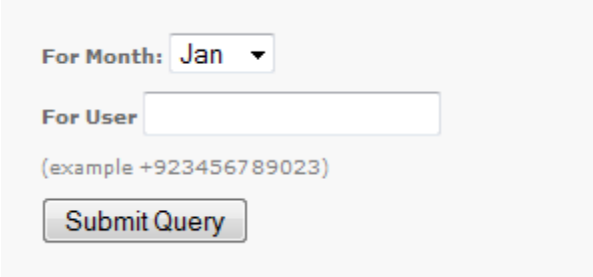

Figure 17: Input to view monthly consumption

A "select" query is used to retrieve information from the "consumption" table according to the SIM number. After all the consumption records have been retrieved, the records for the particular month are sifted and plotted as data points in the energy usage curve.

To view the yearly consumption, the administrator enters the year and uses the submit query button to confirm his choice.

A bar graph showing the monthly consumption breakdown is displayed. If an incorrect year, that is some year in the future is entered, an error message is displayed "Enter Correct Year".

Profiling System

This option enables the administrator to have a bird's eye view of all the consumers. Different consumer locations of the users are pin-pointed on the map according to their GPS coordinates (hypothetical coordinates for the prototype), illustrated in Figure 18.

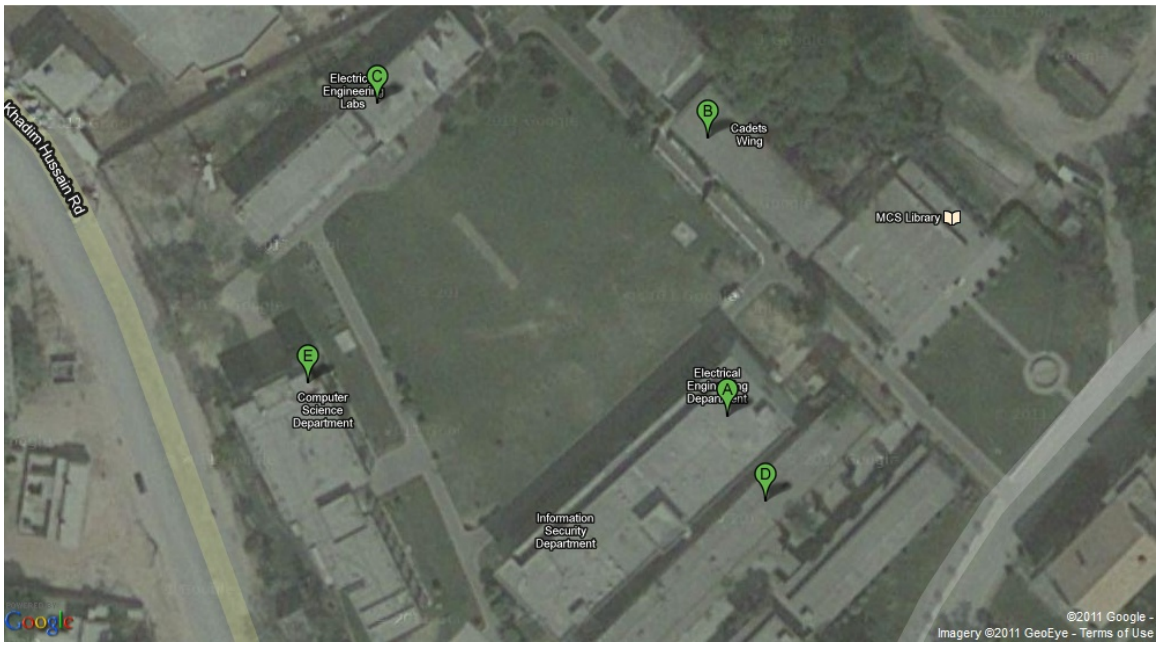

Figure 18: Consumer locations pointed on the map

When the supplier clicks on the marker for any user, for instance A,B,C,D or E, the user's consumption history for the current month is displayed in the form of a usage curve.

3.8. Additional Feature-GPS

After reviewing literature about meter thefts, an additional feature was added to the system to counter these thefts. This required the use of Global Positioning System module, which was interfaced serially with the microcontroller via the level converter circuit already in use. Later on, the SIM548c module was used to optimize the system, which eliminated the use of level converter as both SIM548c and the microcontroller work on TTL logic, and are hence compatible.

The meter interface unit was programmed to calculate the location of the meter with an accuracy of +-15meters through the method of triangulation. The GPS location is then

transmitted via the GSM network in the form of an SMS to the remote server. This can be done just once a day if automatic location of the meter is required to be recorded, but is not a direct module of our project.

The GPS module was initially tested with hyper terminal, serially connecting it to the PC, to observe the pattern in which the information is received from the GPS module. The GPS module sends data in the always NMEA protocol. The code was then formulated to sift the required data pertaining to the latitude and longitude readings only to the accuracy of .0001 minute.

NMEA is a standard protocol, used by GPS receivers to transmit data. They use 4800 bps, 8 data bits, no parity and one stop bit. NMEA 0183 sentences are all ASCII and each sentence begins with a dollar sign ($) and ends with a carriage return linefeed (<CR><LF>). Data is comma delimited, and all commas must be included as they act as markers. Data received from a GPS module is in the form displayed in Figure 19. The microcontroller has been programmed to store the $GPRMC line in a buffer.

$GPRMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,xxxxxx,x.x,a,i*hh<CR><LF>

The RMC message contains the time, date, position, course, and speed data provided by the GPS navigation receiver, details of which are given in Table 3. The program applies a check on '$GPRMC', once a match has been found, the whole line is stored in a buffer, which is then scanned to extract the relevant fields. The program takes the latitude and longitude value and transmits it over the GSM network to the server end. The interval, after which meter location needs to be transmitted, can be set as per requirement. The server end application puts a check on the location received of each and every consumer,

and compares it to the actual location. In case of a mismatch, the required alert is generated.

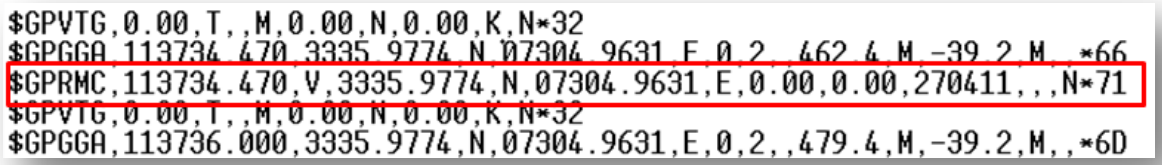

Figure 19: Data received from GPS

Table 3: GPRMC fields and description

	Field	Example	Comments
1	Sentence ID	$GPRMC	Recommended Minimum Specific GNSS Data
2	UTC Time	113734	Hhmmss.sss
3	Status	V	A=Valid, V=Invalid
4	Latitude	3335.9774	dddmm.mmmm
5	N/S Indicator	N	N=North, S=South
6	Longitude	07304.9631	dddmm.mmmm
7	E/W Indicator	E	E=East, W=West
8	Speed over ground	0.00	Knots
9	Course over ground	0.00	Degrees
10	UTC Date	270411	DDMMYY

Use of Global Positioning System in the prototype is an efficient measure that forestalls the meter theft, adding security to the system.

4. Equipment used and circuit modifications

4.1. Hardware

4.1.1. Electricity Meter

The electronic meter being used in the prototype is A283, manufactured by MicroTech Limited. It is an electronic single phase energy meter, developed for domestic and commercial users. The meter generates pulses at the rate of 3200 pulses per KWh.

Meter Components

The core of the meter is TERIDIAN 71M6521DE/FE which is a highly integrated System On a Chip (SOC) with a Microprocessor (MPU) core, Real Time Clock (RTC), FLASH, LCD driver and a variety of I/O pins, shown in Figure 20. Various current sensor technologies are supported including Current Transformers (CT), and Resistive Shunts.

Meter Working

In a typical application, IA and IB are connected to current transformers that sense the current on each phase of the line voltage, whereas VA and VB are typically connected to voltage sensors through resistor dividers. Current transformer and voltage divider step down the current and voltage values, respectively. A single A/D converter digitizes the voltage and current inputs (VA, VB, IA, IB) and forwards the samples to the Computation Engine (CE). The CE performs the precision computations necessary to accurately measure energy. Along with several other computations, it multiplies each current sample with its associated voltage sample to obtain the energy per sample, when multiplied with the constant sample time, followed by product summation. These measurements are then accessed by the MPU, processed further and output using the peripheral devices available to the MPU.

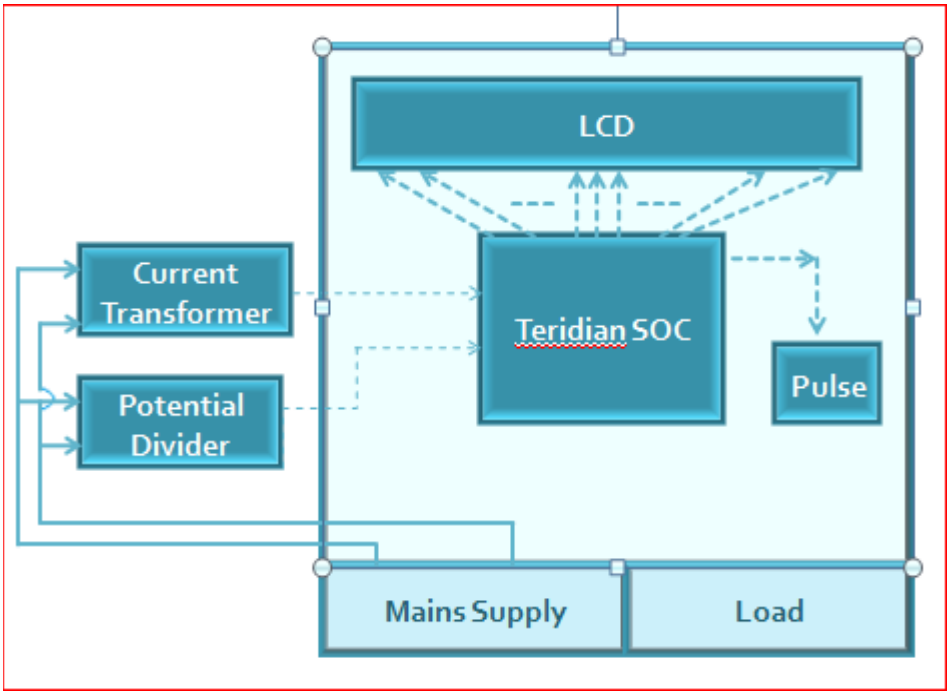

Figure 20: Inside of a single phase Digital Energy Meter

4.1.2. Serial GSM Modem

An off-the-shelf GSM/GPRS modem of Wavecom has been used, developed on the Siemens TC35i GSM module. It is a dual-band module having a transmitting speed of 115kB/s, with a 3V SIM card slot, AC/DC adaptor, RS232 cable, antenna. This modem, shown in Figure 21, is interfaced with the meter interface unit for transmission and reception of SMS over the GSM network.

Figure 21: GSM Modem used initial phases

4.1.3. USB GSM modem

Apache wireless is a GSM/GPRS/EDGE wireless modem. It comes with a built-in antenna and quad-band feature. It can transmit and receive data in high speed upto 460kbps. It has standard USB 2.0 interface and 3V SIM interface. The USB GSM modem is used at the server end of the prototype. It is being employed to receive meter readings being transmitted from all the consumers. Moreover, it is responsible for sending control messages from the server end to the consumers.

4.1.4. GPS Module

The GPS device which serially connected to the computer was based on the module Holux M-89. M-89 is an ultra miniature 25.4 * 25.4 * 3 mm GPS engine board designed by low power consumption MTK GPS solution. M-89 provides 2-wire digital UART port for communication of GPS position data, capable of 4800 to 115200 baud rate. This device, shown in Figure 22, was used to provide the microcontroller with the location of the meter in the form of latitudes and longitudes. It was then replaced by SIM548c module for optimization of the system.

Figure 22: GPS Module used in initial phases

4.1.5. Microcontroller

Microcontroller is the brain of the Meter Interface Unit, an essential component which is responsible for the whole processing being done at the consumer end of the system. It

performs the function of acquiring reading from the meter, bringing it into the requisite format, and then handing it over to the GSM modem. It takes location from GPS module as well. Moreover, it performs the important function of message reception and interpretation, hence infusing control in the system. The basic 89C52 microcontroller was used during the development of the prototype.

AT89C52

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. There are various high-level programming language compilers for the 8051. Several C compilers are available for the 8051, most of which feature extensions that allow the programmer to specify where each variable should be stored. The compiler used in this prototype for 8051 programming is Keil uVision 4.

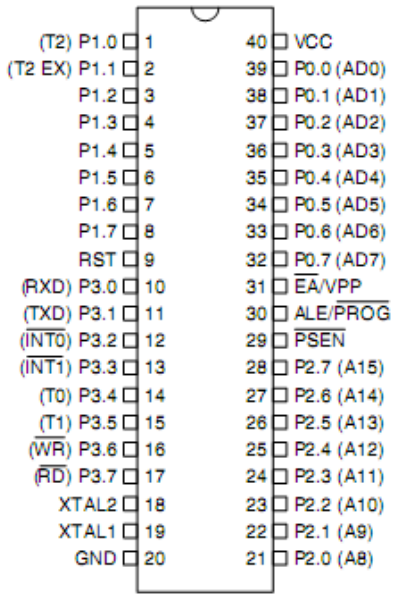

Figure 23: 89c52 Pinouts

The basic pins, depicted in Figure 23, which are used to serve the prototype's requirements are listed below along with their specific purpose,

- Pin 31 & 40 are connected to 5V supply whereas pin 20 is grounded
- XTAL oscillator of 11.0592MHz is connected between pin 18 & 19, along with two stabilizing capacitors of 33pF. The 8051 uses the crystal to synchronize it's operation as it needs an external oscillator that decides it's operating frequency
- A reset circuit is connected to pin 9. Initial charging of the capacitor makes RST high, and blocks DC when it is fully charged
- Pin 10 (RXD) & 11(TXD) are connected to Max232 level converter circuit for serial communication of 8051 with the modules attached (GSM & GPS)
- An LCD has been interfaced with the microcontroller so the energy values, received message, and other relevant notifications can by displayed elaborating the processing of the 8051. The RS, RW, and E pins

of the LCD are connected to 3, 4, 5 pins of 8051, respectively. The data pins DB0-DB7 are connected to pins 21-28 of the uC.

- The pulses from the meter are input to the uC at pin 32
- Pin is used as enable pin for switching of the relay circuit, for remote shutdown and power up of the energy meters.
- Pin is used to switch the multiplexer circuit, to select from GSM and GPS as per requirement of the program

4.1.6. Level Converter

Serial RS-232 communication works with voltage levels, given in Table 4, which are not compatible with the TTL, used in the microcontroller.

Table 4: Voltage Levels of RS232 and TTL

Logic	RS-232	TTL
High	-15… -3V	+2… +5V
Low	+3… +15V	0… +0.8V

To establish serial communication between a microcontroller and a GSM (or GPS) module, a level converter is required to convert RS-232 to TTL, and vice versa. For this purpose, a level converter, consisting of a **Max232** and five **capacitors, has been used, where MAX232** is an integrated circuit that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits.

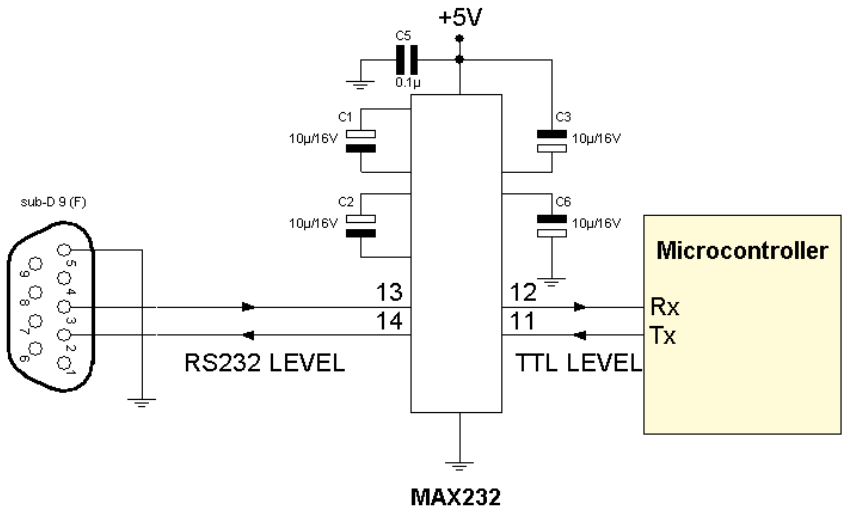

Figure 24: Level Converter Circuit

The level converter circuit, given in Figure 24, was employed when Wavecom GSM modem and a separate GPS module were being interfaced as both were interfaced via RS232. With the use of SIM548c, this need became unnecessary as the module, being TTL compatible, was directly interfaced with the microcontroller.

4.1.7. Multiplexer Circuit

The 89c52 microcontroller being used has only one UART port for serial communication, while the project communicates with two serial devices, GPS and GSM, and so, requires two UART ports. To cater for this requirement, a multiplexer circuit has been included, which can switch between the two devices, GPS and GSM.

MAX333, a quad single-pole-double-throw(SPDT) analog switch has been used for multiplexing data being received from the two devices. One of the four independent switches has been used to fulfill the purpose.

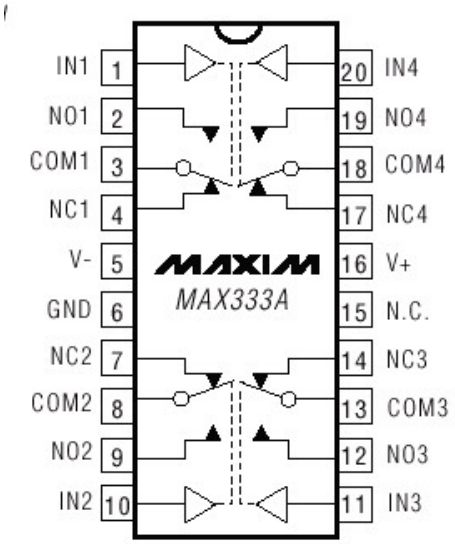

Figure 25: MAX333 pinouts

Pinouts of the MAX333 IC have been given in Figure 25. The TX pin of the GPS module has been connected to the Normally Open (NO) pin, and the GSM modem to the Normally Closed (NC) pin. The COM pin goes to the RX of the microcontroller, while the IN pin is connected to the microcontroller's pin dedicated for multiplexer switching, which acts as the enable pin. In normal operation, the GSM's TX is connected to the RX of the microcontroller as it is on the NC pin. When, data from the GPS module is required, the enable pin coming from the microcontroller needs to be a high signal, which will connect the COM to the NO, that is, the RX of the microcontroller will then be connected with the TX of GPS. Similarly a low signal on the enable pin will activate the GSM modem again.

4.1.8. SIM548c

After thoroughly experimenting with the individual GSM modem and GPS module, the SIM548c GSM/GPRS and GPS module was considered for use. This was a step towards an integrated prototype, which led to a compact and efficient prototype.

SIM548C module is a Quad-Band GSM/GPRS enabled a compact plug-in module that is also equipped with GPS technology for satellite navigation. With a tiny configuration of

Figure 26: SIM548c Module

55mm*33mm*8.2mm, shown in Figure 26, SIM548C offers a compact design, and makes it easy to integrate GSM/GPRS & GPS as an all-in-one solution.

The physical interface to the mobile application is a 60-in board-to-board connector, which provides all hardware interfaces between the module and customers' boards except

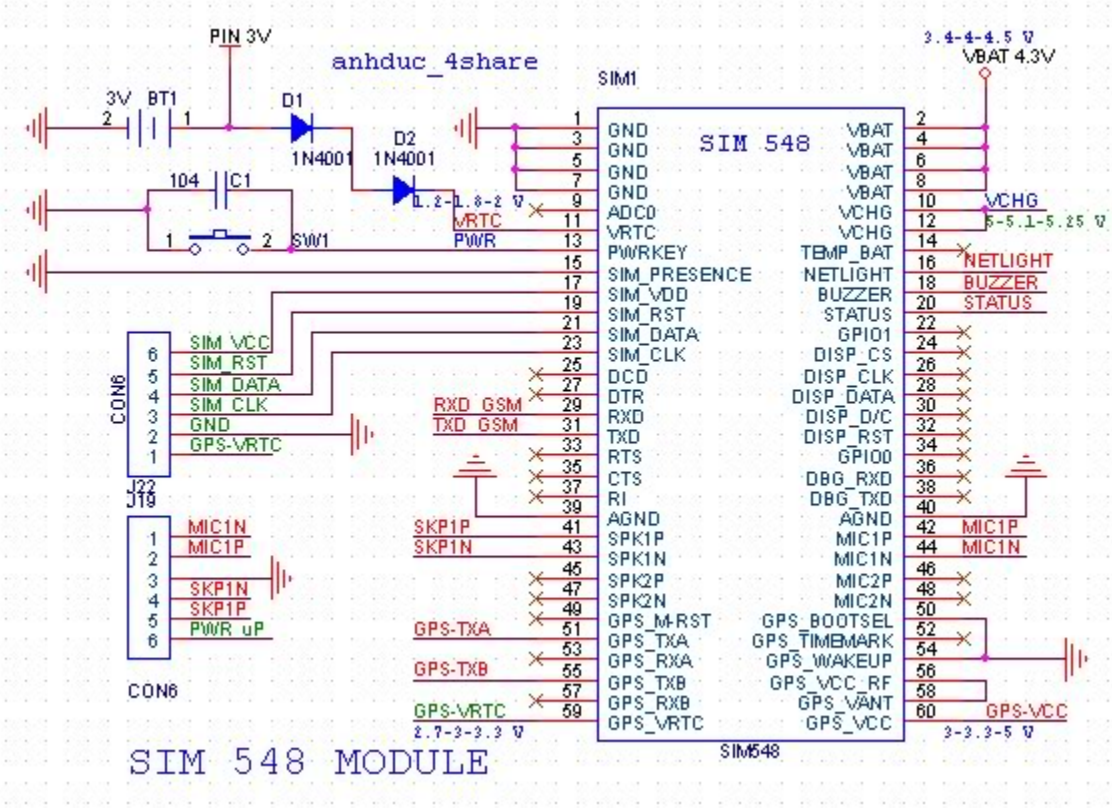

Figure 27: SIM548c Schematic

the RF antenna interface. the GSM and GPS serial ports help in developing applications easily. Two audio channels include two microphones inputs and two speakers' outputs, which can be configured by AT command. Also available for use, is the charge interface, which makes it suitable for battery power application. The Schematic of the SIM548c module to operate it is shown n Figure 27, and the pinouts of the module along with their description, is given in Table 5.

Table 5: Pinouts of SIM548c and their description

Pin Number	Pin Name	Description
2,4,6,8	VBAT	Dedicated to connect the supply voltage, 3.4V…4.5V
11	VRTC	RTC current input from the backup battery when the VBAT is not supplied for the system, 1.2V….2.0V
10,12	VCHG	Voltage input for the charge circuit, 1.1*VBAT….5.25V
60	GPS-VCC	Power supply for GPS whole part, 3V….5.0V
29	GSM RXD	Receive GSM data
31	GSM TXD	Transmit GSM data
51	GPS-TXA	Serial data output for port A
55	GPS-TXB	Serial data output for port B
59	GPS_VRTC	Apply 3V DC for backup RTC and SRAM, 2.7V…3.3V

GPS Application Interface

A GPS receiver with high performance has been integrated to offer GPS full function; it continuously tracks all satellites in view and provides accurate satellite position data.

In the RF section, the GPS signal detected by the antenna is amplified, filtered and converted to an intermediate frequency (IF). An A/D converter converts it to a digital IF signal, which on being received is passed to the baseband section. the on-board processor runs an algorithm that calculates the position, velocity and time. the calculate, termed as navigation solution, can be transformed into the desired coordinate system. The data of the navigation solution is available at the GPS UART. The GPS module has the following specifications,

- Autonomous position accuracy: <10m
- Update rate: Default 1Hz.
- Max. Altitude: <60.000 ft
- Max. Velocity: <1.000 knots
- Protocol: Default NMEA, 4800bps

Antenna Interface

To suit the physical design of individual applications, the module offers alternatives; recommended approach is to connect the antenna via antenna connector on the component side of the PCB, another approach is that of antenna pad and grounding plane placed on the bottom side.

A dual antenna, supporting both GSM and GPS, has been used with the SIM548c used in the project. To obtain excellent GPS reception performance, a good antenna will always be required. The antenna is the most critical item for successful GPS reception in a weak signal environment. Proper choice and placement of the antenna ensures that satellites at all elevations can be seen, and therefore, accurate fix measurements are obtained.

4.2. Softwares Used

4.2.1. PROTEUS

Throughout our project we used Lab Center's PROTEUS to simulate the working of our design before implementing it on hardware. The GSM module and the GPS module were both connected serially to the computer, and the COM port in PROTEUS simulated the working of the computer's Communication port. The data being serially transmitted and received was displayed through the feature of virtual terminal provided in PROTEUS. An LCD is used to display the reading being sent, or the GPS location being calculated, and also to display the SMS received by the GSM modem. The Schematic of the transmission side module used for the purpose of testing in Proteus, is given in Figure 28.

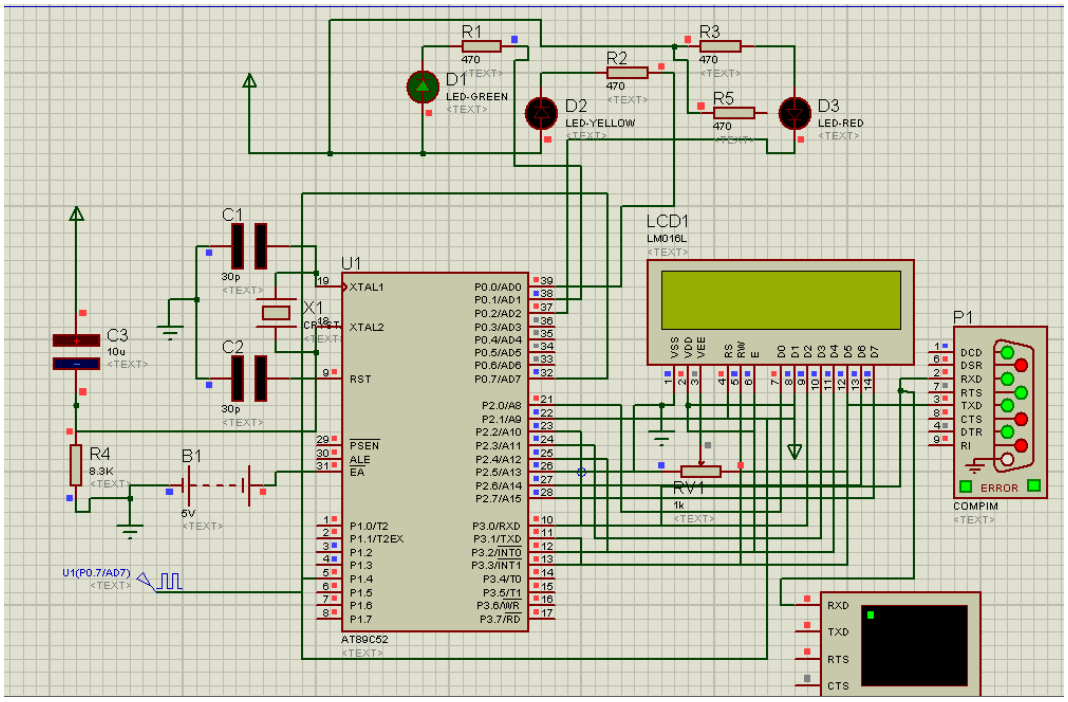

Figure 28: Meter Interface Unit Schematic

4.2.2. Keil uVision 4

The programming of the microcontroller 8051 has extensively been done in C compiler Keil uVision 4. The code of each module was individually constructed. Once all the modules were completed, they were integrated together, giving a complete form to the transmission side of the prototype.

4.2.3. Hyper terminal

Hyperterminal is a program that is designed to perform the functions of communication and terminal emulation. It makes use of serial ports and the controls associated with external devices. These devices can vary and include such options as radio communications equipment, robots, and instruments used for scientific measurements and similar endeavors. The connections provided by Hyperterminal make it easy to

retrieve data from these sources, as well as be able to execute commands to the devices from the main computer system.

GSM modem was tested using hyper terminal and various AT commands were explored and experimented with, as shown in Figure 29, in order to check if the modem's settings were configured correctly or if your modem is connected properly,. The hexa-equivalents for various AT commands were also reviewed and then incorporated into the microcontroller code.

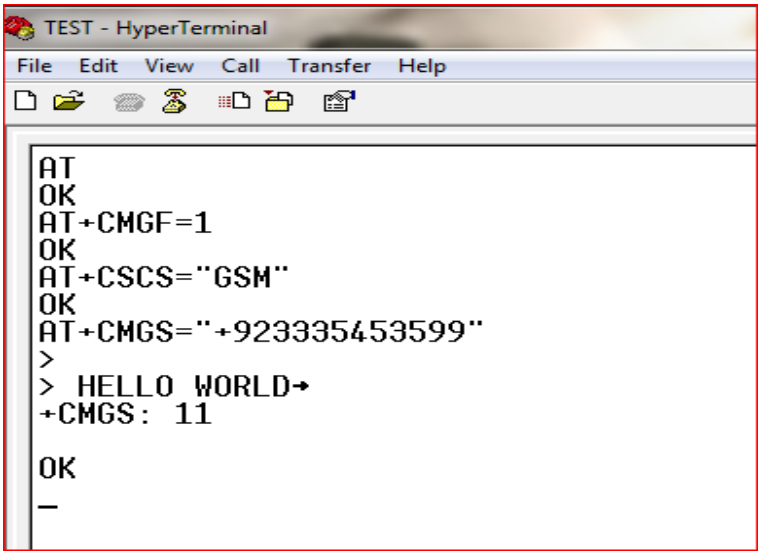

Figure 29: Testing of GSM Modem in Hyperterminal

Moreover, the GPS module was also tested using the hyperterminal. It also helped in the analysis of the data being received by the GPS, displayed in Figure 30, which was then used to ascertain ways of sorting out the relevant data.

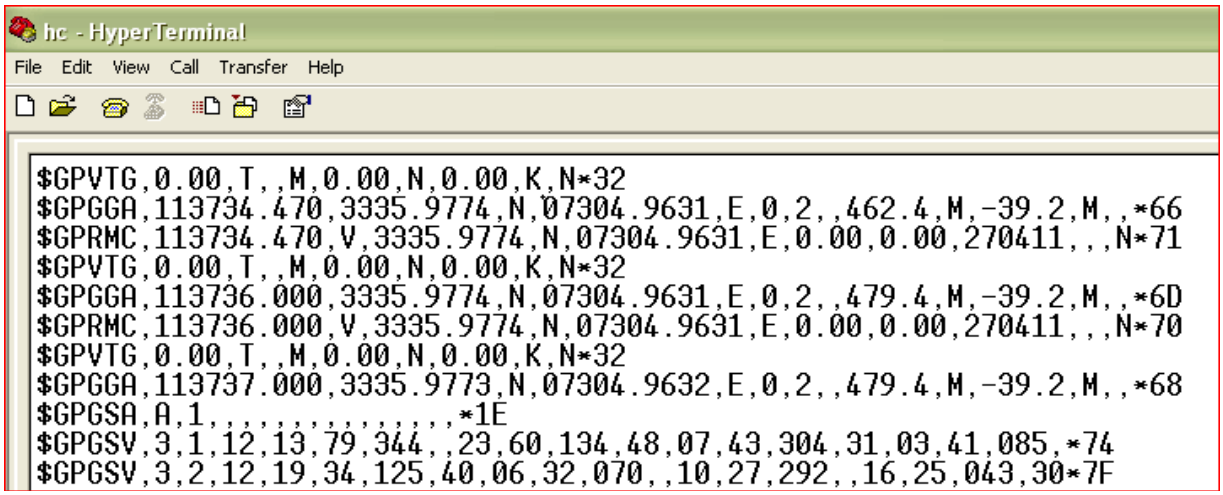

Figure 30: Testing of GPS Module in Hyperterminal

4.2.4. Apache

The USB GSM Modem is configured using the Apache Wireless Modem software. The software came with the device, having a number of features that can be used with the modem, including sending and receiving messages, making calls, etc. The only feature used in the project has been that of initial configuration. The function of SMS transmission and reception is being performed by the application designed in .NET, to be used at the server end.

4.2.5. MySQL

MySQL client version: 5.0.22, which is an open source relational database management system was used to maintain the record for all consumers and to maintain their consumption history. The database was developed using phpMyAdmin version 3.2.4 which is a free Web-based front end widely installed by Web hosts worldwide, since it is developed in PHP. Various tables in the database are accessed by the ASP.Net service or PHP code when required using standard SQL queries of

· **Insert**-to add consumer's reading to the "consumption" table

- **Update**-to update user record from “users”

- **Delete**- to delete user record from “users” table

- **Select**- to retrieve data from either table

4.2.6. Visual Studio 2010, C# .NET

A windows Form application was developed in Visual Studio, C#.Net. C# was chosen because of its versatility and its importance in today’s research environment. A code was formulated to receive SMS from the communication port of the designated server system. The database maintained in MySQL was linked to the . The link to the database was formed using the .dll library mysql.data. A limitation here is that the connector for MySQL database and .Net was only available for 32-bit systems. This problem can however be resolved if the supplier company purchases the licensed Microsoft SQL Server instead of the open source MySQL. A service was created in ASP.Net and placed at the MCS server. The C#.Net code extracts the meter reading and the consumer’s credentials from the message and calls the service. The service is then used to actually submit the information into the MySQL database.

Visual Studio was not only used for database update but also for checking for meter theft. The ASP.NET service also queries the users table in the MySQL database “billing” for latitude longitude values, compares the retrieved values with those received in the SMS, and submits the result of comparison back to the C#.Net Windows Application. In case of mismatch, the application automatically sends an alert of “Meter Theft” to the consumer.

4.2.7. PHP

PHP (Hypertext Preprocessor), version 5.3.1.0., was chosen as the scripting language to create the web pages for the web interface that is the website. The website may be accessed by the consumer with user rights, or by the supplier with administrator rights. Illustrative user profiles were generated and information was displayed when desired using the data from the MySQL database. PHP code, embedded into the HTML source document used standard SQL queries to attain data from the database once a connection was established with the local host at the web server.

5. ANALYSIS

In the project a working prototype of a remote energy monitoring, profiling and control system was built with efficient utilization of the existing digital meters and the already installed GSM infrastructure.

With the advent of digital technology, analog electromechanical meters are continuously being replaced by digital meters. Digital energy meters offer greater convenience to implement and establish automatic meter reading system electronically. Various AMR methods and technologies using Power Line Carrier(PLC) communications, Supervisory Control and Data Acquisition(SCADA), telephone modem, Internet, Ethernet, Embedded RF module, Wi-Fi, Bluetooth and ZigBee were established and developed to provide and demonstrate the solution of efficiency, reliability and effectiveness of AMR. All the methods are either too expensive to implement and operate, require complex setup of infrastructure, short operating distance and still require field intervention of human operators or prone to error and have reliability issues due to noise in the transmission line or weather conditions. The developed project utilizes the wide spread and already installed infrastructure of GSM network. The store and forwarding features of SMS allow reliable meter reading delivery when the GSM signal is affected by poor weather conditions.

5.1. Simple and cost effective Solution

The prototype makes efficient usage of the ageless yet simple 8052 microcontroller which is an easy to use, low cost and versatile controller. A single microcontroller is utilized as a meter interface unit and also formulates the messages containing the meter reading via the transmission module. Synchronization in pulse counting is not required

with the digital meter since the readings are transmitted in the form of pulse counts and not energy readings in Kilo-Watt hours. In case of a power failure when the microcontroller restarts it sends a startup message to the server which processes the received pulse counts accordingly so that there is no information loss.

5.2. Location Monitoring

Though it wasn't a part of our initial objectives, as an additional feature a GPS module was also incorporated in the prototype so that an immediate low level alert to the consumer could be generated in case the position of the meter was changed indicating meter theft.

5.3. No data loss

Since there are a huge number of consumers, the prototype modules have been programmed to transmit their readings at different instances in time so that no messages are lost due to network congestion at the server end. The retrieval of all individual meter readings through SMS may take some time depending on the GSM network traffic and weather conditions for the particular GSM cell area. If a message is not delivered to the server end, it is not a problem since the module sends the cumulative pulse count since the meter was turned on, not the pulse count between successive messages.

5.4. Compact Design

After experimenting with various modules, the compact plug-in module, SIM548C was used in the final prototype that is equipped with GPS technology for satellite navigation along with Quad-Band GSM/GPRS functionality. The compact design of the prototype makes it easy to integrate into the existing digital meter casing. This lays off the cost for redesigning the existing meters and saves both time and cost for the integration of additional hardware components.

5.5. Automated System

At the server end the SIM number is used to identify and retrieve the consumer details from the received messages so the overhead of assigning, transmitting and maintaining unique UID's is obviated. The process of extracting useful data and uploading it to the database is completely automated and requires no human intervention. This also removes the possibility of human error which is a major cause of discrepancy in the meter reading process.

5.6. Efficient Monitoring

An efficient mechanism for monitoring of energy consumption has been ensured with detailed consumption information in the form of a dynamically uploaded "consumption" table in database, as shown in Figure 31.

			SIM	entryTime	units
☐	✎	✕	+923229808188	2011-04-28 09:31:01	44
☐	✎	✕	+923215817813	2011-04-26 14:43:33	56
☐	✎	✕	+923459206484	2011-04-28 09:27:24	12
☐	✎	✕	+923459206484	2011-04-27 14:38:19	23
☐	✎	✕	+923459206484	2011-04-27 14:44:38	76
☐	✎	✕	+923229808188	2011-04-28 09:27:27	20
☐	✎	✕	+923215817813	2011-04-19 14:43:08	24
☐	✎	✕	+923335453599	2011-05-18 21:46:58	100
☐	✎	✕	+923335453599	2011-03-30 21:45:23	55
☐	✎	✕	+923335453599	2011-03-23 21:44:58	22

Figure 31: Database Consumption Table

The user database is also maintained in the form of a separate data table, shown in Figure 32, which may be edited easily using the administrative command of update.

+923335453599	Rafia Malik	rafia	12345	rafia@rems.mcs.edu.pk	House 987, Street ABC, Sector 456, City XYZ	33D34	73D03	2011-01-20 21:43:23	2011-07-17 17:34:13	2011-03-21 00:00:00	4217
+923215817813	Sulaiman Sadiq	sulaiman	1234	sulaiman@rems.mcs.edu.pk	House 456, Street ABC, Sector 456, City XYZ	34D34	73D03	2011-03-17 00:00:00	2011-07-03 14:13:21	2011-03-22 09:07:00	395
+923459206484	Sanam Rashid	sanam	12345	sanam@rems.mcs.edu.pk	House 567, Street ABC, Sector 456, City XYZ			2011-03-17 00:00:00	2011-07-17 01:47:32	2011-03-22 08:00:00	305
+923229808188	Anam Ajmal	anam	12345	anam@rems.mcs.edu.pk	House 789, Street ABC, Sector 456, City XYZ	35D34	73D03	2011-01-20 22:40:51	2011-07-17 17:06:20	2011-03-22 06:00:30	745
+923215148523	User	user	12345	user@rems.mcs.edu.pk	House XYZ, Street 123,			2011-06-23 00:00:00	2011-07-17 17:41:32	*NULL*	0

Figure 32: Users' Database Table

5.7. Ubiquitous Web Interface

An illustrative and interactive web interface not only makes accessing user, consumption and billing information more easier, it also makes it ubiquitous in the sense that both the consumer as well as the supplier may access their desired information from anywhere and at any time. The data maintained is both elaborate and current, such that it is updated in real time with every received SMS, containing the energy readings, at the server.

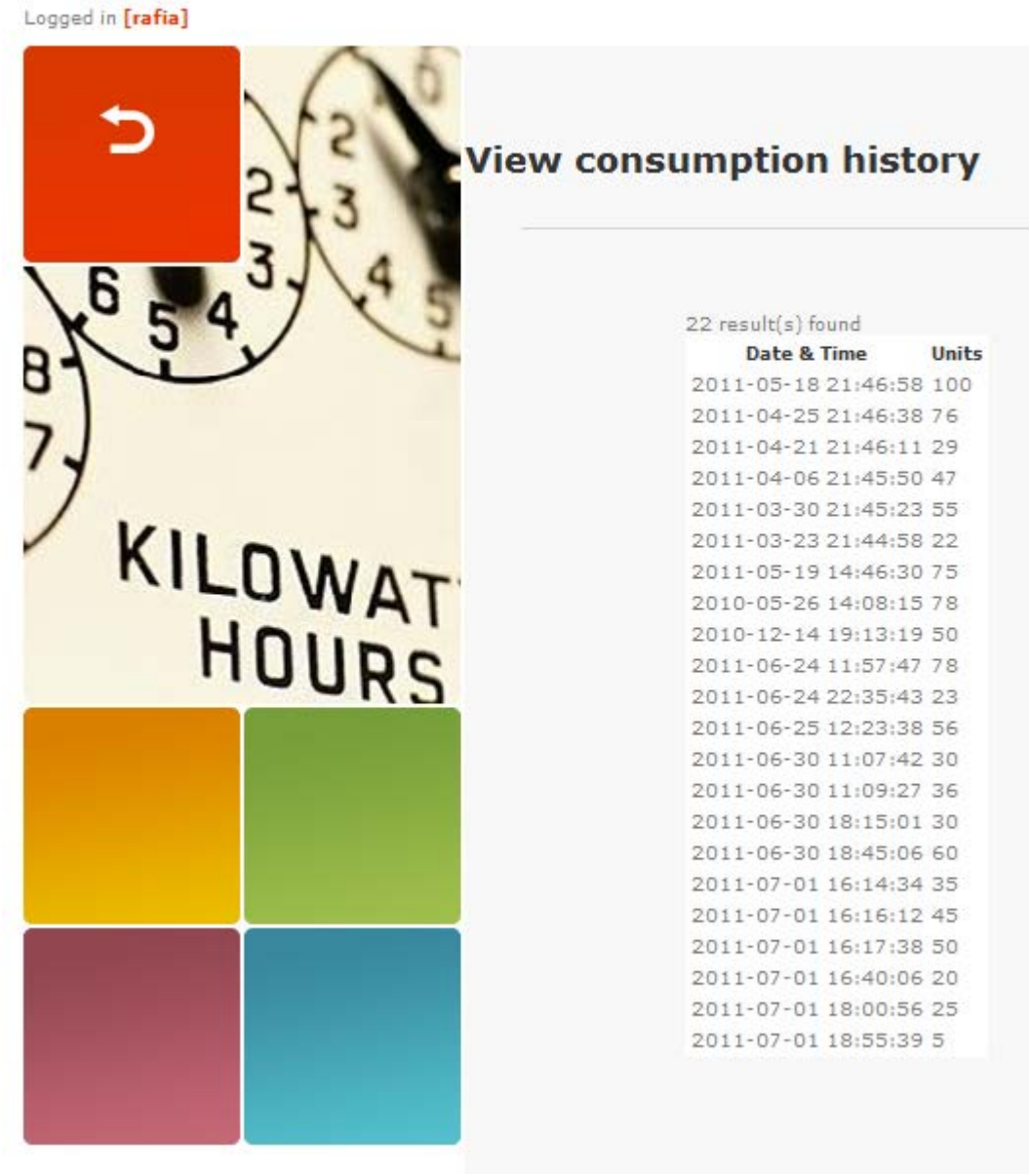

Figure 33: Consumption History Page

The user may view his consumption history in tabular form, as shown in Figure 33, or in the form of an energy usage curve for any specified duration of time, as in Figure 34. Vivid data points, that is exact Date and time value for any consumption may also be viewed.

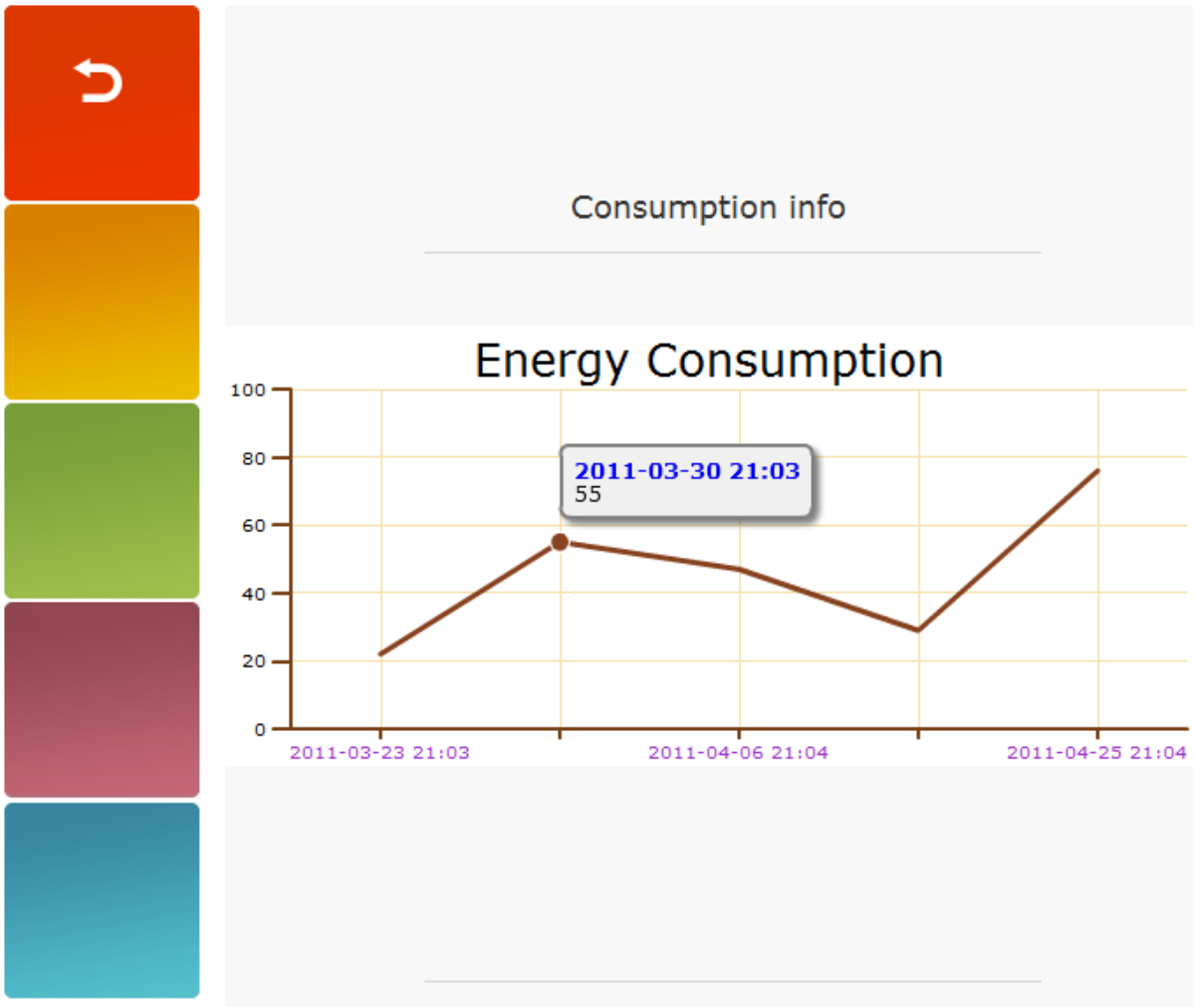

Figure 34: Energy Usage Curve (x-axis – units, y-axis – date/time)

5.8. Transparent Billing

The system is developed with an inherent transparency feature, such that the consumer can access his billing information at any instance of time and be satisfied with the elaborate records being maintained. Bill amount for various consumption values may be examined by the consumer in the form of a demonstrative billing information curve. This also removes the need of any hassle for later verification of bills. The user is also shown his bill so far, calculated starting from his last payment, displayed in Figure 35.

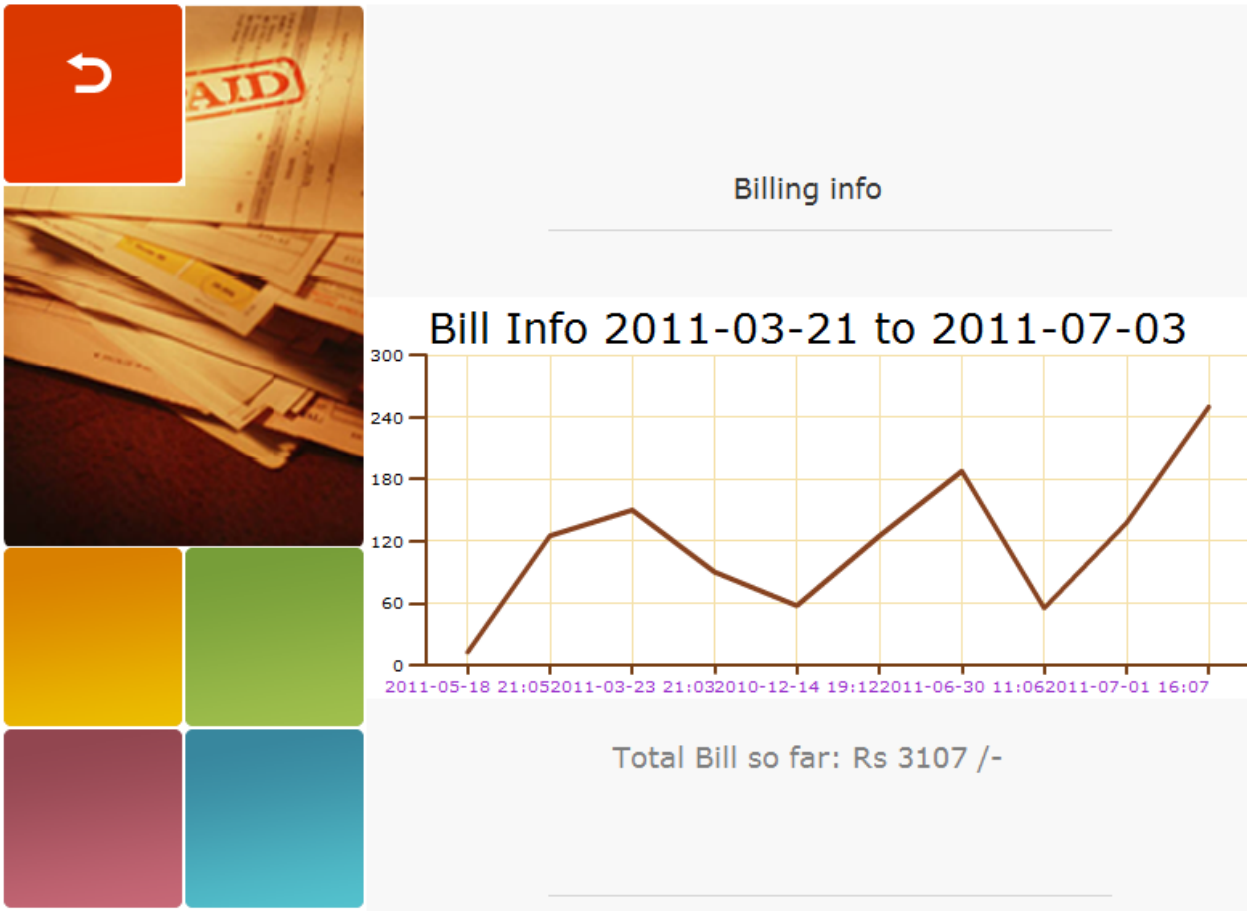

Figure 35: Billing vs. Consumption (x-axis – units, y-axis – date/time)

5.9. Elaborate Profiling

User profiles are maintained in the form of consumption trends, such that the administrator (supplier) has the option of viewing both the yearly cumulative consumption of all users in a monthly breakdown fashion in the form of a Bar graph, an example of which has been depicted in Figure 36, or the supplier may view an individual consumer's energy usage for a particular month, as can be seen from Figure 37.

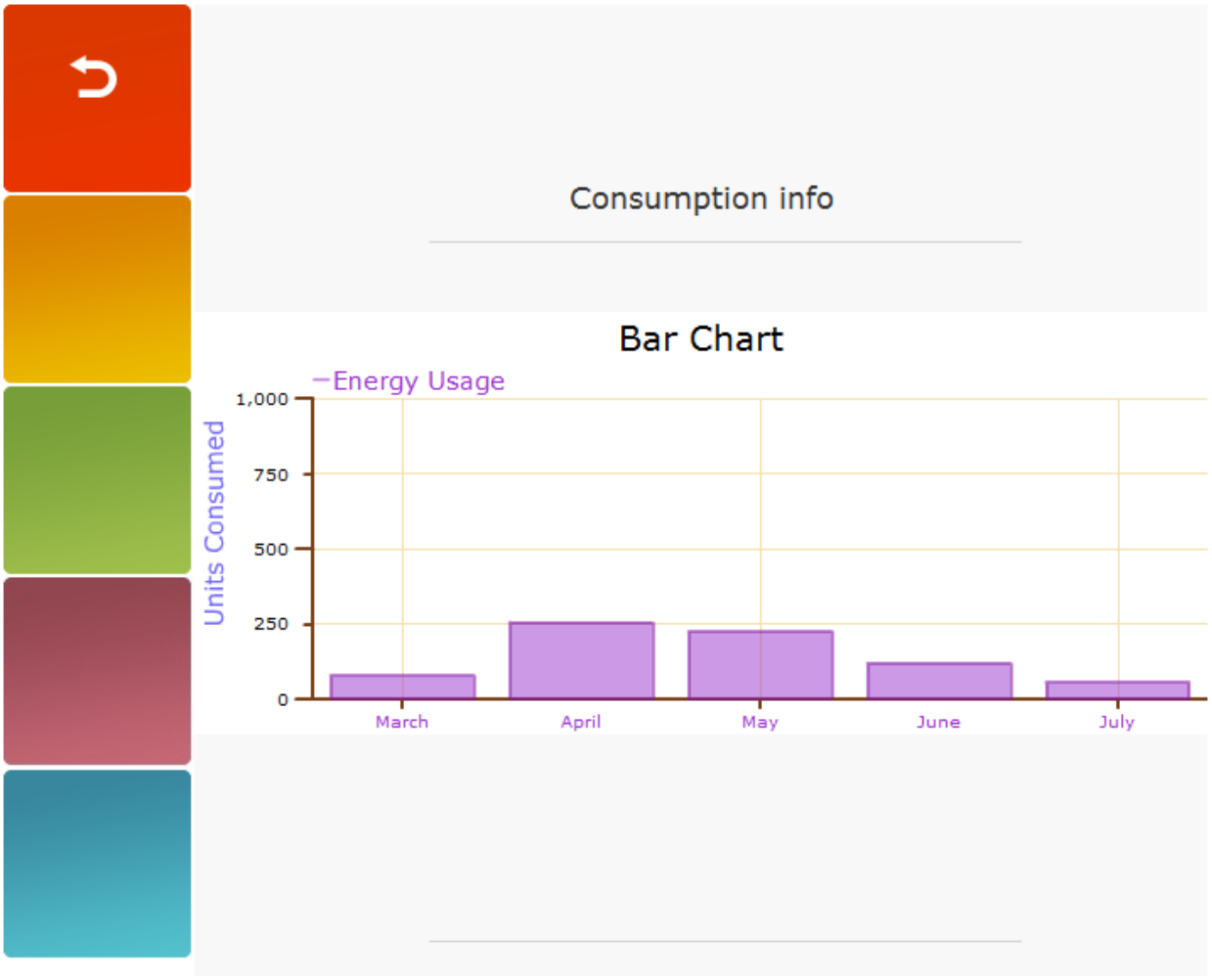

Figure 36: Yearly Consumption (x-axis – units, y-axis – month)

The energy curve for any month is plotted for the specified user for the current year.

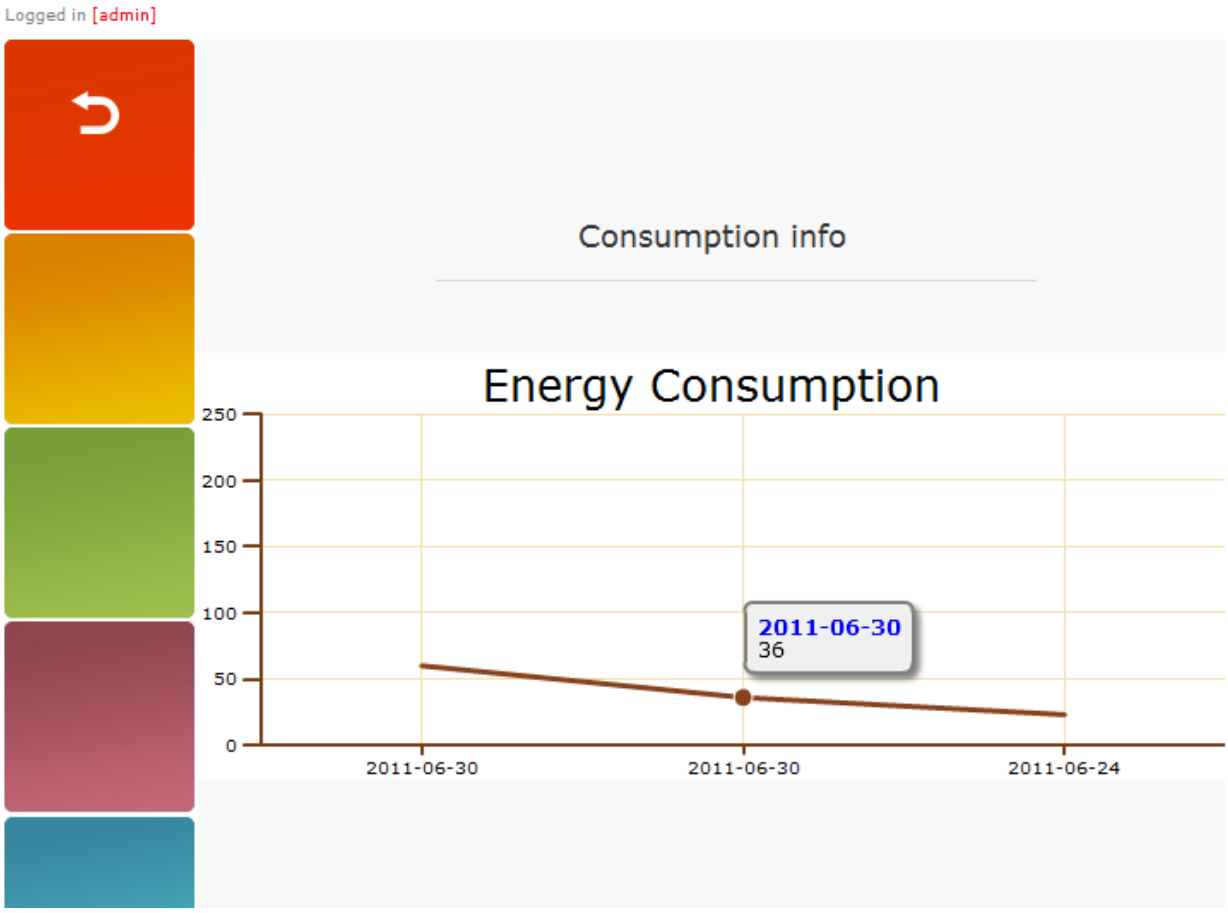

Figure 37: Monthly Consumption for single user (x-axis – units, y-axis – date/time)

6. Future Work

This project was taken up with the idea to automate the energy metering system, which would result in efficiency in the way consumption of electricity is recorded, and the corresponding billing of the energy supply network. It aims at providing some relief from the current energy crisis, because the implementation of such a metering system would result in transparency. Moreover, it provides a means of keeping the users aware of their energy consumption, which might help in inculcating in them a sense to restrict their use in case they go overboard. The future work in this field would require making the system more proficient by adding more features, as required by the electricity supply companies. The constraints of the system need to be targeted and overcome, so the system is capable enough to be implemented commercially.

6.1. Constraints

The objective of developing a remote energy monitoring and profiling system has been successfully achieved, along with a number of additional features that add to the efficiency of the system. Nevertheless, the devised prototype has some constraints, which if overcome would lead to even better and improved performance by the system.

6.1.1. Underuse of Electronic Meter Processing

The approach employed to acquire reading from the meter, does not make full use of the digital energy meters installed. The electrical meter, A283, being used in the prototype, displays following information,

- Meter Serial Number
- kWh(Total)

- Last Month MDI
- Current Month MDI
- Active Power(kW)

The meter interface unit only attains the number of consumed units, i.e the number of kWh utilized, from the meter, without making use of the other data being output by the meter's processor, hence, not making optimum use of the advanced processor of the digital electrical meters.

6.1.2. Currently for residential users

The profiling system devised checks for aberrant behavior in a user's consumption pattern, by comparing it to the general usage pattern of a residential household. This makes the system residential user specific, as it does not have the ability to discriminate commercial usage from residential usage, for now. So, commercial consumption, being many times higher than that of a normal household, would automatically generate alerts of over consumption, even in case of normal consumption.

6.1.3. Tradeoff between reading interval and users served

The interval after which meter readings are transmitted, depends on the number of houses to which are served by one server. In a crowded area, with more number of consumers, the interval after which a meter sends it's meter reading, will become large, to accommodate other users as well. But if the supply company requires shorter intervals between consecutive meter readings from users, it will then have to target lesser number of houses, which would easily fit in that short interval. Hence, a tradeoff is required

between the length of interval, after which readings are transmitted, to the number of users that can be served by a server.

6.2. Improvements

Owing to the constraints present in the prototype, the future work would require improvements in the prototype that would overcome the flaws, that might be faced when the system is considered for mass implementation. These improvements will include addition of more features that would add to the efficiency of the system.

6.2.1. Optimum use of meter processing

The underuse of meter processing needs to be countered, by introducing such an approach of meter reading acquisition that would collect not only the meter reading in kWh, but also other useful information like the meter serial number, current and previous month MDI, etc. this approach would prove beneficial, and would lead to optimum use of the advanced meter processing, available these days in the electronic energy meters.

6.2.2. Tampering detection

An effective tampering detection system can also be incorporated in the already devised prototype. This can be achieved if the reading from the substation point, from which the energy is being branched out to the consumers, can be attained. The readings from all the users, who are served by that substation, are summed and compared to the amount of energy supplied. Keeping the losses in mind, if energy supplied is greater than the energy billed, tampering will be detected, alerting the Admin.

6.2.3. Probabilistic Profiles of Consumers

An improvement in the profiling system can be in the form of forming probabilistic profiles of consumers. For now, the consumption of users is compared to a general usage trend. But this approach has the tendency to be improved, because different consumers have different user trends, and comparing their consumption pattern to one standard trend might not be very effective. For better results, user profiles can be formed by theory of probability, which can be used to predict the behavior of a user in terms of energy consumption. Comparison for detection of aberrant behavior can then be done according to these probabilistic profiles.

6.2.4. Detailed consumption distribution

The user has access to view his consumption history on the website, once he logs into his account. An even better feature would be to equip the consumers to view detailed consumption of their households. This can be done if all the phases in a user's household are monitored, such and their consumption records made available to the user, such that the user can check which phase has the maximum load. This would enable the user to log into his account, and view the consumption distribution in his house, remotely.

6.2.5. Multiple SIMs for MIU

Mobile Networks have the tendency to get down for some time, which can be due to a number of unforeseen reasons. Such a collapse in the system can be overcome by the use of multiple SIMs, from different mobile service provider companies. So that when the network of one mobile service provider company is down, the MIU can switch to another SIM, so that no loss of readings occur at the server end.

6.2.6. System Expansion

The monitoring and profiling system can be expanded to include the consumption record of both, water and gas meters, as well. Only the means of acquiring reading meter has to be modified a bit so it could work for water and gas meter. The electomechanical meters would have to be interfaced with some pulse generating mechanism, such that the frequency of pulses generated is in specific proportion to the resource consumed. This would be interfaced to the meter interface unit of the prototype, and the remaining system would remain the same.[1]

6.2.7. Profiling standard for commercial users

The constraint mentioned that the profiling standard is residential user specific can be overcome by introduction of probabilistic profiles, or by introducing a separate standard for commercial users. This would require a way of sifting commercial consumer data from residential user data, and treating them accordingly.

6.2.8. Sending Meter Readings via GPRS

Another improvement in the system can be to use GPRS as the transmission medium instead of GSM. It would not require changes in the hardware as the SIM548c module supports GSM as well as GPRS. The module needs to be configured for the GPRS support. A key advantage of GPRS over GSM is that GPRS has a higher data transmission speed. If SMS over GPRS is used, an SMS transmission speed of about 30 SMS messages per minute may be achieved.

6.3. Commercial Implementation

The project developed is a prototype, which can be implemented commercially, after making some required modifications to make the system fit for mass implementation. These considerations have been listed below, which would be encountered in the process of effectuation.

6.3.1. Compatibility with analog meters

A major percentage of the electric meters installed, is that of analog meters. For mass implementation, the MIU needs to be made compatible with the analog meters as well, which would require finding a way by which the MIU can acquire readings from these meters. The main difference is the addition of some device to generate pulses relating to the amount of consumption monitored, that translates to the actual reading on the meter dials for the case of the electro-mechanical meter

6.3.2. Database Expansion

With the expansion of the system, the need to define more fields would arise. Separate databases for commercial and residential users would be required, so that different standards and thresholds can be maintained for both. Moreover, there would be a need to define timezones for users from different regions. Hence, an even more elaborate database would be suffice.

7. Conclusion

The project paves the way to automatic metering of electricity consumption. A complete working prototype of Remote Energy Monitoring and Profiling System has been built to demonstrate an automatic power meter reading system using GSM network. It takes the advantage of existing GSM infrastructure that has virtually full coverage of all housing and building area. The prototype includes simple and easy installation of meter interface units, which need to be attached to the already installed energy meters, and the profiling system would require trained personnel to monitor the servers. The prototype developed, provides effective, reliable and efficient wireless automatic power reading, billing, and notification by automating the metering system, through the use of GSM network. The additional features equip the system with the capability to perform many other different functions that are beneficial to both the electricity supplier as well as the electricity user. Not forgetting the system devised is not only confined to the electricity field but can also be extended to be used with water meters as well as gas meters or even a combination of these three different meters.

8. Bibliography

[1] Tan S. Y., M. Moghawemi, "PIC-BASED AUTOMATIC METER READING AND CONTROL OVER THE LOW VOLTAGE DISTRIBUTION NETWORK", 2002

[2] P.K. LEE and L.L. LAI, "A PRACTICAL APPROACH OF SMART METERING IN REMOTE MONITORING OF RENEWABLE ENERGY APPLICATIONS"

[3] Sasan Ardalan, and Arnold Bragg (2005) Remote Access to Electronic Meters Using the Short Message Service, **United States Patent US6900737**

[4] Arun Sehgal, Itron Inc. Spokane, Washington, "AMR offers multiple Benefits"

[5] H.G. Rodney Tan, IEEE, C.H.Lee and V.H. Mok, IEEE, "Automatic Power Meter Reading Using GSM Network", published in The 8th International Power Engineering Conference (IPEC 2007)

[6] Mohit Arora Freescale Semiconductor, "Prevent hacking, tampering in energy metres"

9. Glossary

Automatic Meter Reading (AMR)

The technology of automatically collecting consumption, diagnostic, and status data from water meter orenergy metering devices (gas, electric) and transferring that data to a central database for billing, troubleshooting, and analyzing

Energy Profiling System (EPS)

A system that records and analyses the electricity consumption of users

Global Positioning System (GPS)

A space-based global navigation satellite system (GNSS) that provides location and time information in all weather, anywhere on or near the Earth

Global System for Mobiles (GSM)

A standard set developed by the European Telecommunications Standards Institute (ETSI) to describe technologies for second generation (or "2G") digital cellular networks

MySQL

It is a relational database management system (RDBMS)[2] that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language.

Hypertext Preprocessor (PHP)

A general-purpose scripting language originally designed for web development to produce dynamic web pages.

Telemetry

Automatic transmission and measurement of data from remote sources by wire, radio or other means

System On a Chip (SOC)

It refers to integrating all components of a computer or ther electronic system into a single integrated circuit (chip)

Short Message Service (SMS)

Text communication service component of mobile communication systems, using standardized communications protocols that allow the exchange of short text messages between fixed line or mobile phone devices

Encryption

The activity of converting data or information into a code, to make it unreadable for unintended receivers

Real time system

A real-time system is one that processes information and produce a response within a specified time

Polling

To trigger a response from a client or a terminal

Remote control

A feature that enables server to wirelessly operate or manipulate the working of a device

Electrical grid

An electrical grid is an interconnected network for delivering electricity from suppliers to consumers.

Signal-to-noise ratio

Ratio that compares the level of a desired signal to the level of background noise

Attenuation

A general term that refers to any reduction in the strength of a signal

routing protocols

Radio Frequency (RF)

The mode of communication for wireless technology

Intermediate frequency (IF)

The frequency to which a carrier frequency is shifted as an intermediate step in transmission or reception

Power Line Communications (PLC)

The use of existing electrical cables to transport data

Wi-Fi

A wireless standard for connecting electronic devices

Transistor-Transistor Logic (TTL) **levels**

Voltage levels have been 5.0 volts, with a high being any voltage above about 3.5 volts and a low being any voltage below about 1.5 volts

Interrupt service routine (ISR)

A software routine that hardware invokes in response to an interrupt

Serial Data Buffer (SBUF)

A buffer that stores the data that is to be transmitted, or data that has been received serially

NMEA

A combined electrical and data specification for communication between marine electronic devices such as echo sounder, GPS receivers and many other types of instruments. It has been defined by, and is controlled by, the U.S.-based National Marine Electronics Association

APPENDIX A

CODES

MICROCONTROLLER (AT89c52) CODE IN C (Keil uvision 4.14.16.0 Compiler)

//For calculation and transmission of meter readings, GPS coordinates and for receiving commands from server//

```
#include<reg52.h>
#include<stdio.h>
#include<stdlib.h>
#define port2 P2

sbit rs = P1^2;
sbit rw = P1^3;
sbit e = P1^4;
sbit GLED=P0^2;
sbit YLED=P0^1;
sbit RLED=P0^0;
sbit shut=P0^4;
sbit INPUT=P0^7;
sbit IN=P0^6;
char bufr[16];
char info[32];
char poll[4]=("UPD");
char test[3]={"A,3"};
char tes[2]={",,"};
unsigned int check=0,i,chec=0,chk=0;
unsigned int counter=0;
unsigned int energy=19;

unsigned int c3=25;
int recv=0;
int rxcount=0;


//DELAY FUNCTION
void delay(unsigned int msec)
{
int i,j ;
for(i=0;i<msec;i++)
for(j=0;j<1275;j++);
}
```

```
// LCD COMMAND SENDING FUNCTION
void lcd_cmd(unsigned char item)
{
port2 = item;
rs= 0;
rw=0;
e=1;
delay(1);
e=0;
return;
}

// LCD DATA SENDING FUNCTION
void lcd_data(unsigned char item)
{
port2 = item;
rs= 1;
rw=0;
e=1;
delay(1);
e=0;
return;
}

// LCD STRING SENDING FUNCTION
void lcd_string(unsigned char *str)
{

int i=0;
lcd_cmd(0x01);
while(str[i]!='\0')
{
lcd_data(str[i]);
i++;
delay(10);
}
return;
}

// SERIAL PORT SETTING
void serial()
{
TMOD=0x20;    //MODE=2
TH1=0xfa;      // 4800 BAUD
SCON=0x50  ;  // SERIAL MODE 1 ,8- BIT DATA ,1 STOP BIT ,1 START BIT ,
RECEIVING ON
```

```
TR1=1;          //TIMER START
}

// FUNCTION TO TRANSMIT DATA THROUGH SERIAL PORT
void transmit_data(unsigned char str)       {
SBUF=str;                           //Store data in SBUF
while(TI==0);
TI=0;
}


//FUNCTION DISPLAYS A NULL-TERMINATED STRING ON THE RS232 PORT
void send_serial(unsigned char *s)
{
delay(50);
while(*s!=0x0)
{
SBUF=*s;
while(TI==0);
{
}
TI=0;
s++;
}
}

//FUNCTION FOR ENERGY TRANSMISSION
void intchar()              // Function to display character on LCD
{
int plac = energy%10000;
sprintf(bufr, "%d", plac);
lcd_string(bufr);
if(energy==10)
GLED=0;
if(energy==20)
YLED=0;
if(energy==30)
RLED=0;
delay(30);
GLED=1;
YLED=1;
RLED=1;

send_serial("AT+CMGS=\"+923229808188\"\r");//server number here
delay(70);
if(chk==0)
```

```
send_serial("S");
send_serial(bufr);
transmit_data(0x1A);
delay(100);
chk++;

return;
}


//FUNCTION TO DISPLAY MESSAGE ON LCD

void lcd_msg()
{
c3=25;
lcd_cmd(0x01);              //Clear LCD display
lcd_cmd(0x84);
delay(120);         //Move cursor to position 6 of line 1
lcd_string("message");        //Showing message
lcd_cmd(0xC0);              //Beginning of second line
lcd_data(info[c3+1]);
lcd_data(info[c3+2]);
lcd_data(info[c3+3]);
delay(70);
}

//FUNCTION FOR REMOTE CONTROL
void control()
{
IN=0;
IE=0x00;
send_serial("AT\r");
send_serial("AT+CMGF=1\r");
delay(50);
chec=0;
IE=0x90;
send_serial("AT+CMGR=1\r");
lcd_msg();                          //Showing message
IE=0x00;

//////////////////////////////////////////////////////
if(info[c3+1]=='1' && info[c3+2]=='0')
GLED=0;
if(info[c3+1]=='2' && info[c3+2]=='0')
YLED=0;
if(info[c3+1]=='3' && info[c3+2]=='0')
```

```
RLED=0;
delay(50);
GLED=YLED=RLED=1;
delay(50);
if(info[c3+1]=='O' && info[c3+2]=='f'&& info[c3+3]=='f')
{
shut=0;
delay(70);
recv=1;
}
if(info[c3+1]=='U' && info[c3+2]=='P'&& info[c3+3]=='D')
{
lcd_string("POLL OK");
send_serial("AT+CMGD=1\r");
delay(50);

intchar();
}
if(info[c3+1]=='O' && info[c3+2]=='n')
{
shut=1;
delay(80);
recv=0;
}
chec=0;
delay(20);
send_serial("AT+CMGD=1\r");
delay(50);
IE=0x90;

//////////////////////////////////////////////////////////

}
void receive_data()          interrupt 4
{
if(IN==1)
{
info[check++]=SBUF;      //Read SBUF
if(check<4)
{
if(info[check-1]!=test[check-1])
check=0;
}
}
else if(IN==0)
{
```

```
info[chec++]=SBUF;      //Read SBUF
if(chec<3)
{
if(info[chec-1]!=tes[chec-1])
chec=0;
}

}
RI=0;
}

void lcd_shape()              //Shape of degree symbol
{
lcd_cmd(64);
lcd_data(10);
lcd_data(17);
lcd_data(17);
lcd_data(10);
lcd_data(0);
lcd_data(0);
lcd_data(0);
lcd_data(0);
}

void lcd_latitude()
{
unsigned int c3=1;
lcd_shape();
lcd_cmd(0x01);               //Clear LCD display
lcd_cmd(0x84);               //Move cursor to position 6 of line 1
lcd_string("LATITUDE");        //Showing latitude
lcd_cmd(0xC0);               //Beginning of second line
lcd_data(info[c3+1]);
lcd_data(info[c3+2]);
lcd_data(0);
lcd_data(info[c3+3]);
lcd_data(info[c3+4]);
lcd_data(info[c3+5]);
lcd_data(info[c3+6]);
lcd_data(info[c3+7]);
lcd_data(info[c3+8]);
lcd_data(info[c3+9]);
lcd_data(0x27);            //ASCII of minute sign(')
lcd_data(info[c3+10]);
lcd_data(info[c3+11]);
delay(250);
```

```
}

void lcd_longitude()
{
unsigned int c5=13;
lcd_cmd(0x01);              //Clear LCD display
lcd_cmd(0x84);              //Move cursor to position 4 of line 1
lcd_string("LONGITUDE");        //Showing longitude
lcd_cmd(0xC0);              //Beginning of second line
lcd_data(info[c5+1]);
lcd_data(info[c5+2]);
lcd_data(info[c5+3]);
lcd_data(0);
lcd_data(info[c5+4]);
lcd_data(info[c5+5]);
lcd_data(info[c5+6]);
lcd_data(info[c5+7]);
lcd_data(info[c5+8]);
lcd_data(info[c5+9]);
lcd_data(info[c5+10]);
lcd_data(0x27);             //ASCII of minute sign(')
lcd_data(info[c5+11]);
lcd_data(info[c5+12]);
delay(250);
}

void compare()
{
IE=0x00;
lcd_latitude();
lcd_longitude();
check=0;
c3=1;
send_serial("AT\r");
send_serial("AT+CMGF=1\r");
send_serial("AT+CMGS=\"+923229808188\"\r");//server number here
delay(70);

send_serial("LAT\:");

transmit_data(info[c3+1]);
transmit_data(info[c3+2]);
transmit_data('D');
transmit_data(info[c3+3]);
transmit_data(info[c3+4]);
transmit_data(info[c3+5]);
```

```
transmit_data(info[c3+6]);
transmit_data(info[c3+7]);
transmit_data(info[c3+8]);
transmit_data(info[c3+9]);
transmit_data(0x27);

transmit_data(info[c3+11]);
send_serial("\r\nLONG\:");
c3=13;
transmit_data(info[c3+2]);
transmit_data(info[c3+3]);
transmit_data('D');
transmit_data(info[c3+4]);
transmit_data(info[c3+5]);
transmit_data(info[c3+6]);
transmit_data(info[c3+7]);
transmit_data(info[c3+8]);
transmit_data(info[c3+9]);
transmit_data(info[c3+10]);
transmit_data(0x27);
transmit_data(info[c3+12]);
transmit_data(0x1A);
delay(200);
lcd_cmd(0x01);                          //Clear LCD display
lcd_cmd(0x80);          //Move cursor to position 6 of line 1
lcd_string("Message Sent");
}



//MAIN PROGRAM
void main()
{       int plac;
serial();
lcd_cmd(0x38);      //2 LINE, 5X7 MATRIX
lcd_cmd(0x0e);    //DISPLAY ON, CURSOR BLINKING
lcd_string("program running");
IE=0x00;

send_serial("AT\r");
send_serial("AT+CMGF=1\r"); // SET TEXT MODE

IN=0;
IE=0x90;

while(1)
```

```
{
info[c3+1]=info[c3+2]=0;

if (INPUT==1)
{

RLED=0;
counter=counter+1;

delay(70);
RLED=1;
if (counter==2)
{
counter=0;
lcd_cmd(0x01);
energy=energy+1;

plac = energy%10000;
sprintf(bufr, "%d", plac);
lcd_string(bufr);

if(energy%10==0 && counter==0)
intchar();
delay(20);
if(energy%5==0 && counter==0)
control();
}
}
if(energy%7==0 && counter==0)
{         IN=1;
if(check==35)
{
compare();
IN=0;
IE=0x00;
}
}
rxcount++;
if(recv==1 && rxcount%30000==0)
{
rxcount=0;
control();
}
}
delay(100);
}
```

WINDOWS FORM APPLICATION (Visual Studio C#.Net 2010)

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;
using MySql.Data.MySqlClient;
using System.IO;
using System.Text;
using TestSMS.pk.edu.mcs.rems;
using System.Text.RegularExpressions;
using System.Diagnostics;

namespace TestSMS
{
        /// <summary>
        /// Summary description for Form1.
        /// </summary>
        public class Form1 : System.Windows.Forms.Form
  {
                private System.Windows.Forms.Label label1;
                private System.Windows.Forms.TextBox textBox1;
                private System.Windows.Forms.Button button1;
                private System.Windows.Forms.CheckBox checkBox1;
                private System.Windows.Forms.Label label2;
     private System.Windows.Forms.Button button2;
                private System.Windows.Forms.TextBox textBox2;
     private System.Windows.Forms.Label label3;
     private TextBox textBox3;
     private Emant.SMS sms1;
                private System.ComponentModel.IContainer components;



                public Form1()
                {
                        //
                        // Required for Windows Form Designer support
                        //
                        InitializeComponent();

                        //
                        // TODO: Add any constructor code after InitializeComponent call
                        //
                }
```

```
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        protected override void Dispose( bool disposing )
        {
                if( disposing )
                {
                        if (components != null)
                        {
                                components.Dispose();
                        }
                }
                base.Dispose( disposing );
        }

        #region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
this.components = new System.ComponentModel.Container();
this.label1 = new System.Windows.Forms.Label();
this.textBox1 = new System.Windows.Forms.TextBox();
this.button1 = new System.Windows.Forms.Button();
this.checkBox1 = new System.Windows.Forms.CheckBox();
this.label2 = new System.Windows.Forms.Label();
this.button2 = new System.Windows.Forms.Button();
this.textBox2 = new System.Windows.Forms.TextBox();
this.label3 = new System.Windows.Forms.Label();
this.textBox3 = new System.Windows.Forms.TextBox();
this.sms1 = new Emant.SMS(this.components);
this.SuspendLayout();
//
// label1
//
this.label1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
this.label1.Location = new System.Drawing.Point(16, 144);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(228, 108);
this.label1.TabIndex = 1;
this.label1.Click += new System.EventHandler(this.label1_Click);
//
// textBox1
```

```
            //
            this.textBox1.Location = new System.Drawing.Point(16, 96);
            this.textBox1.Name = "textBox1";
            this.textBox1.Size = new System.Drawing.Size(160, 20);
            this.textBox1.TabIndex = 2;
            this.textBox1.Text = "Test Message";
            //
            // button1
            //
            this.button1.Location = new System.Drawing.Point(184, 96);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(75, 23);
            this.button1.TabIndex = 3;
            this.button1.Text = "Send";
            this.button1.Click += new System.EventHandler(this.button1_Click);
            //
            // checkBox1
            //
            this.checkBox1.Location = new System.Drawing.Point(104, 32);
            this.checkBox1.Name = "checkBox1";
            this.checkBox1.Size = new System.Drawing.Size(120, 24);
            this.checkBox1.TabIndex = 4;
            this.checkBox1.Text = "Connect to Phone";
            this.checkBox1.CheckedChanged += new
System.EventHandler(this.checkBox1_CheckedChanged);
            //
            // label2
            //
            this.label2.Location = new System.Drawing.Point(16, 8);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(100, 16);
            this.label2.TabIndex = 5;
            this.label2.Text = "Comm Port No";
            this.label2.Click += new System.EventHandler(this.label2_Click);
            //
            // button2
            //
            this.button2.Location = new System.Drawing.Point(24, 272);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(75, 23);
            this.button2.TabIndex = 6;
            this.button2.Text = "Query";
            this.button2.Click += new System.EventHandler(this.button2_Click);
            //
            // textBox2
            //
```

```
            this.textBox2.Location = new System.Drawing.Point(16, 64);
            this.textBox2.Name = "textBox2";
            this.textBox2.Size = new System.Drawing.Size(100, 20);
            this.textBox2.TabIndex = 7;
            this.textBox2.Text = "+923214567890";
            //
            // label3
            //
            this.label3.Location = new System.Drawing.Point(128, 64);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(128, 23);
            this.label3.TabIndex = 8;
            this.label3.Text = "Destination Phone no";
            this.label3.Click += new System.EventHandler(this.label3_Click);
            //
            // textBox3
            //
            this.textBox3.Location = new System.Drawing.Point(16, 34);
            this.textBox3.Name = "textBox3";
            this.textBox3.Size = new System.Drawing.Size(62, 20);
            this.textBox3.TabIndex = 9;
            this.textBox3.Text = "COM13";
            this.textBox3.TextChanged += new
System.EventHandler(this.textBox3_TextChanged);
            //
            // sms1
            //
            this.sms1.sresult = null;
            this.sms1.onReceived += new System.EventHandler(this.sms1_onReceived);
            //
            // Form1
            //
            this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
            this.ClientSize = new System.Drawing.Size(272, 302);
            this.Controls.Add(this.textBox3);
            this.Controls.Add(this.label3);
            this.Controls.Add(this.textBox2);
            this.Controls.Add(this.button2);
            this.Controls.Add(this.checkBox1);
            this.Controls.Add(this.button1);
            this.Controls.Add(this.textBox1);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.label2);
            this.Name = "Form1";
            this.Text = "Form1";
            this.Load += new System.EventHandler(this.Form1_Load);
```

```
        this.ResumeLayout(false);
        this.PerformLayout();

                }
                #endregion

                /// <summary>
                /// The main entry point for the application.
                /// </summary>
                [STAThread]
                static void Main()
                {
                        Application.Run(new Form1());
                }

                private void checkBox1_CheckedChanged(object sender,
System.EventArgs e)
                {
                        if (checkBox1.Checked)
                                sms1.Open(textBox3.Text);
                        else
                                sms1.Close();
                }

                private void button2_Click(object sender, System.EventArgs e)
                {
                sms1.Query();
                }


                private void button1_Click(object sender, System.EventArgs e)
                {
                        sms1.Send(textBox1.Text, textBox2.Text);
                }

      private void sms1_onReceived(object sender, EventArgs e)
      {
        Control.CheckForIllegalCrossThreadCalls = false;

        label1.Text = sms1.sresult;
        if (sms1.sresult.Trim().Equals("Modem Initialised OK")) { return; }

        //int a = 0;

        string line;
        string senderNumber = null;
```

```
        string time = null;
        string units = null;
        string sLat = null;
        string sLong = null;

        using (StringReader sr = new StringReader(sms1.sresult.Trim()))
        {
            string sHeaderPattern = "^\\+CMT: \"(?<sender>\\+[0-9]*)\",\"\",\"(?<time>[0-
9,:/]*)\\+[0-9]*\"$";
            string sConsumptionPattern = "^(?<units>[S]{0,1}[0-9]*)$";
            string sLocationPattern = "^LAT:(?<lat>[0-9]{2}D[0-9]{2})\\.[0-
9]{4}'N$|^LONG:(?<long>[0-9]{2}D[0-9]{2})\\.[0-9]{4}'E$";
            string sPattern = string.Format("{0}|{1}|{2}", sHeaderPattern,
sConsumptionPattern, sLocationPattern);

            Regex re = new Regex(sPattern);

            while ((line = sr.ReadLine()) != null)
            {
                Match m = re.Match(line);

                if (!m.Success) { return; }

                if (m.Groups["sender"].Success && m.Groups["time"].Success)
                {
                    senderNumber = m.Groups["sender"].Value;
                    time = m.Groups["time"].Value;
                }
                else if (m.Groups["units"].Success)
                {
                    units = m.Groups["units"].Value;
                }
                else if(m.Groups["lat"].Success)
                {
                    sLat = m.Groups["lat"].Value;
                }
                else if (m.Groups["long"].Success)
                {
                    sLong = m.Groups["long"].Value;
                }
            }
        }

        if (!string.IsNullOrEmpty(senderNumber) && !string.IsNullOrEmpty(time))
        {
```

```
            DateTime dtm = DateTime.ParseExact(time, "yy/MM/dd,HH:mm:ss",
System.Globalization.CultureInfo.CurrentCulture);
            time = dtm.ToString("yyyy-MM-dd HH:mm:ss");
         }
         else { return; }


         if (!string.IsNullOrEmpty(sLat) && !string.IsNullOrEmpty(sLong))
         {
            ProcessLocation(senderNumber, sLat, sLong);
         }
         else if (!string.IsNullOrEmpty(units))
         {
            ProcessConsumption(senderNumber, time, units);
         }

      }

      private void ProcessConsumption(string sim, string entrytime, string units)
      {
         int intunits = -1;
         if (units.StartsWith("S", StringComparison.CurrentCultureIgnoreCase)) { intunits
= int.Parse(units.Substring(1)); }
         else { intunits = Convert.ToInt32(units); }
         RemsService service = new RemsService();
         if (intunits >= 40)
         {
            string num = service.RefNo(sim);
            textBox2.Text = num;
            textBox1.Text = "Over-Usage";
            sms1.Send(textBox1.Text, textBox2.Text);
         }

         if (intunits >= 65)
         {
            sms1.Send("Over-Usage", "+923155118243");
         }

         bool success = service.SubmitConsumption(sim, entrytime, units);



      }

      private void ProcessLocation(string sim, string locLat, string locLong)
      {
```

```
            RemsService service = new RemsService();
            // (-1) Error, (0) Success-No Change, (1) Success-Location Changed
            int iRet = service.CompareLatLong(sim, locLat, locLong);
            if (iRet == 1)
            {
                string num = service.RefNo(sim);
                textBox2.Text = num;
                textBox1.Text = "Meter Theft";
                sms1.Send(textBox1.Text, textBox2.Text);
            }
        }



        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void textBox3_TextChanged(object sender, EventArgs e)
        {

        }
            }
}
```

WEB SERVICE (Visual Studio ASP.Net 2010)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using MySql.Data;
using MySql.Data.MySqlClient;
using System.Data;

namespace REMS
{
        /// <summary>
        /// Summary description for Service1
        /// </summary>
        [WebService(Namespace = "http://tempuri.org/")]
        [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
        [System.ComponentModel.ToolboxItem(false)]
        // To allow this Web Service to be called from script, using ASP.NET AJAX,
uncomment the following line.
        // [System.Web.Script.Services.ScriptService]
        public class RemsService : System.Web.Services.WebService
        {

                [WebMethod]
                public bool SubmitConsumption(string sim, string entryTime, string units)
                {
                        if (units.StartsWith("S",
StringComparison.InvariantCultureIgnoreCase))
                        {
                                return AddConsumption(sim, entryTime, units.Substring(1,
units.Length - 1));
                        }

                        string sPrevUnits = GetLastReading(sim);
                        if (string.IsNullOrEmpty(sPrevUnits)) { return false; }

                        int iPrevUnits = int.Parse(sPrevUnits);
                        int iUnits = int.Parse(units);
                        int iNewUnits = iUnits - iPrevUnits;

                        return AddConsumption(sim, entryTime, iNewUnits.ToString());

                }

                private bool AddConsumption(string sim, string entryTime, string units)
```

```
        {
            try
            {
                using (MySqlConnection conn = new
MySqlConnection("server=localhost;database=billing;UID=rsas;Password=sasr"))
                {
                    string sql = string.Format("Insert into
consumption(SIM, entryTime, units) VALUES('{0}', '{1}', '{2}');",
                        sim, entryTime, units);
                    using (MySqlCommand cmd = new
MySqlCommand(sql, conn))
                    {
                        conn.Open();
                        int iRet = cmd.ExecuteNonQuery();
                        conn.Close();
                        return true;
                    }
                }
            }
            catch
            {
                return false;
            }
        }

        [WebMethod]
        public int CompareLatLong(string sim, string lat, string lng)
        {
            DataRow row = GetLatLong(sim);
            if (null == row) { return -1; }

            string sLat = row["LAT"] as string;
            string sLong = row["LONG"] as string;

            if (string.IsNullOrEmpty(sLat) && string.IsNullOrEmpty(sLong))
            {
                if (AddLatLong(sim, lat, lng)) { return 0; }
                else { return -1; }
            }

            if (sLat.Equals(lat) && sLong.Equals(lng)) { return 0; }
            else { return 1; }
        }

    public string RefNo(string sim)
    {
```

```
        DataRow row = GetRefNo(sim);

        string refno = row["RefNo"] as string;

        return refno;

    }

            private bool AddLatLong(string sim, string lat, string lng)
            {
                    using (MySqlConnection conn = new
MySqlConnection("server=localhost;database=billing;UID=rsas;Password=sasr"))
                    {
                            string sql = string.Format("UPDATE users SET LAT =
'{0}', `LONG` = '{1}' WHERE SIM = '{2}';", lat, lng, sim);
                            using (MySqlCommand cmd = new MySqlCommand(sql,
conn))
                            {
                                    try
                                    {
                                            conn.Open();
                                            int iRet = cmd.ExecuteNonQuery();
                                            conn.Close();
                                            return true;
                                    }
                                    catch
                                    {
                                            return false;
                                    }
                            }
                    }
            }

            private DataRow GetLatLong(string sim)
            {
                    using (MySqlConnection conn = new
MySqlConnection("server=localhost;database=billing;UID=rsas;Password=sasr"))
                    {
                            string sql = string.Format("Select * from users WHERE
SIM = '{0}';", sim);
                            using (MySqlCommand cmd = new MySqlCommand(sql,
conn))
                            {
                                    using (MySqlDataAdapter daSql = new
MySqlDataAdapter(cmd))
                                    {
```

```
                                        using (DataTable table = new
DataTable("consumption"))
                                        {
                                                daSql.Fill(table);
                                                if (table.Rows.Count > 0)
                                                {
                                                        return table.Rows[0];
                                                }
                                                else
                                                {
                                                        return null;
                                                }
                                        }
                                }
                        }
                }
        }

    public DataRow GetRefNo(string sim)
    {
       using (MySqlConnection conn = new
MySqlConnection("server=localhost;database=billing;UID=rsas;Password=sasr"))
        {
           string sql = string.Format("Select * from users WHERE SIM = '{0}';", sim);
           using (MySqlCommand cmd = new MySqlCommand(sql, conn))
           {
              using (MySqlDataAdapter daSql = new MySqlDataAdapter(cmd))
              {
                 using (DataTable table = new DataTable("consumption"))
                 {
                    daSql.Fill(table);
                    if (table.Rows.Count > 0)
                    {
                       return table.Rows[0];
                    }
                    else
                    {
                       return null;
                    }
                 }
              }
           }
        }
    }

                private string GetLastReading(string sim)
```

```
                {
                        using (MySqlConnection conn = new
MySqlConnection("server=localhost;database=billing;UID=rsas;Password=sasr"))
                        {
                                string sql = string.Format("Select * from consumption
WHERE SIM = '{0}' ORDER BY `entryTime` DESC LIMIT 1;", sim);
                                using (MySqlCommand cmd = new MySqlCommand(sql,
conn))
                                {
                                        using (MySqlDataAdapter daSql = new
MySqlDataAdapter(cmd))
                                        {
                                                using (DataTable table = new
DataTable("consumption"))
                                                {
                                                        daSql.Fill(table);
                                                        if (table.Rows.Count > 0)
                                                        {
                                                                return
table.Rows[0]["units"].ToString();
                                                        }
                                                        else
                                                        {
                                                                return null;
                                                        }
                                                }
                                        }
                                }
                        }
                }
        }
}
```

WEB INTERFACES (PHP 5.3.1.0)

Administrator: View Consumption Page

```
<?php
session_start();

if(!isset($_SESSION['user'])){
	header("location:login.php");
}
else {

include('mysql_conn.php');
$SIM = $_POST['user'];
$YEAR = $_POST['year'];

		$sql1 = "SELECT * FROM billing.users WHERE SIM = '".$SIM."'";
	$result1 = mysql_query($sql1,$sql_conn) or die(mysql_error());
	while ($fh = mysql_fetch_array($result1))
	{
		$user = $fh['SIM'];

	}

	if (isset($_POST['Submit'])){

	if ($user['0']=="")
	{
	$error = "Please enter correct SIM";
	echo $error;
	}
	else if($user['0']!="")
	{
		$_SESSION['simcon'] = $SIM;
		$_SESSION['month'] = $_REQUEST['month'];
		$_SESSION['year']="";
		header("location:mon_user_graph.php");
	}

	}
	else if (isset($_POST['submitt'])){
		if ($YEAR<=date('Y') && $YEAR!="")
	{
		$_SESSION['year'] =$YEAR;
		$_SESSION['simcon'] ="";
		$_SESSION['month'] ="";
		header("location:mon_user_graph.php");
```

```
        }
        else if ($YEAR=="" || $YEAR>date('Y'))
{
        echo ("Enter Correct Year! ");
        //header ("")
}
        }
?>
```

Administrator: View Users

```
<?php
session_start();
if(!isset($_SESSION['user'])){
        header("location:login.php");
}
else {

include('mysql_conn.php');

$search_data = "";
$search_by = "";

if (isset($_POST['submitted'])){

        $search_by = $_POST['searchby'];
        $search_data = $_POST['searchdata'];
        $sql = "SELECT * FROM billing.users WHERE $search_by LIKE
'%".$search_data."%'";

        $result = mysql_query($sql,$sql_conn) or die(mysql_error());
}

$result2 = "";
$num_rows2 = "";

?>
```

Administrator: Manage

```
<?php
session_start();
if(!isset($_SESSION['user'])){
        header("location:login.php");
}
```

```
else {

include('mysql_conn.php');

$search_data = "";
$search_by = "";

if (isset($_POST['submitted'])){

        $search_by = $_POST['searchby'];
        $search_data = $_POST['searchdata'];
        $sql = "SELECT * FROM billing.users WHERE $search_by LIKE
'%".$search_data."%'";

        $result = mysql_query($sql,$sql_conn) or die(mysql_error());
}

$result2 = "";
$num_rows2 = "";

// Display data
if (isset($_POST['submitted'])){        $num_rows = mysql_num_rows($result);
echo "$num_rows result(s) found <br />";

if ($num_rows>0){echo "<table bgcolor=\"#DFD8D0\" cellpadding=\"5\">";

echo "<tr align=\"left\"> <th>Username</th> <th>Full Name</th>  <th>Action</th>
</tr>";
while($row = mysql_fetch_array($result)){echo "<tr bgcolor=\"#FFF\"><td>";
echo $row['username'];echo "</td><td>";echo $row['fullname'];     echo "</td><td>";
echo " | <a href=\"delete.php?SIM=".$row['SIM']."\"><img src=\"images/del.jpg\"></a>"
;
echo " | <a href=\"update.php?SIM=".$row['SIM']."\"><img
src=\"images/update.jpg\"></a>" ;
echo " | <a href=\"view.php?SIM=".$row['SIM']."\"><img
src=\"images/view.jpg\"></a>" ;
echo "</td></tr>";
} // endwhile
echo "</table>";
}
} // end if isset if
else{
$sql_all = "SELECT * FROM billing.users";$result_all =
mysql_query($sql_all,$sql_conn) or die(mysql_error());

$num_rows_all = mysql_num_rows($result_all);
```

```
echo "$num_rows_all record(s) in database <br />";


if ($num_rows_all>0){

echo "<table bgcolor=\"#DFD8D0\" cellpadding=\"5\">";

echo "<tr align=\"left\"> <th>Username</th> <th>Full Name</th>  <th>Action</th>
</tr>";
while($row = mysql_fetch_array($result_all)){
echo "<tr bgcolor=\"#FFF\"><td>";
echo $row['username'];
echo "</td><td>";
echo $row['fullname'];
echo "</td><td>";
echo " | <a href=\"delete.php?SIM=".$row['SIM']."\"><img src=\"images/del.jpg\"></a>"
;
echo " | <a href=\"update.php?SIM=".$row['SIM']."\"><img
src=\"images/update.jpg\"></a>" ;
echo " | <a href=\"view.php?SIM=".$row['SIM']."\"><img
src=\"images/view.jpg\"></a>" ;
echo "</td></tr>";
} // endwhile
echo "</table>";
}

if(isset($_SESSION['mssg_del'])){
echo $_SESSION['mssg_del'];
$_SESSION['mssg_del'] = "";
}
}

?>
```

Consumer: Home

```
<?php
error_reporting(0);
session_start();
if(!isset($_SESSION['user'])){
        header("location:login.php");
}
else{

include('mysql_conn.php');

$username = $_SESSION['user'];

$sql = "SELECT * FROM billing.users WHERE username='".$username."'";

$result = mysql_query($sql,$sql_conn) or die(mysql_error());

$record = mysql_fetch_array($result);
$SIM = $record['SIM'];
$username = $record['username'];
$fullname = $record['fullname'];
$address = $record['address'];
$password = $record['password'];
$email = $record['email'];
$joindate = $record['joinDate'];
$lastlogin = $record['lastLogin'];
$lastpayment = $record['lastPayment'];

$ul=date('Y-m-d H:i:s');

$sql2 = "SELECT * FROM billing.consumption WHERE SIM='".$SIM."' AND
entryTime between '".$lastpayment."' and '".$ul."'";
$result2 = mysql_query($sql2,$sql_conn) or die(mysql_error());
$todaysum = 0;
while($sql2 = mysql_fetch_assoc( $result2 ) ) {
$us[] = $sql2['units'];
$dt[] = $sql2['entryTime'];
$f=strtotime("".$sql2['entryTime']."");
$f=date('Y-m-d',$f);

if ($f== date('Y-m-d'))
{
        $today = $sql2['units'];
}
        else
```

```
{
        $today = 0;
}
$todaysum=$todaysum+$today;
}

$d=date('m');
$usage = array_sum($us);

include('billcalc.php');
$sql3 = mysql_query("UPDATE billing.users SET billtodate = '$bill' WHERE
username='$username'");

?>
```

Consumer: Billing

```
<?php
session_start();
$con = mysql_connect("localhost","rsas","sasr");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
$auser = $_SESSION['user'];
$sim = $_SESSION['SIM'];

$result1 = mysql_query("SELECT * FROM billing.users WHERE SIM = '".$sim."'");
while($sql2 = mysql_fetch_assoc( $result1) ) {
$lp[] = $sql2['lastPayment'];
}

$ul=date('Y-m-d H:m:s');
$x=array();

$result = mysql_query("SELECT * FROM billing.consumption WHERE SIM =
'".$sim."' AND entryTime between '".$lastpayment."' and '".$ul."'");
while($sql = mysql_fetch_assoc( $result) ) {
$us[] = $sql['units'];
$dt[] = $sql['entryTime'];
$f=strtotime("".$sql['entryTime']."");
$x[]=date('Y-m-d H:m',$f);
}
$max=count($us)-1;
$labs=array();
```

```
for ($i=$max; $i>=0; $i--)
{
        $usage=$us["$i"];
include('billcalc.php');
$labs["$i"]=$bill;

}

$ll = strtotime("".$lp["0"]."");
$ll= date('Y-m-d',$ll);

$ul=date('Y-m-d');


// use the chart class to build the chart:
include_once( 'ofc-library/open-flash-chart.php' );

$g = new graph();

// Bill information
$g->title('Bill Info '. $ll. ' to '.$ul, '{font-size: 26px;}');

$g->set_data($labs);
// label each point with its value
$g->set_x_labels($x);

$g->set_x_label_style( 10, '#9933CC', 0, 2 );

// set the Y max
$g->set_y_max( 300 );
// label every 20 (0,20,40,60)
$g->y_label_steps( 5 );

// display the data
echo $g->render();

?>
```

Bill Calculation

```
<?php
$rate_low = 2.5;
$rate_medium = 5;
$rate_high =7.5;

$limit_down = 100;
```

```
$limit_up = 300;


if ($usage <= $limit_down)
{
        $bill = $usage * $rate_low;
}
elseif ($usage > $limit_down and $usage <= $limit_up)
{
        $bill = ($usage-$limit_down) * $rate_medium + $limit_down*$rate_low;
}
elseif ($usage> $limit_up)
{
        $bill = ($usage-$limit_up)*$rate_high +  $limit_up*$rate_medium +
$limit_down*$rate_low;
}
?>
```

Consumer: Energy Consumption Data

```
<?php
session_start();
$con = mysql_connect("rems.mcs.edu.pk","rsas","sasr");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
$auser = $_SESSION['user'];
$sim = $_SESSION['SIM'];

$entresult = mysql_query("SELECT * FROM billing.graphdata WHERE
username='".$auser."'");
while($sqll = mysql_fetch_assoc( $entresult) ) {
$llim[] = $sqll['lowerlim'];
$ulim[] = $sqll['upperlim'];
}

$result = mysql_query("SELECT * FROM billing.consumption WHERE SIM =
'".$sim."' AND entryTime between '".$llim['0']."' and '".$ulim['0']."' ORDER BY
entryTime");
while($sql = mysql_fetch_assoc( $result) ) {
$data[] = $sql['units'];
$labs[] = $sql['entryTime'];
$f=strtotime("".$sql['entryTime']."");
```

```
$x[]=date('Y-m-d H:m',$f);

}

// use the chart class to build the chart:
include_once( 'ofc-library/open-flash-chart.php' );

$g = new graph();

// Energy Consumption
$g->title('Energy Consumption', '{font-size: 26px;}');

$g->set_data($data);
// label each point with its value
$g->set_x_labels($x);

$g->set_x_label_style( 10, '#9933CC', 0, 2 );

// set the Y max
$g->set_y_max( 100 );
// label every 20 (0,20,40,60)
$g->y_label_steps( 5 );

// display the data
echo $g->render();

?>
```

Consumer: Energy Consumption Graph

```
<?php
define("L_LANG", "en_US");

$mydate = isset($_POST["date1"]) ? $_POST["date1"] : "";
$mydate2 = isset($_POST["date2"]) ? $_POST["date2"] : "";

// Load the calendar class
require('calendar/tc_calendar.php');

                $myCalendar = new tc_calendar("date1", true);
                echo $myCalender->date1;
```

```
                $myCalendar->setIcon("calendar/images/iconCalendar.gif");
                $myCalendar->setDate(date('d'), date('m'), date('Y'));
                $myCalendar->setPath("calendar/");
                $myCalendar->zindex = 150; //default 1
                $myCalendar->setYearInterval(1995, date('Y'));
                $myCalendar->dateAllow('1960-03-01', date('Y-m-d'));
                //$myCalendar->autoSubmit(true, "calendar");
                $myCalendar->disabledDay("sat,sun");
                $myCalendar->setSpecificDate(array("2011-04-14", "2010-12-25"), 0,
'month');
                $myCalendar->setSpecificDate(array("2011-04-01"), 0, 'year');
                $myCalendar->setAlignment('right', 'bottom'); //optional
                $myCalendar->writeScript();

//////////////////////////////////////////////////////

                $myCalendar = new tc_calendar("date2", true);
                $myCalendar->setIcon("calendar/images/iconCalendar.gif");
                $myCalendar->setDate(date('d'), date('m'), date('Y'));
                $myCalendar->setPath("calendar/");
                $myCalendar->zindex = 150; //default 1
                $myCalendar->setYearInterval(1995, date('Y'));
                $myCalendar->dateAllow('1960-03-01', date('Y-m-d'));
                //$myCalendar->autoSubmit(true, "calendar");
                $myCalendar->disabledDay("sat,sun");
                $myCalendar->setSpecificDate(array("2011-04-14", "2010-12-25"), 0,
'month');
                $myCalendar->setSpecificDate(array("2011-04-01"), 0, 'year');
                $myCalendar->setAlignment('right', 'bottom'); //optional
                $myCalendar->writeScript();
```

Open Graph

```
include_once 'ofc-library/open_flash_chart_object.php';
open_flash_chart_object( 500, 250, 'http://'. $_SERVER['SERVER_NAME']
./condata.php');
$con = mysql_connect("216.227.215.208","rsas","sasr");
?>
```