**CHAPTER 1**

**INTRODUCTION**

**1.1 OUTLINE OF THE PROJECT**

Now a days, things are changing rapidly in the world. Many machines are developed which makes human life easier. Now, people are so much busy in their work. People think that there should be a technology that will reduce their work load. Internet of Thing fulfils this requirement of people. We proposed the design and construction of an automated product purchasing system based upon their weight. In household needs like gas, water and monthly purchasable things purchased on measuring kilograms criteria. We deployed the automatic system that was continuously monitoring the weight of those products if any one products reduced in their weight system automatically alert server about that item which is need to purchase and book. From android side user view that indication alert and order required item.

Household items weight are measured, each items having individual weight in its own when those all full. On every day usage it may get reduced day by day. Weights of those products are monitored by load cell sensing unit. Load cell connected with microcontroller, which monitor the sensor value and calculate the weight of the product. When product weight gone below minimum level alert message is given to android mobile through Zigbee. Zigbee is connected with microcontroller and receiver Zigbee connected along with mobile by OTG cable. In user mobile receive the alert message, user placed the order based on requirement. Payment also done through mobile.

**1.2 LITERATURE SURVEY**

Paper 1: “An Improved Real-Time Surveillance System For Home Security System Using Beagleboard SBC, Zigbee And FTP Webserver”,Rakesh V S,Sreesh P R Sudhish N George, 978-1-4673-2272-0/12,2012 IEEE.

**Description:** Real-time surveillance is an important aspect of an intelligent building with modern security demands. The proposed system implements an embedded system for monitoring wireless sensor nodes and camera installed inside a building for security surveillance. A number of surveillance devices in a Zigbee protocol (IEEE 802.15.4) based wireless network are connected to a BeagleBoard Single Board Computer (SBC) based surveillance management system. Remote alerting on fire and intruder detection are the key features of the system. When smoke or intruder movement is detected, the system sends warning messages through Short Message Service (SMS) to cell phones, starts capturing real-time video for fixed duration and makes the alarm on. The captured video clip is immediately uploaded to an FTP (File Transfer Protocol) webserver so that it can be retrieved later from anywhere around the world. The advantages of the system are that it guarantees reliability by integrating various components of a security system (sensors, alarm, camera, wireless connectivity etc.) and utilizes an FTP server for camera feeds.

Paper 2: “Automatic LPG Gas Cylinder Booking Software and Weight Measurement Using Load Cell & GSM”,Prof.P.S.Sonawane, Darade Pooja, Kankrale Pratiksha, *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*

**Description:**Liquefied petroleum gas (LPG) is the most important part of domestic daily human life but the safety of human life is required for gas explosion.this methodology are used to gives the protection to human life.when there is a gas leakaged by using automatic indication.it also provide the feature for automatic LPG gas booking when owner is busy and measure continues weight of LPG cylinder using load cell. Keywords: ARM7 microcontroller,LCD display,GSM module,Load cell,MQ6 gas sensor.

Paper 3: “ IoT applications on Secure Smart Shopping System”,Ruinian Li, Tianyi Song, Nicholas Capurso, Jiguo Yu, Jason Couture, and Xiuzhen Cheng,2016 IEEE.

**Description:**The Internet of Things (IoT) is changing human lives by connecting everyday objects together. For example, in a grocery store all items can be connected with each other, forming a smart shopping system. In such an IoT system, an inexpensive RFID tag can be attached to each product which, when placed into a smart shopping cart, can be automatically read by a cart equipped with an RFID reader. As a result, billing can be conducted from the shopping cart itself, preventing customers from waiting in a long queue at checkout. Additionally, smart shelving can be added into this system, equipped with RFID readers, and can monitor stock, perhaps also updating a central server. Another benefit of this kind of system is that inventory management becomes much easier, as all items can be automatically read by an RFID reader instead of manually scanned by a laborer. To validate the feasibility of such a system, in this work we identify the design requirements of a smart shopping system, build a prototype system to test functionality, and design a secure communication protocol to make the system practical. To the best of our knowledge, this is the first time a smart shopping system is proposed with security under consideration.

Paper 4 : “Mobile Application for Creating Presence Lists”,Zuzana VANTOVÁ, Ján PARALI, Vladimír GAŠPAR,IEEE 15th International Symposium on Applied Machine Intelligence and Informatics , 2017

**Description:**This paper presents an application for managing education by creating presence lists. An administration application has also been created in order to allow manipulating with the content of the enumerables in the presence lists. Overviews of possible solutions are presented. Our main motivation was to provide secure and reliable way of evaluating student attendance on specific lectures. In order to accomplish this task, we utilized the NFC technology on a smartphone and student ISIC ID card, the ownership of which is compulsory for every university student at the Technical University of Košice. The application has been implemented for specific conditions at the Technical University of Košice

Paper 5: “Energy Efficient Solar Based Digital Electronic Weighing Machine”,Sandip N. Rikame, 5th International Conference on Computer and Communication Technology, 2014

**Description:**The aim of proposed system is to develop an energy efficient digital electronic weighing machine which operates with the help of solar PV panel. This system describes the design and implementation of energy efficient, low cost and portable solar powered digital electronic weighing system. The system has highly reduced circuitry as it utilizes a standalone microcontroller chip. As this machine does not use the electricity to charge, it saves the electrical energy. This article gives a detailed weighing controller design method.

**CHAPTER 2**

**AIM AND SCOPE OF PRESENT INVESTIGATION**

**2.1 SCOPE OF THE PROJECT**

The main aim of the project is to automate the daily day-to-day activities which involve manual intervention and can be automated. This could reduce the strain, in keeping track of the quantities of the daily household commodities and having to manually order them. This could ensure that none of the household commodities go out of stock.

**2.2 EXISTING SYSTEM:**

In Existing system**,** Every products of home is purchased after manually check. User check the came to when it’s almost empty its takes some time to deliver. In monthly base purchase need to take list then only purchase order raised.Although the current system keeps track of the quantities of the household products, it does not connect with the suppliers to place the order, to fully automate the process of booking.

**2.3 PROPOSED SYSTEM:**

In our proposed system household items weight are measured, each items having individual weight in its own when those all full. On every day usage it may get reduced day by day. Weights of those products are monitored by load cell sensing unit. Load cell connected with microcontroller, which monitor the sensor value and calculate the weight of the product. When product weight gone below minimum level alert message is given to android mobile through Zigbee. Zigbee is connected with microcontroller and receiver Zigbee connected along with mobile by OTG cable. In user mobile receive the alert message, user placed the order based on requirement. Payment also done through mobile.

*ADVANTAGES:*

* Arduino simplifies the amount of hardware and software development you need to do in order to get a system running.
* The Arduino hardware platform already has the power and reset circuitry setup as well as circuitry to program and communicate with the microcontroller over USB.

**CHAPTER 3**

**EXPERIMENTAL OR MATERIALS USED; ALGORITHM USED**

**3.1 LANGUAGE SPECIFICATION:**

This chapter describes the requirement analysis in accordance with the input and the resources and it also describes the implementation of the project with the technology used.

**3.2 REQUIREMENT ANALYSIS**

Requirement analysis determines the requirements of a new system. This project analyses on product and resource requirement, which is required for this successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements give in brief about the software and hardware that are needed to achieve the required functionality.

**3.3 HARDWARE ENVIRONMENT**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

* Processor : [Core i3](https://en.wikipedia.org/wiki/List_of_Intel_microprocessors#Core_i3_2)/i5/i7
* RAM : 2-4GB
* HDD : 500 GB
* Embedded Fabrication Kit

**3.4 SOFTWARE ENVIRONMENT**

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team’s and tracking the team’s progress throughout the development activity.

* Platform : Windows Xp/7/8
* Front End : Java-JDK1.7,Android-sdk and Eclipse,
* Back End : Apache tomcat, Embedded C

**3.5 TECHNOLOGIES USED:-**

**3.5.1 ANDROID**

Android is an open source and Linux-based Operating System for mobile devices such as smart phones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008. On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance. The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public

Android is a powerful operating system competing with Apple 4GS and supports great features. Few of them are listed below.

**Table 1: Feature Description**

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Beautiful UI | Android OS basic screen provides a beautiful and intuitive user interface. |
| Connectivity | GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX. |
| Storage | SQLite, a lightweight relational database, is used for data storage purposes. |
| Media support | H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg  Vorbis, WAV, JPEG, PNG, GIF, and BMP |
| Messaging | SMS and MMS |
| Web browser | Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript  Engine supporting HTML5 and CSS3. |
| Multi-touch | Android has native support for multi-touch which was initially made available in handsets  Such as the HTC Hero. |
| Multi-tasking | User can jump from one task to another and same time various application can run  Simultaneously. |
| Resizable widgets | Widgets are resizable, so users can expand them to show more content or shrink them to  save space |
| Multi-Language | Supports single direction and bi-directional text. |
| GCM | Google Cloud Messaging (GCM) is a service that lets developers send short message data  to their users on Android devices, without needing a proprietary sync solution. |
| Wi-Fi Direct | A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer  connection |
| Android Beam | A popular NFC-based technology that lets users instantly share, just by touching two NFC enabled  phones together. |

# Android Core Building Blocks



Fig 3.1: Android Core Building Blocks

**3.5.2 EMBEDDED SYSTEM**

An embedded system is a [computer system](http://en.wikipedia.org/wiki/Computer_system) designed to perform one or a few dedicated functions often with [real-time computing](http://en.wikipedia.org/wiki/Real-time_computing) constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a [personal computer](http://en.wikipedia.org/wiki/Personal_computer), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Embedded systems are controlled by one or more main processing cores that is typically either a [microcontroller](http://en.wikipedia.org/wiki/Microcontroller) or a [digital signal processor](http://en.wikipedia.org/wiki/Digital_signal_processor) (DSP). The key characteristic is however being dedicated to handle a particular task, which may require very powerful processors.

Physically, embedded systems range from portable devices such as [digital watches](http://en.wikipedia.org/wiki/Digital_watch) and [MP3 players](http://en.wikipedia.org/wiki/Digital_audio_player), to large stationary installations like [traffic lights](http://en.wikipedia.org/wiki/Traffic_light), [factory controllers](http://en.wikipedia.org/wiki/Programmable_logic_controller), or the systems controlling [nuclear power plants](http://en.wikipedia.org/wiki/Nuclear_power_plant). Complexity varies from low, with a single [microcontroller](http://en.wikipedia.org/wiki/Microcontroller) chip, to very high with multiple units, peripherals and networks mounted inside a large [chassis](http://en.wikipedia.org/wiki/Chassis) or enclosure.

### 3.5.3 PERIPHERALS

Embedded Systems talk with the outside world via [peripherals](http://en.wikipedia.org/wiki/Peripheral), such as:

* Serial Communication Interfaces (SCI): [RS-232](http://en.wikipedia.org/wiki/RS-232), [RS-422](http://en.wikipedia.org/wiki/RS-422), [RS-485](http://en.wikipedia.org/wiki/RS-485) etc
* Synchronous Serial Communication Interface: [I2C](http://en.wikipedia.org/wiki/I2C), [SPI](http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus), SSC and ESSI (Enhanced Synchronous Serial Interface)
* [Universal Serial Bus](http://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB)
* Multi Media Cards (SD Cards, Compact Flash etc)
* Networks: [Ethernet](http://en.wikipedia.org/wiki/Ethernet), [Controller Area Network](http://en.wikipedia.org/wiki/Controller_Area_Network), [LonWorks](http://en.wikipedia.org/wiki/LonWorks), etc
* Timers: [PLL](http://en.wikipedia.org/wiki/PLL)(s), Capture/Compare and [Time Processing Units](http://en.wikipedia.org/wiki/Time_Processing_Unit)
* Discrete IO: aka [General Purpose Input/Output](http://en.wikipedia.org/wiki/General_Purpose_Input/Output) (GPIO)
* Analog to Digital/Digital to Analog ([ADC](http://en.wikipedia.org/wiki/Analog-to-digital_converter)/[DAC](http://en.wikipedia.org/wiki/Digital-to-analog_converter))
* Debugging: [JTAG](http://en.wikipedia.org/wiki/JTAG), [ISP](http://en.wikipedia.org/wiki/In-System_Programming), [ICSP](http://en.wikipedia.org/wiki/In_Circuit_Serial_Programming_%28ICSP%29), [BDM](http://en.wikipedia.org/wiki/Background_Debug_Mode_interface) Port, BITP DP9 port

**3.5.4 SENSORS**

Sensors are devices (usually electro-mechanical) which help us measure a physical parameter (such as temperature, pressure, force, acceleration etc.) by providing a signal that either quantitatively measures (level) that physical parameter or provides a simple binary signal that indicates a yes/no signal that tells us if something occurred or not (such as a touch sensor). Most sensors require power to be provided to a sensing element and an electrical signal is then generated after the measurement.

**3.5.5 ZIGBEE**

ZigBee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless M2M networks. The ZigBee standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz.

The 802.15.4 specification upon which the ZigBee stack operates gained ratification by the Institute of Electrical and Electronics Engineers (IEEE) in 2003. The specification is a packet-based radio protocol intended for low-cost, battery-operated devices. The protocol allows devices to communicate in a variety of network topologies and can have battery life lasting several years.

The ZigBee protocol has been created and ratified by member companies of the ZigBee Alliance. Over 300 leading semiconductor manufacturers, technology firms, OEMs and service companies comprise the ZigBee Alliance membership. The ZigBee protocol was designed to provide an easy-to-use wireless data solution characterized by secure, reliable wireless network architectures.

**3.5.6 ARDUINO**

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world.

The project is based on microcontroller board designs, manufactured by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers.

The first Arduino was introduced in 2005, aiming to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

**CHAPTER 4**

**RESULTS AND DISCUSSIONS AND PERFORMANCE ANALYSIS**

**4.1 WORKING OF HARDWARE:**

The household products are placed on the load cell, which continuously monitors the weight of the household product. The weight is monitored continuously and sent to the servers. When the weight of that particular product goes below the threshold value an order is initiated to the seller of that particular product. The weight measured by the Load Cell is sent to the servers with the help of an zigbee module connected to an Arduino.The server receives the signal sent by the load cell module with the help of another zigbee module.

**4.2 ARCHITECTURE DIAGRAM:**

**Arduino Micro-Controller**

**Power supply 5v**

**ZIGBEE**

**Load cell sensor**

Fig 4.1: Hardware Architecture

**4.3 DATA FLOW DIAGRAM:**

LEVEL 0:

House hold products

Particular department

Fig 4.2: Level-0 DataFlow Diagram

LEVEL 1:

House hold things

PC

Load cell

Corresponding department

User

Android application

Fig 4.3: Level-1 DataFlow Diagrams

* 1. **MODULES**

A modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comps ability, Modular Understandability, Modular continuity, Modular Protection.

The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

* + 1. **EMBEDDED HARDWARE INTERFACE:**

In this module, Embedded Hardware is connected with Load cell unit to measure the load of the object along with the Wifi Module which is used for communication. We use Arduino micro controller as the micro controller for effective hardware processing. Load cell unit is used for analysis of the weight of the corresponding item / object.

**4.4.2 ZIGBEE COMMUNICATION:**

ZigBee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless M2M networks. The ZigBee standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz . The ZigBee protocol was designed to provide an easy-to-use wireless data solution characterized by secure, reliable wireless network architectures. Support for multiple network topologies such as point-to-point, point-to-multipoint and mesh networks.

* Low duty cycle – provides long battery life
* Low latency
* Direct Sequence Spread Spectrum (DSSS) Up to 65,000 nodes per network
* 128-bit AES encryption for secure data connections, Collision avoidance, retries and acknowledgements.

We would use a pair of zigbees for communication. One zigbee would be connected to the server and the other to the micro controller. The zigbee is used to pass a signal from the device when the product weight is low and nearing completion. The other zigbee would receive this signal and place an order automatically.

* + 1. **ANDROID APPLICATION:**

In this module, Android application is deployed to view the booking information automatically made by the software application. The booking process would happen automatically once the load of the corresponding weight of the product / item is reduced. All the records of booking information can be viewed via android application by the user.

* + 1. **LOAD CALCULATION:**

In this module, Load cell connected in the embedded hardware measures the weight of the corresponding product / item and if the weight is identified as low from the standard / expected weight then the embedded hardware communicates to the main server via Wifi Module, so that the ordering process is initiated automatically in the java server end. The product / item is ordered by the keyword communicated by the embedded hardware to the main sever via Wifi Module.

* + 1. **AUTOMATED ORDERING:**

In this Module, once the java server application receives the ordering request by the analysis made by the embedded hardware – Load Cell via Wifi Module, ordering of the corresponding product / item is placed automatically by the server itself by feting the ordering information stored in the database of the server. All the ordering details are stored in the main java server. Once the application gets the ordering request from the hardware, order is made by the java main server so that replacement can happen very easily without any manual interface.

**4.3 SCREEN SHOT**

**SCREENSHOTS OF OUTPUT:**

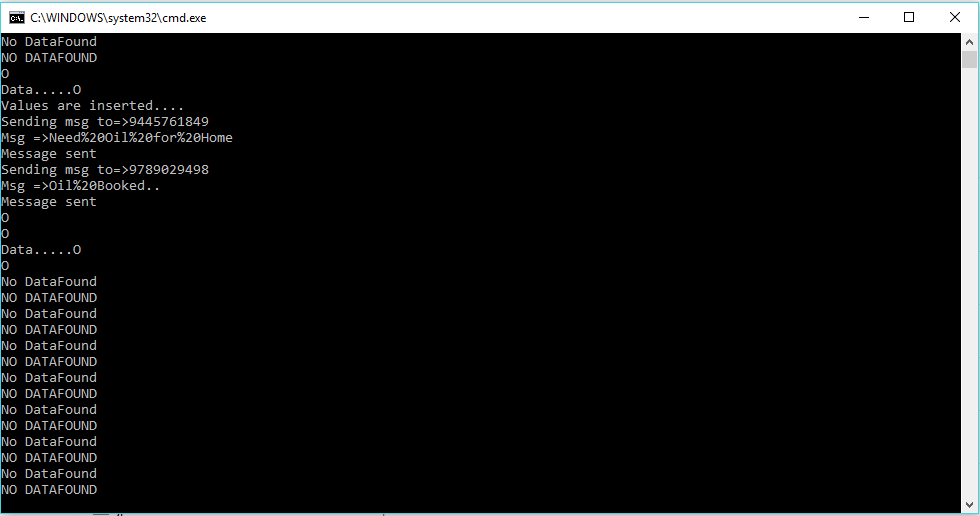


Fig 8.1: GAS SERVER RUNNING

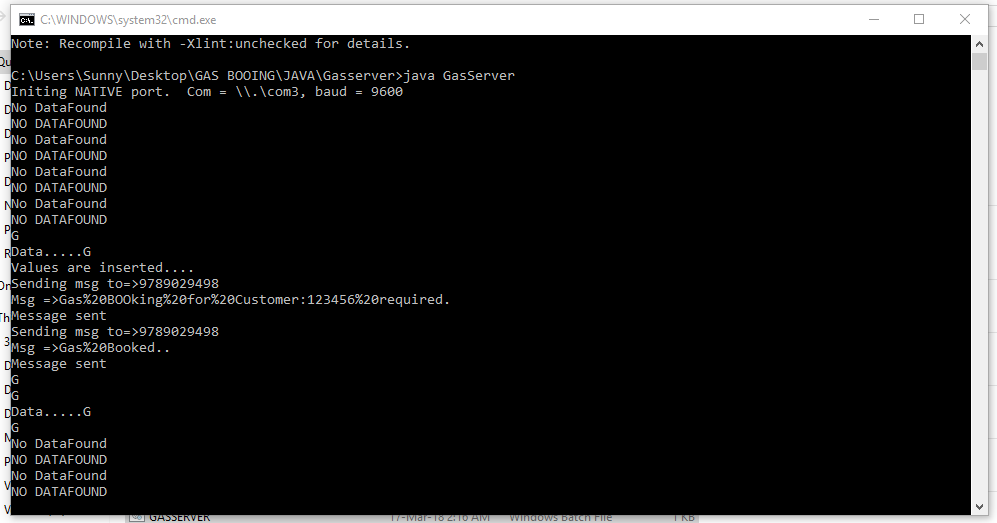


Fig 8.2: OIL SERVER RUNNING

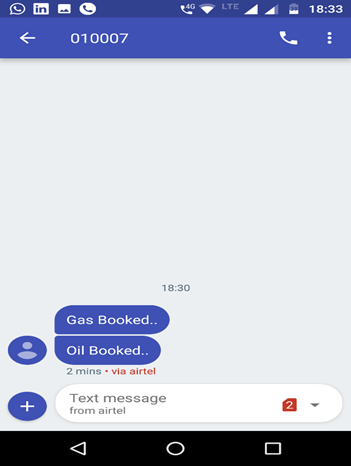
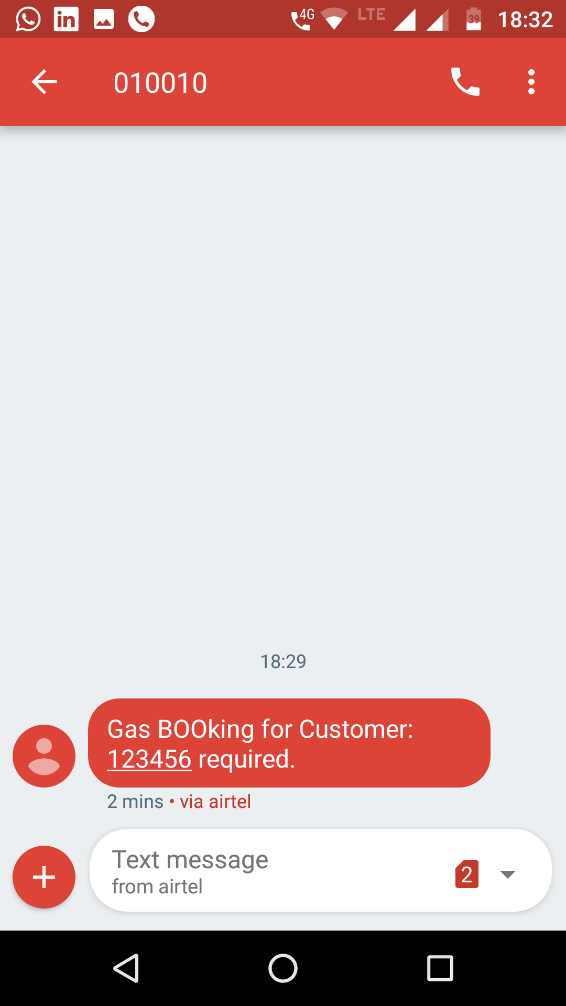


Fig 8.3: RETAILER MESSAGE Fig 8.4: CUSTOMER MESSAGE

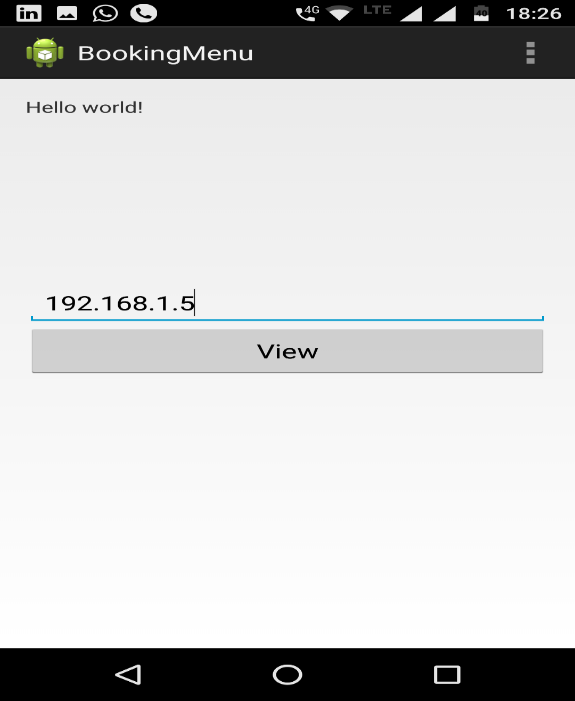


Fig 8.5: ANDROID APPLICATION FOR BOOKING HISTORY

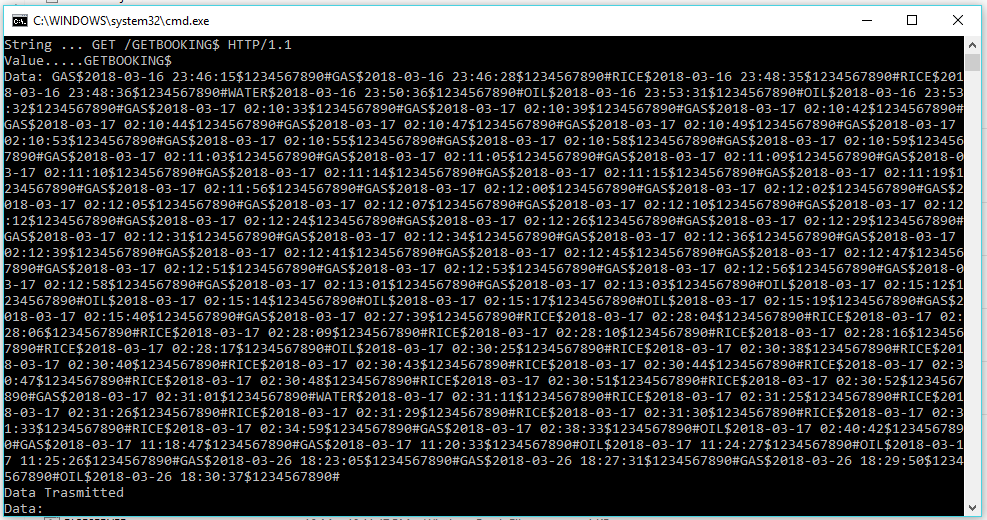


Fig 8.6: BOOKING HISTORY RETRIEVAL FROM MYSQL

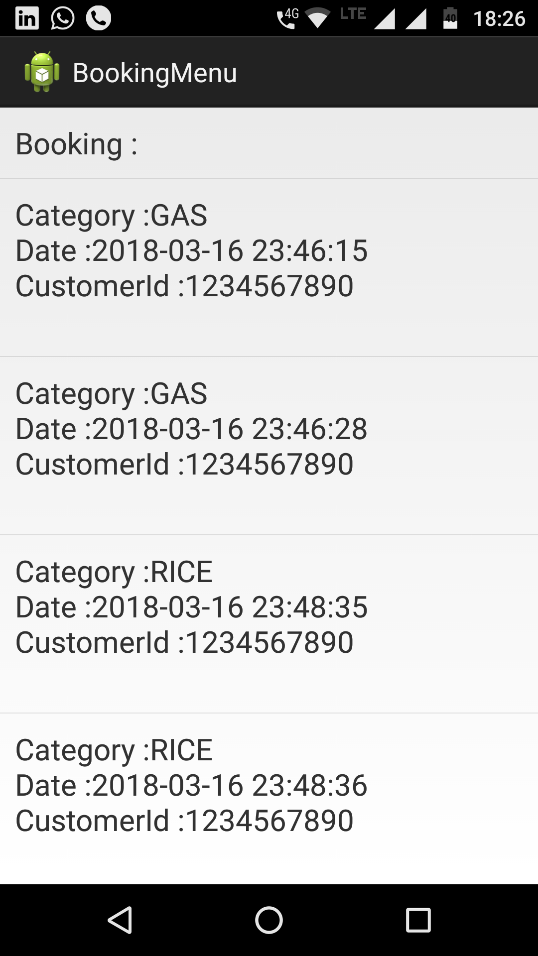


Fig 8.7: BOOKING HISTORY DISPLAY MENU

**CHAPTER 5**

**SUMMARY AND CONCLUSION**

**5.1 SUMMARY**

Thus the project infer that automatic item booking using IOT. System will automatically book the items if that item is going to empty. So for that we are implementing automatic booking system for the house hold items. We are including android application for the purpose to know the booking details.

**5.2 CONCLUSION**

This project can be further enhanced to automate the manual process of monitoring and ordering of all the household products, thus efficiently managing the commodities more efficiently and effectively. This project can be extended to commercial and industrial applications to manage the usage of commodities. This can be used to effectively manage the usage of various commodities in industries to ensure that the quality of the manufactured goods. It can be used in resource scare sectors like hotel industry, to monitor the amount of commodities used for the manufacturing .

**APPENDIX**

**A.SOURCE CODE:**

#include <LiquidCrystal.h>

LiquidCrystal lcd(11, 10 , 5, 4, 3, 2);

const int analogInPin = A0;

int potvalue = 0;

char gas=0,oil=0,rice=0,water=0;

boolean stringcomplete = false;

char rec=0,start=0;

int key;

char D1=0,D2=0,D3=0,D4=0;

int first=0;

int one=0;

void setup()

{

unsigned char STATUS;

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

pinMode(9, OUTPUT);

lcd.begin(16, 2);

delay(2000);

Serial.begin(9600);

}

void loop()

{

potvalue = analogRead(analogInPin);

lcd.setCursor(0, 0);

lcd.print("Value:");

lcd.print(potvalue);

delay(500) ;

if(gas==1 && first==1)

{

if(potvalue<230)

{

Serial.write('G');first=0;

}

}

else if(rice==1 && first==1)

{

if(potvalue<190)

{

Serial.write('R');first=0; }

}

else if(water==1 && first==1)

{

if(potvalue<160)

{

Serial.write('W');first=0;

}

}

else if(oil==1 && first==1)

{

if(potvalue<140)

{

first=0;

Serial.write('O');

}

}

if (stringcomplete)

{

first=1;

if(rec=='A')

{

lcd.setCursor(0, 1);

lcd.print("Gas");gas=1;oil=0;rice=0;water=0;

}

else if(rec=='B')

{

lcd.setCursor(0, 1);

lcd.print("Rice");gas=0;oil=0;rice=1;water=0;

}

else if(rec=='C')

{

lcd.setCursor(0, 1);

lcd.print("Water"); gas=0;oil=0;rice=0;water=1;

}

else if(rec=='D')

{

lcd.setCursor(0, 1);

lcd.print("Oil");gas=0;oil=1;rice=0;water=0;

}

stringcomplete=false;

}

}

void serialEvent()

{

while (Serial.available())

{

char inChar=(char)Serial.read();

rec=inChar;

stringcomplete=true; } }

**ANDROID APPLICATION DEVELOPMENT-**

**Main Page:**

public class MainActivity extends Activity {

EditText ip\_Addr\_txt;

Button View\_btn;

@SuppressLint("NewApi")

@TargetApi(Build.VERSION\_CODES.GINGERBREAD)

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

if (android.os.Build.VERSION.SDK\_INT > 9) {

StrictMode.ThreadPolicy policy =

new StrictMode.ThreadPolicy.Builder().permitAll().build();

StrictMode.setThreadPolicy(policy);

}

ip\_Addr\_txt=(EditText)findViewById(R.id.editText\_ip);

View\_btn=(Button)findViewById(R.id.button\_view);

View\_btn.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

ServerIPAddress.setserverIPaddress(ip\_Addr\_txt.getText().toString());

Intent i=new Intent(MainActivity.this,DisplayBooking.class);

startActivity(i);

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

}

**Server IP address:**

package com.example.bookingmenu;

public class ServerIPAddress {

public static String serverIPaddress = null;

public static String ID = "1";

public static String Kural = null;

public static String Meaning=null;

public static String url=null;

public static String getMeaning() {

return Meaning;

}

public static void setMeaning(String Meaning) {

ServerIPAddress.Meaning = Meaning;

}

public static String getKural() {

return Kural;

}

public static void setKural(String Kural) {

ServerIPAddress.Kural = Kural;

}

public static String getID() {

return ID;

}

public static void setID(String ID) {

ServerIPAddress.ID = ID;

}

public static String getserverIPaddress() {

return serverIPaddress;

}

public static void setserverIPaddress(String serverIPaddress) {

ServerIPAddress.serverIPaddress = serverIPaddress;

}

public static String geturl() {

return url;

}

public static void seturl(String url) {

ServerIPAddress.url = url;

}

}

**Display booking system:**

public class DisplayBooking extends ListActivity {

static String text = "";

String message = "";

String Encode;

private HttpResponse response;

private HttpClient httpclient;

private String responseText = "";

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

LinearLayout linearLayout = new LinearLayout(this);

String reply = getServerConnection("GETBOOKING$");

if(reply!=null && reply.intern()!="NODATA"){

message = reply;

}

StringTokenizer st = new StringTokenizer(message,"#");

ArrayList<String> al = new ArrayList<String>();

al.add("Booking :");

while(st.hasMoreTokens()){

String parent=st.nextToken();

StringTokenizer stt=new StringTokenizer(parent,"$");

String category=stt.nextToken().toString();

String date=stt.nextToken().toString();

String cusId=stt.nextToken().toString();

StringBuilder sb=new StringBuilder();

sb.append("Category :"+category+"\n");

sb.append("Date :"+date+"\n");

sb.append("CustomerId :"+cusId+"\n");

al.add(sb.toString());

}

al.add("Back");

Log.v("Search output",al.toString());

setListAdapter(new ArrayAdapter<String>(this, R.layout.list\_item, al));

ListView lv = getListView();

lv.setTextFilterEnabled(true);

lv.setOnItemClickListener(new OnItemClickListener() {

public void onItemClick(AdapterView<?> parent, View view,

int position, long id) {

String selectedText = ((TextView) view).getText().toString();

if(selectedText.intern() == "Back"){

Intent intent = new Intent(DisplayBooking.this,MainActivity.class);

startActivity(intent);

}else{

text = selectedText;

}

}

});

}

private String getServerConnection(String data){

try {

httpclient = new DefaultHttpClient();

HttpGet httpget = new HttpGet("http://"+ServerIPAddress.getserverIPaddress()+":222/"+data);

response = httpclient.execute(httpget);

HttpEntity entity = response.getEntity();

responseText = EntityUtils.toString(entity);

}catch (Exception e){

Toast.makeText(this,"error"+e.toString(), Toast.LENGTH\_LONG).show();

}

return responseText.trim(); }}

**JAVA SERVER CREATION-**

**Gas server:**

public class GasServer{

Connection con=null;

Statement stmt=null;

ResultSet rs=null;

String c,bname,bauthor;

SimpleSerial m\_SerialPort = null; // Serial port

int m\_PortIndex

boolean m\_IsNative = true;

String inputString ;

static String klmString=null;

StringTokenizer stringToken = null;

String replyMessage = "";

String mobileNo = "";

String name = null;

int fine;

OutputStream outputStream;

boolean status=true;

String vehicleNo,gpslocation;

static final String m\_PrefsFileName = new String("JavaTerm.prefpublic static void main(String[] argh) {

new GasServer();

}

private void initSerialPort() throws IOException {

if (m\_SerialPort != null) {

m\_SerialPort.close();

m\_SerialPort = null;

}

if (m\_IsNative) {

m\_SerialPort = new SimpleSerialNative(3);

}

if (!m\_SerialPort.isValid()) {

throw (new IOException("Serial port not opened"));

}

}

GasServer() {

int ii;

try {

DataInputStream prefs = new DataInputStream(new FileInputStream(m\_PrefsFileName));

m\_PortIndex = prefs.readInt();

m\_IsNative = prefs.readBoolean();

if (m\_PortIndex < 0 ) {

throw new IOException(m\_PrefsFileName + " is corrupt");

}

initSerialPort();

}

catch(IOException e) {

System.out.println("preferences file 'JavaTerm.pref' not found / didn't open or there was a problem opening serial port. Searching for serial port");

int m=4;

find\_open\_serial\_port:

for (ii = 0; ii < m; ii++) {

try {

m\_PortIndex = ii + 1; // This is the serial port we want to open

initSerialPort(); // Try opening this serial port. Throws exception if there's a problem

System.out.println("Opening serial port Comm" + m\_PortIndex);

break find\_open\_serial\_port; // If we haven't thrown an exception, we're done

}

catch (IOException ee) { // wind up here if initSerialPort() above has a problem

if (ii == m - 1) {

System.out.println("Couldn't open any serial ports");

System.exit(0); // can't open any serial ports.

} } } }

// Infinite loop. WindowListener above will break us out of loop with call to System.exit(0);

for (;true;) {

// Get any pending characters from serial port.

// Returns empty string if there's nothing to read.

// This is in contrast to readByte() which patiently waits for data.

if (m\_SerialPort != null) {

inputString = m\_SerialPort.readString();

try{

if(inputString.intern()=="")

{

System.out.println("No DataFound");

System.out.println("NO DATAFOUND");

outputStream = m\_SerialPort.getOutputStream(); outputStream.write("A".getBytes());

}else{

try {

System.out.println(inputString);

File file = new File("help.txt");

boolean bool = file.exists();

if(bool){

file.delete();

}

FileOutputStream fout1 = new FileOutputStream("help.txt",true);

fout1.write(inputString.getBytes());

fout1.close();

//Thread.sleep(2000);

FileInputStream fin = new FileInputStream("help.txt");

byte b[] = new byte[fin.available()];

fin.read(b);

fin.close();

String data = new String(b);

System.out.println("Data....."+data);

//Thread.sleep(10000);

if(data.startsWith("G")){

if(status){

con = getConnection();

if(con==null){

System.out.println("Connection not found");

System.exit(0);

}else{

stmt = con.createStatement();

int i = stmt.executeUpdate("insert into booking\_details(category, date\_d, customer\_no) values('GAS',now(),'1234567890')");

if(i>0){

System.out.println("Values are inserted....");

SendSMS sendSMS=new SendSMS();

SendSMS.sendSMS("9789029498", "Gas BOOking for Customer:123456 required.");

SendSMS.sendSMS("9789029498", "Gas Booked..");

status=false;

}else{

System.out.println("Values are not inserted....");}}}}

}catch (Exception e){

e.printStackTrace();

System.out.println(e);

}System.out.println(inputString);

}

}catch(Exception ex){System.out.println(ex);}

}

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {}

}

}private Connection getConnection() {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/gasbooking", "root", "admin");

return connection;

} catch (Exception ex) {

ex.printStackTrace();

System.out.println(ex);

}

return null;

}

}

**Rice server:**

public class RiceServer{

Connection con=null;

Statement stmt=null;

ResultSet rs=null;

String c,bname,bauthor;

SimpleSerial m\_SerialPort = null; // Serial port

int m\_PortIndex; // Which comm port to use (1-based value -- there is no Comm0)

boolean m\_IsNative = true;

String inputString ;

static String klmString=null;

StringTokenizer stringToken = null;

String replyMessage = "";

String mobileNo = "";

String name = null;

int fine;

OutputStream outputStream;

String vehicleNo,gpslocation;

// Use local code, or use JavaComm from Sun

static final String m\_PrefsFileName = new String("JavaTerm.pref"); // name of preferences file

public static void main(String[] argh) {

new RiceServer();

}

private void initSerialPort() throws IOException {

if (m\_SerialPort != null) {

m\_SerialPort.close();

m\_SerialPort = null;

}

if (m\_IsNative) {

m\_SerialPort = new SimpleSerialNative(m\_PortIndex);

}

if (!m\_SerialPort.isValid()) {

throw (new IOException("Serial port not opened"));

}

}

RiceServer() {

int ii;

try {

DataInputStream prefs = new DataInputStream(new FileInputStream(m\_PrefsFileName));

m\_PortIndex = prefs.readInt();

m\_IsNative = prefs.readBoolean();

if (m\_PortIndex < 0 ) {

throw new IOException(m\_PrefsFileName + " is corrupt");

}

initSerialPort();

}

catch(IOException e) {

System.out.println("preferences file 'JavaTerm.pref' not found / didn't open or there was a problem opening serial port. Searching for serial port");

int m=4;

find\_open\_serial\_port:

for (ii = 0; ii < m; ii++) {

try {

m\_PortIndex = ii + 1; // This is the serial port we want to open

initSerialPort(); // Try opening this serial port. Throws exception if there's a problem

System.out.println("Opening serial port Comm" + m\_PortIndex);

break find\_open\_serial\_port; // If we haven't thrown an exception, we're done

}

catch (IOException ee) { // wind up here if initSerialPort() above has a problem

if (ii == m - 1) {

System.out.println("Couldn't open any serial ports");

System.exit(0); // can't open any serial ports.

}

}

}

}

// Infinite loop. WindowListener above will break us out of loop with call to System.exit(0);

for (;true;) {

// Get any pending characters from serial port.

// Returns empty string if there's nothing to read.

// This is in contrast to readByte() which patiently waits for data.

if (m\_SerialPort != null) {

inputString = m\_SerialPort.readString();

try{

if(inputString.intern()=="")

{

System.out.println("No DataFound");

System.out.println("NO DATAFOUND");

outputStream = m\_SerialPort.getOutputStream(); outputStream.write("B".getBytes());

}else{

try {

System.out.println(inputString);

File file = new File("help.txt");

boolean bool = file.exists();

if(bool){

file.delete();

}

FileOutputStream fout1 = new FileOutputStream("help.txt",true);

fout1.write(inputString.getBytes());

fout1.close();

//Thread.sleep(2000);

FileInputStream fin = new FileInputStream("help.txt");

byte b[] = new byte[fin.available()];

fin.read(b);

fin.close();

String data = new String(b);

System.out.println("Data....."+data);

if(data.startsWith("R")){

con = getConnection();

if(con==null){ System.out.println("Connection not found");

System.exit(0);

}else{

stmt = con.createStatement();

int i = stmt.executeUpdate("insert into booking\_details(category, date\_d, customer\_no) values('RICE',now(),'1234567890')");

if(i>0){ System.out.println("Values are inserted....");

}else{ SendSMS sendSMS=new SendSMS(); sendSMS.sendSMS("9789029498", "Need Oil for Home"); sendSMS.sendSMS("9789029498", "Gas Booked.."); System.out.println("Values are not inserted....");

}}}

}catch (Exception e){

e.printStackTrace();

System.out.println(e);

}System.out.println(inputString);

}

}catch(Exception ex){System.out.println(ex);}

}

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {}

}

}

private Connection getConnection() {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection connection = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/gasbooking", "root", "admin");

return connection;

} catch (Exception ex) {

ex.printStackTrace();

System.out.println(ex);

}

return null;

}}

**NOTE:** Similarly Servers can be set for other household items like Oil, Water, Cereals,etc.

**Main server:**

public class MainServer implements Runnable{

static ServerSocket serverSocket = null;

static Socket socket = null;

private Connection con=null;

private Statement stmt=null;

private ResultSet rs=null;

boolean fake=false;

boolean expiry=false;

boolean theft=false;

//SimpleWrite sb;

private String dbRecords = "",purchaseList="",ccNumber,pinNumber;

private StringTokenizer stringToken = null;

int availableBalance = 0,purchaseAmt = 0, totalBalance = 0;

boolean cardAuthenticaionflag = false,balanceflag= false;

String charset = "UTF-8"; // or "ISO-8859-1" String QRData="";

PrintWriter out = null;

public MainServer(){

try{

//sb=new SimpleWrite();

serverSocket = new ServerSocket(222);

}catch(Exception ex){

ex.printStackTrace();

System.out.println(ex);

}

}

public void run(){

try{

out = new PrintWriter(socket.getOutputStream());

String str = HttpTunnel.undoHttpTunneling(socket.getInputStream());

str = str.substring(1);

System.out.println("Value....."+str);

if(str.startsWith("GETBOOKING")){

String dbdata="", record="";

boolean status=true;

try{

con = getConnection();

if(con==null){ System.out.println("Connection not found");

System.exit(0);

}else{

stmt = con.createStatement();

rs= stmt.executeQuery("select category, date\_d, customer\_no from booking\_details");

while(rs.next()){ dbdata+=rs.getString(1)+"$"+rs.getString(2)+"$"+rs.getString(3)+"#";

status=true;

}

if(true){

record=dbdata; HttpTunnel.doHttpTunneling(out,record);

} HttpTunnel.doHttpTunneling(out,"");

}

}

catch(Exception e){

e.printStackTrace();};

}

else if(str.startsWith("INVALID")){

//new SimpleWrite().writeData("B");

System.out.println("VALID"+"A");

HttpTunnel.doHttpTunneling(out,"CLOSE");

}

System.out.println("Value....."+str);

}catch(Exception ex){

ex.printStackTrace();

System.out.println(ex);

}

}

public static void main(String[] args) {

try{

MainServer ms = new MainServer();

while(true){

socket = serverSocket.accept();

Thread thread = new Thread(ms);

thread.start();

}

}catch(Exception ex){

ex.printStackTrace();

System.out.println(ex);

} }

private Connection getConnection() {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection connection = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/gasbooking", "root", "admin");

return connection;

} catch (Exception ex) {

ex.printStackTrace();

System.out.println(ex);

} return null;

}

}

**B.PAPER:**

***REAL-TIME MONITORING AND AUTOMATED BOOKING OF HOUSEHOLD PRODUCTS***

S.Dhamodaran,

Assistant Professor, Dept. of CSE,

Sathyabama University,

Chennai, TamilNadu, India.

[s.dhamodaran07@gmail.com](mailto:s.dhamodaran07@gmail.com)

Santosh .V,

UG Student,Dept. of CSE,

Sathyabama University,

Chennai, TamilNadu, India.

[vsantosh222@gmail.com](mailto:vsantosh222@gmail.com)

Sailesh .M.S,

UG Student, Dept. of CSE,

Sathyabama University,

Chennai, TamilNadu, India.

[sailesh37@yahoo.in](mailto:sailesh37@yahoo.in)

***Abstract---* Now a days, things are changing rapidly in this ultra-modern world. Many machines have been developed which make human life easier. Now, people are getting so much busy in their day-to-day work. People think that there should be a technology that will reduce their work load. Internet of Things fulfils this requirement of people. We proposed the design and construction of an automated product purchasing system based upon their weight. In household needs like gas, water and monthly purchasable things purchased on measuring kilograms criteria. We have deployed the automatic system that would continuously monitoring the weight of those products if any one product is found to be reduced in their weight, system automatically alerts the server about that item which is need to be purchased or booked. From android side user would be able to view that indication alert and booking history of the items purchased.**

**The project intends to make use of Arduino to enable the server to continuously monitor the weight of the household commodities with the help of a load cell and automatically make booking to the corresponding seller of that corresponding commodity.**

***Keywords—automation; booking; weight monitoring;***

1. **INTRODUCTION**

Now a days, things are changing rapidly in the world. Many machines are developed which makes human life easier. Now, people are so much busy in their work. People think that there should be a technology that will reduce their work load. Internet of Thing fulfils this requirement of people. We proposed the design and construction of an automated product purchasing system based upon their weight. In household needs like gas, water and monthly purchasable things purchased on measuring kilograms criteria. We deployed the automatic system that was continuously monitoring the weight of those products if any one products reduced in their weight system automatically alert server about that item which is need to purchase and book. From android side user view that indication alert and order required item.

Household items weight are measured, each items having individual weight in its own when those all full. On every day usage it may get reduced day by day. Weights of those products are monitored by load cell sensing unit. Load cell connected with microcontroller, which monitor the sensor value and calculate the weight of the product. When product weight gone below minimum level alert message is given to android mobile through Zigbee. Zigbee is connected with microcontroller and receiver Zigbee connected along with mobile by OTG cable. In user mobile receive the alert message, user placed the order based on requirement. Payment also done through mobile.

1. **REVIEW OF RELATED WORK**

Paper 1: “An Improved Real-Time Surveillance System For Home Security System Using Beagleboard SBC, Zigbee And FTP Webserver”,Rakesh V S,Sreesh P R Sudhish N George, 978-1-4673-2272-0/12,2012 IEEE.

**Description:**

Real-time surveillance is an important aspect of an intelligent building with modern security demands. The proposed system implements an embedded system for monitoring wireless sensor nodes and camera installed inside a building for security surveillance. A number of surveillance devices in a Zigbee protocol (IEEE 802.15.4) based wireless network are connected to a BeagleBoard Single Board Computer (SBC) based surveillance management system. Remote alerting on fire and intruder detection are the key features of the system. When smoke or intruder movement is detected, the system sends warning messages through Short Message Service (SMS) to cell phones, starts capturing real-time video for fixed duration and makes the alarm on. The captured video clip is immediately uploaded to an FTP (File Transfer Protocol) webserver so that it can be retrieved later from anywhere around the world. The advantages of the system are that it guarantees reliability by integrating various components of a security system (sensors, alarm, camera, wireless connectivity etc.) and utilizes an FTP server for camera feeds.

Paper 2: “Automatic LPG Gas Cylinder Booking Software and Weight Measurement Using Load Cell & GSM”,Prof.P.S.Sonawane, Darade Pooja, Kankrale Pratiksha, *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*

**Description:**

Liquefied petroleum gas (LPG) is the most important part of domestic daily human life but the safety of human life is required for gas explosion. This methodology are used to gives the protection to human life. When there is a gas leaked by using automatic indication.it also provide the feature for automatic LPG gas booking when owner is busy and measure continues weight of LPG cylinder using load cell. Keywords: ARM7 microcontroller, LCD display, GSM module, Load cell, MQ6 gas sensor.

Paper 3: “ IoT applications on Secure Smart Shopping System”,Ruinian Li, Tianyi Song, Nicholas Capurso, Jiguo Yu, Jason Couture, and Xiuzhen Cheng,2016 IEEE.

**Description:**

The Internet of Things (IoT) is changing human lives by connecting everyday objects together. For example, in a grocery store all items can be connected with each other, forming a smart shopping system. In such an IoT system, an inexpensive RFID tag can be attached to each product which, when placed into a smart shopping cart, can be automatically read by a cart equipped with an RFID reader. As a result, billing can be conducted from the shopping cart itself, preventing customers from waiting in a long queue at checkout. Additionally, smart shelving can be added into this system, equipped with RFID readers, and can monitor stock, perhaps also updating a central server. Another benefit of this kind of system is that inventory management becomes much easier, as all items can be automatically read by an RFID reader instead of manually scanned by a laborer. To validate the feasibility of such a system, in this work we identify the design requirements of a smart shopping system, build a prototype system to test functionality, and design a secure communication protocol to make the system practical. To the best of our knowledge, this is the first time a smart shopping system is proposed with security under consideration.

paper 4 : “Mobile Application for Creating Presence Lists”,Zuzana VANTOVÁ, Ján PARALI, Vladimír GAŠPAR,IEEE 15th International Symposium on Applied Machine Intelligence and Informatics , 2017

**Description:**

This paper presents an application for managing education by creating presence lists. An administration application has also been created in order to allow manipulating with the content of the enumerables in the presence lists. Overviews of possible solutions are presented. Our main motivation was to provide secure and reliable way of evaluating student attendance on specific lectures. In order to accomplish this task, we utilized the NFC technology on a smartphone and student ISIC ID card, the ownership of which is compulsory for every university student at the Technical University of Košice. The application has been implemented for specific conditions at the Technical University of Košice

1. **OUTLINE OF THE PROJECT:**

The main aim of the project is to automate the daily day-to-day activities which involve manual intervention and can be automated. This could reduce the strain, in keeping track of the quantities of the daily household commodities and having to manually order them. This could ensure that none of the household commodities go out of stock.

Household items weight are measured, each items having individual weight in its own when those all full. On every day usage it may get reduced day by day. Weights of those products are monitored by load cell sensing unit. Load cell connected with microcontroller, which monitor the sensor value and calculate the weight of the product. When product weight gone below minimum level alert message is given to android mobile through Zigbee. Zigbee is connected with microcontroller and receiver Zigbee connected along with mobile by OTG cable. In user mobile receive the alert message, user placed the order based on requirement. Payment also done through mobile.

1. **EXISTING SYSTEM**

In Existing system**,** Every products of home is purchased after manually check. User check the came to when it’s almost empty its takes some time to deliver. In monthly base purchase need to take list then only purchase order raised.Although the current system keeps track of the quantities of the household products, it does not connect with the suppliers to place the order, to fully automate the process of booking.

**DRAWBACKS OF EXISTING SYSTEM:**

Although the existing system is able to measure the weight of the products it is not able to automatically place orders to the corresponding vendors. Further the existing system is unable to monitor the weight continuously.

1. **PROPOSED SYSTEM**

In our proposed system household items weight are measured, each items having individual weight in its own when those all full. On every day usage it may get reduced day by day. Weights of those products are monitored by load cell sensing unit. Load cell connected with microcontroller, which monitor the sensor value and calculate the weight of the product. When product weight gone below minimum level alert message is given to android mobile through Zigbee. Zigbee is connected with microcontroller and receiver Zigbee connected along with mobile by OTG cable. In user mobile receive the alert message, user placed the order based on requirement. Payment also done through mobile.

*ADVANTAGES:*

* Arduino simplifies the amount of hardware and software development you need to do in order to get a system running.
* The Arduino hardware platform already has the power and reset circuitry setup as well as circuitry to program and communicate with the microcontroller over USB.

1. **ARCHITECTURE DIAGRAM**

Major Parts:

* Ardiuno
* Zigbee Module
* Load cell sensors
* Power supply

**Arduino Micro-Controller**

**Power supply 5v**

**ZIGBEE**

**Load cell sensor**

Figure1: architecture of Automatic booking system WORKING

### PERIPHERALS

### Embedded Systems talk with the outside world via [peripherals](http://en.wikipedia.org/wiki/Peripheral), such as:

* Serial Communication Interfaces (SCI): [RS-232](http://en.wikipedia.org/wiki/RS-232), [RS-422](http://en.wikipedia.org/wiki/RS-422), [RS-485](http://en.wikipedia.org/wiki/RS-485) etc
* Synchronous Serial Communication Interface: [I2C](http://en.wikipedia.org/wiki/I2C), [SPI](http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus), SSC and ESSI (Enhanced Synchronous Serial Interface)
* [Universal Serial Bus](http://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB)
* Multi Media Cards (SD Cards, Compact Flash etc)
* Networks: [Ethernet](http://en.wikipedia.org/wiki/Ethernet), [Controller Area Network](http://en.wikipedia.org/wiki/Controller_Area_Network), [LonWorks](http://en.wikipedia.org/wiki/LonWorks), etc
* Timers: [PLL](http://en.wikipedia.org/wiki/PLL)(s), Capture/Compare and [Time Processing Units](http://en.wikipedia.org/wiki/Time_Processing_Unit)
* Discrete IO: aka [General Purpose Input/Output](http://en.wikipedia.org/wiki/General_Purpose_Input/Output) (GPIO)
* Analog to Digital/Digital to Analog ([ADC](http://en.wikipedia.org/wiki/Analog-to-digital_converter)/[DAC](http://en.wikipedia.org/wiki/Digital-to-analog_converter))
* Debugging: [JTAG](http://en.wikipedia.org/wiki/JTAG), [ISP](http://en.wikipedia.org/wiki/In-System_Programming), [ICSP](http://en.wikipedia.org/wiki/In_Circuit_Serial_Programming_%28ICSP%29), [BDM](http://en.wikipedia.org/wiki/Background_Debug_Mode_interface) Port, BITP DP9 port

**SENSORS**

Sensors are devices (usually electro-mechanical) which help us measure a physical parameter (such as temperature, pressure, force, acceleration etc.) by providing a signal that either quantitatively measures (level) that physical parameter or provides a simple binary signal that indicates a yes/no signal that tells us if something occurred or not (such as a touch sensor). Most sensors require power to be provided to a sensing element and an electrical signal is then generated after the measurement.

**ZIGBEE**

ZigBee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless M2M networks. The ZigBee standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz.

The 802.15.4 specification upon which the ZigBee stack operates gained ratification by the Institute of Electrical and Electronics Engineers (IEEE) in 2003. The specification is a packet-based radio protocol intended for low-cost, battery-operated devices. The protocol allows devices to communicate in a variety of network topologies and can have battery life lasting several years.

**ARDUINO**

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world.

The project is based on microcontroller board designs, manufactured by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers.

The first Arduino was introduced in 2005, aiming to provide an inexpensive and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

**EMBEDDED HARDWARE INTERFACE:**

In this module, Embedded Hardware is connected with Load cell unit to measure the load of the object along with the Wifi Module which is used for communication. We use Arduino micro controller as the micro controller for effective hardware processing. Load cell unit is used for analysis of the weight of the corresponding item / object.

**ZIGBEE COMMUNICATION:**

ZigBee is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless M2M networks. The ZigBee standard operates on the IEEE 802.15.4 physical radio specification and operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz . The ZigBee protocol was designed to provide an easy-to-use wireless data solution characterized by secure, reliable wireless network architectures. Support for multiple network topologies such as point-to-point, point-to-multipoint and mesh networks.

* Low duty cycle – provides long battery life
* Low latency
* Direct Sequence Spread Spectrum (DSSS) Up to 65,000 nodes per network
* 128-bit AES encryption for secure data connections, Collision avoidance, retries and acknowledgements.

1. **CONCLUSION**

Thus the project infer that automatic item booking using IOT. System will automatically book the items if that item is going to empty. So for that we are implementing automatic booking system for the house hold items. We are including android application for the purpose to know the booking details.

**REFERENCES**

[1] “An Improved Real-Time Surveillance System for Home Security System using BeagleBoard SBC, Zigbee and FTP Webserver”Author:Rakesh V S,Sreesh P R Sudhish N George, 978-1-4673-2272-0/12/$31.00 ©2012 IEEE

[2] “Automatic LPG Gas Cylinder Booking Software and Weight Measurement Using Load Cell & GSM.”*IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)* Prof.P.S.Sonawane, Darade Pooja, Kankrale Pratiksha, Shah Rozmin

[3] “Energy Efficient Solar Based Digital Electronic Weighing Machine”Sandip N. Rikame,Student, E&TC Department Pradip W. Kulkarni ,HOD, E&TC Department 2014 5th International Conference on Computer and Communication Technology.

[4] “IoT applications on Secure Smart Shopping System”Ruinian Li1, Tianyi Song1, Nicholas Capurso, Jiguo Yu, Jason Couture, and Xiuzhen Cheng2016 IEEE.

[5] “Mobile Application for Creating Presence Lists”Author:Zuzana VANTOVÁ\*, Ján PARALISAMI 2017 • IEEE 15th International Symposium on Applied Machine Intelligence and Informatics • January 26-28, 2017 • Herl’any, Slovakia

**REFERENCES**

[1] “An Improved Real-Time Surveillance System for Home Security System using BeagleBoard SBC, Zigbee and FTP Webserver”Author:Rakesh V S,Sreesh P R Sudhish N George, 978-1-4673-2272-0/12/$31.00 ©2012 IEEE

[2] “Automatic LPG Gas Cylinder Booking Software and Weight Measurement Using Load Cell & GSM.”*IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)* Prof.P.S.Sonawane, Darade Pooja, Kankrale Pratiksha, Shah Rozmin

[3] “Energy Efficient Solar Based Digital Electronic Weighing Machine”Sandip N. Rikame,Student, E&TC Department Pradip W. Kulkarni ,HOD, E&TC Department

2014 5th International Conference on Computer and Communication Technology

[4] “IoT applications on Secure Smart Shopping System”Ruinian Li1, Tianyi Song1, Nicholas Capurso, Jiguo Yu, Jason Couture, and Xiuzhen Cheng2016 IEEE.

[5] “Mobile Application for Creating Presence Lists”Author:Zuzana VANTOVÁ\*, Ján PARALISAMI 2017 • IEEE 15th International Symposium on Applied Machine Intelligence and Informatics • January 26-28, 2017 • Herl’any, Slovakia