1. INTRODUCTION

In metro cities purchasing shopping at super markets and big malls is a daily activity. We have seen big lines for payment of bill at malls on holidays and weekends. When there are special offers and discounts rush is also even more. Customers will purchase many items and put it on to the trolley. After customers done the purchase they need to go to the billing counter for payment. At the billing counter the bill is prepared for the customer using various existing technologies like barcode or manually bill is generated.

1.1 Purpose

In today's competitive world time is the most important resource. No customer is ready to wait for hours together in the line just to get their products billed. This gives a need to develop a time saving and advanced shopping cart system with which the customer enjoys his shopping experience and it's also helpful for management with which they can have a smaller number of employees at the billing counter

1.2 Scope

Our product is taking help of two different technologies viz. ARDUINO and RFID, which is most efficient when compared with other technologies used. ARDUINO processes the data internally and serves as micro controller for the entire system. RFID is frequency-based technology which is 25 times faster than any existing technologies that are being used for the same purpose.

1.3 Features

Our proposed system overcomes all 5the problems in existing system and provides high speed billing efficiency. This is a perfect/optimal solution for saving one's time during shopping. The features of our product are briefly described below:

1. A Completely automated system helps the customer to use the product with ease.

2. When this is applied on large the management can earn profits with reduced labour cost.

3. The customer need not wait all through the long lines for billing but can happily walk out of the store by paying the amount he is supposed to.

2. SOFTWARE REQUIREMEMT SPECIFICATION

2.1 SRS:

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase).

The SRS phase consists of two basic activities:

1. **Problem/Requirement Analysis:**

   The process deals with understanding the problem, goal and constraints.

2. **Requirement Specification:**

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirements phase terminates with the production of the validate SRS document producing the SRS document is the basic goal of this phase.

Document Conventions:

We have used Times New Roman (text size 12). Bold font is used for Main headings (text size 16). Times New Roman font is used for sub headings (text size 14).

Font: Times New Roman

Main Heading: Bold Font

3. EXISTING SYSTEM

There are existing systems for the product we have developed such as:

Barcode System

A **barcode** (also **bar code**) is an optical, machine-readable, representation of data; the data usually describes something about the object that carries the barcode. Traditional barcodes systematically represent data by varying the widths and spacings of parallel lines, and may be referred to as linear or one-dimensional (1D). Later, two-dimensional (2D) variants were developed, using rectangles, dots, hexagons and other geometric patterns, called matrix or 2D barcodes, although they do not use bars as such. Initially, barcodes were only scanned by special optical scanners called barcode readers.

All the super markets and shopping malls uses barcode system to bill the products that are purchased by the customers. But with the growing customers and sales barcode system fails as the barcode readers are very slow. They can only read one product at a time which makes the entire billing process slow and lot of time of customers is wasted.

DISADVANTAGES:

The disadvantage using the traditional barcode system for billing area

- Lack of shopping experience
- Absence of online billing system

4. PROPOSED SYSTEM

Our proposed system overcomes all the problems in existing system and provides high speed billing for your shopping. This is a perfect/optimal solution for saving your time and energy. RFID reader is modeled with micro controller. RFID tags are associated with products in super markets and shopping malls.

Each category of product is associated with two tags; one for adding the product to cart, the other for removing the product from the cart. Once the shopping is done by the customer the bill is automatically generated based on the unique RFID card id.

Barcodes are designed to be scanned one at a time whereas many RFID tags can be scanned at once. Barcodes require that the scanner maintain a line-of-sight with each code, while RFID is a “near field” technology, so the scanner only needs to be within range of the tag to read it.

RFID is an acronym for “radio-frequency identification” and refers to a technology whereby digital data encoded in RFID tags or smart labels (defined below) are captured by a reader via radio waves.

For the more enhancement of product, the bill is automatically displayed on the mobile application which is connected to cloud with Google Firebase.

5. TECHNOLOGIES USED

Software Interfaces:

We are using Arduino IDE and programming language for writing the project code. Operating system will be Windows.

a. Software Requirements:

1. Arduino IDE version 1.8.

2. Operating System: Windows

b. Hardware Requirements:

1. RAM: 1GB Ram or above

2. Hard Disk: 80GB or above

3. Processor: Pentium IV 1GHz or above.

4. RFID Reader 13.56MHz RC522

5. RFID Cards

6. RFID Tags

7. Shopping Basket

8. Arduino board: Arduino UNO

9.ESP8266(Wi-Fi Module)

6. ARCHITECTURE

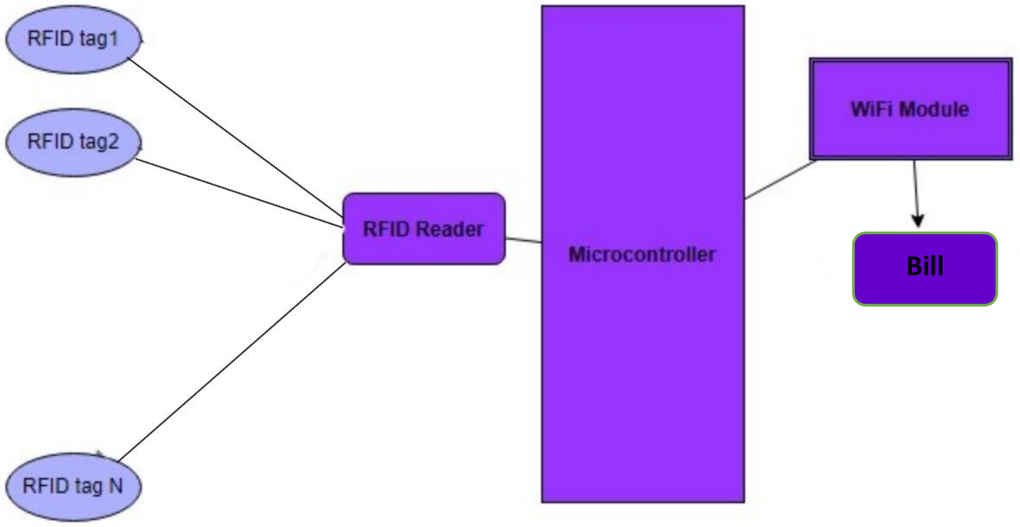


Fig: 6.1 Architecture Diagram for Smart Shopping Cart

The Fig:6.1 shows the architecture of the smart shopping cart with its interfaces. The Arduino microcontroller centrally controls all the devices connected to it. The program has to beefed into the Arduino board and connected to the power supply. The devices are triggered according to the flow of the code. Firstly, the data regarding the prices of products are to be stored on RFID tags. All the authorized RFID cards data is to be stored and code must be written to check the authentication. Upon the user purchasing the product he adds it to his cart by reading the tag. Each product has two tags to serve the following purpose

1.1 **Add Tag:** when this tag is read in the proximity of the reader the cost of the product is added to your bill.

1.2 **Subtract Tag:** when this tag is read in the proximity of the reader the cost of the product is subtracted from your bill i.e. if you wish not to buy any product that is added to your cart you can use this tag.

7. DESIGN MODULES

7.1 Detailed Activity diagrams:

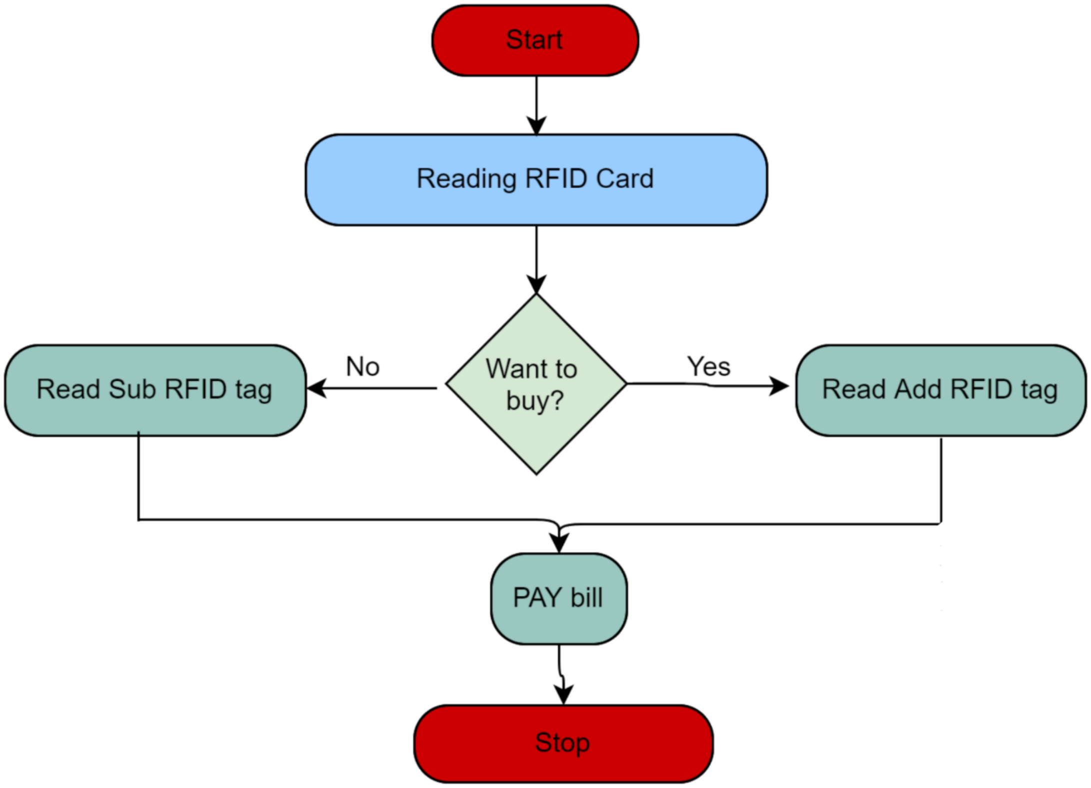


Fig: 7.1.1 Activity Diagram

The Fig:7.1.1 shows the architecture of the smart shopping cart with its interfaces. The Arduino microcontroller centrally controls all the devices connected to it. The program has to beefed into the Arduino board and connected to the power supply. The devices are triggered according to the flow of the code. Firstly, the data regarding the prices of products are to be stored on RFID tags. All the authorized RFID cards data is to be stored and code must be written to check the authentication. Upon the user purchasing the product he adds it to his cart by reading the tag. Each product has two tags to serve their purposes.

7.2 USECASE DIAGRAM

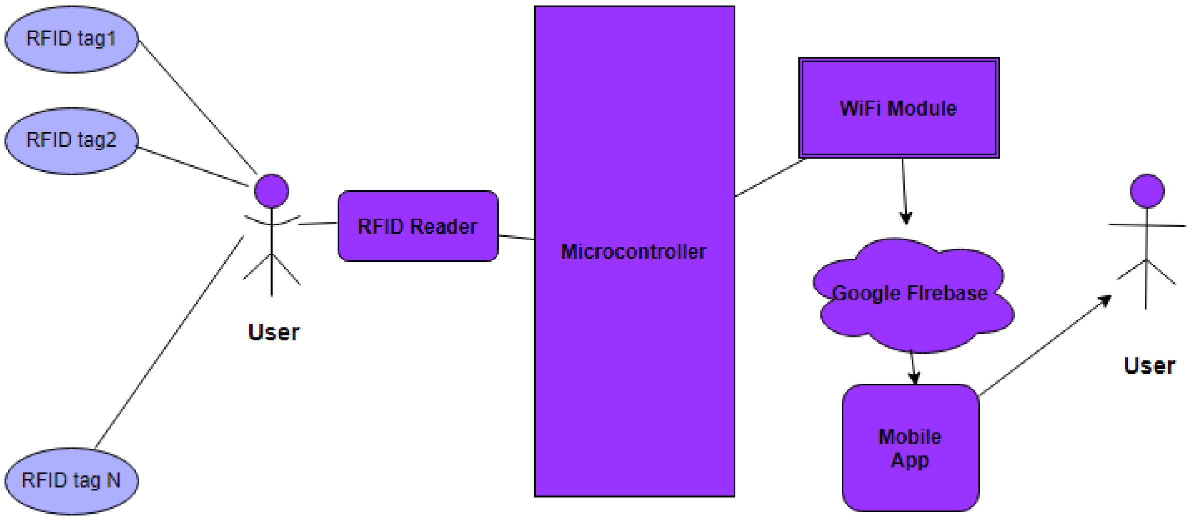


Fig: 7.2.1 Usecase Diagram

The Fig:7.2.1 shows how the smart shopping cart system works. All the products of the shopping mall will have two tags which can be used by the user to add a product to the cart and remove a product from the cart. User add or remove the product by reading the RFID card to the reader. The data stored in the tags ist thus sent to the microprocessor for processing where the price and details of the product are added to the bill which is stored on google firebase using server that is connected microcontroller using ESP8266 WiFi module .

8.IMPLEMENTATION

8.1 IOT Technology:

The Internet of Things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect and exchange data, creating opportunities for more direct integration of the physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions

8.1.1 Internet of Things:

The Internet of Things (IOT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

8.1.2 Arduino:

Fig:8.1.2.1 Arduino UNO

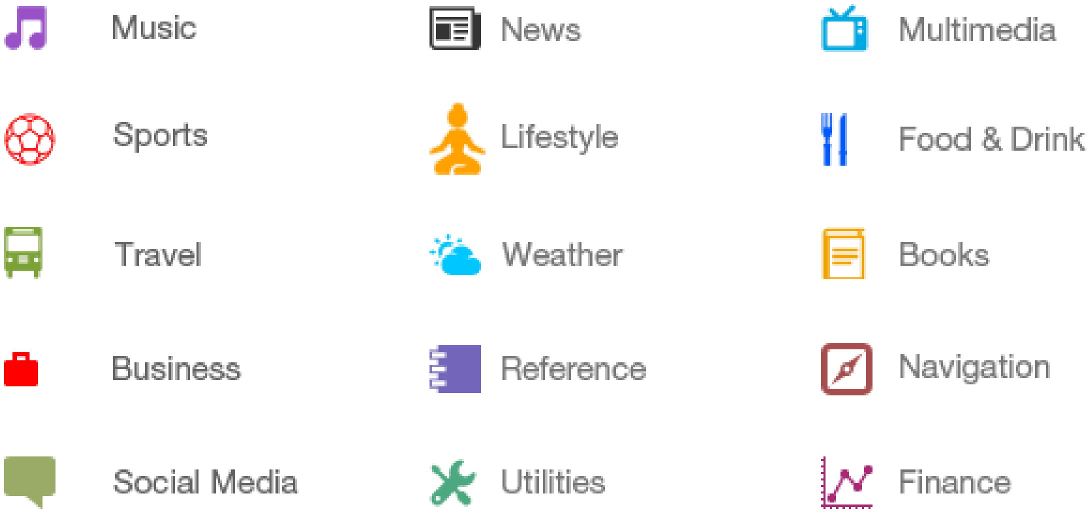
Arduino is an open-source prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing, Max MSP).

The boards can be built by hand or purchased preassembled; the software can be downloaded the hardware reference designs (CAD files) are available under an open-source license, you are free to adapt them to your needs.

Arduino received an Honorary Mention in the Digital Communities section of the 2006 Ars Electronica Prix. The Arduino founders are: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis.

8.1.3 Categories of Arduino Application

There are many Arduino applications in the market. The top categories are:



Music	News	Multimedia
Sports	Lifestyle	Food & Drink
Travel	Weather	Books
Business	Reference	Navigation
Social Media	Utilities	Finance

8.2 Software development

8.2.1 Arduino - Environment Setup

We can start our Arduino application development on either of the following operating systems

- Microsoft Windows XP or later version.
- Mac OS X 10.5.8 or later version with Intel chip.

Second point is that all the required tools to develop Arduino applications are freely available and can be downloaded from the Web. Following is the list of software's to start Arduino application programming.

- ARDUINO IDE or later versions
- RFID library

8.3 RFID Technology:

Radio-frequency identification (**RFID**) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically-stored information. Passive tags collect energy from a nearby RFID reader's interrogating radio waves. Active tags have a local power source (such as a battery) and may operate hundreds of meters from the RFID reader. Unlike a barcode, the tag need not be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method for Automatic Identification and Data Capture (AIDC).

RFID tags are used in many industries, for example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line; RFID-tagged pharmaceuticals can be tracked through warehouses; and implanting RFID microchips in livestock and pets allows for positive identification of animals.

Since RFID tags can be attached to cash, clothing, and possessions, or implanted in animals and people, the possibility of reading personally-linked information without consent has raised serious privacy concerns

8.4 Device construction steps:

Step 1: Solution and Module Preparing.

Step 2: Hardware Connection.

Step 3: Dump the code into Arduino and run.

Step 4: Get and Store product data on RFID tags

Step 5: Make the System Work and System Installation.

8.4.1 Screenshots:

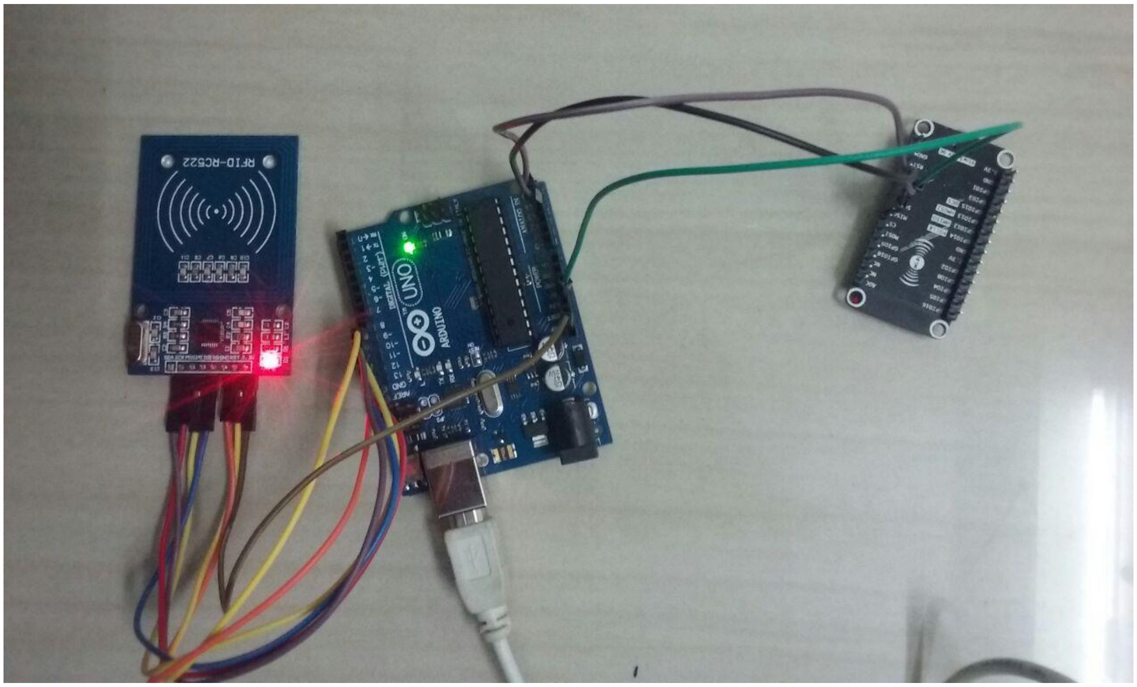


Fig:8.4.1.1 Basic Circuit diagram

The basic circuit diagram with all the connections made to Arduino, RFID reader and node MCU. Here the microprocessor is connected with the RFID reader and also with the ESP8266 also called as Wi-Fi module which is used to connect the entire circuit to internet so that it is cloud connected.

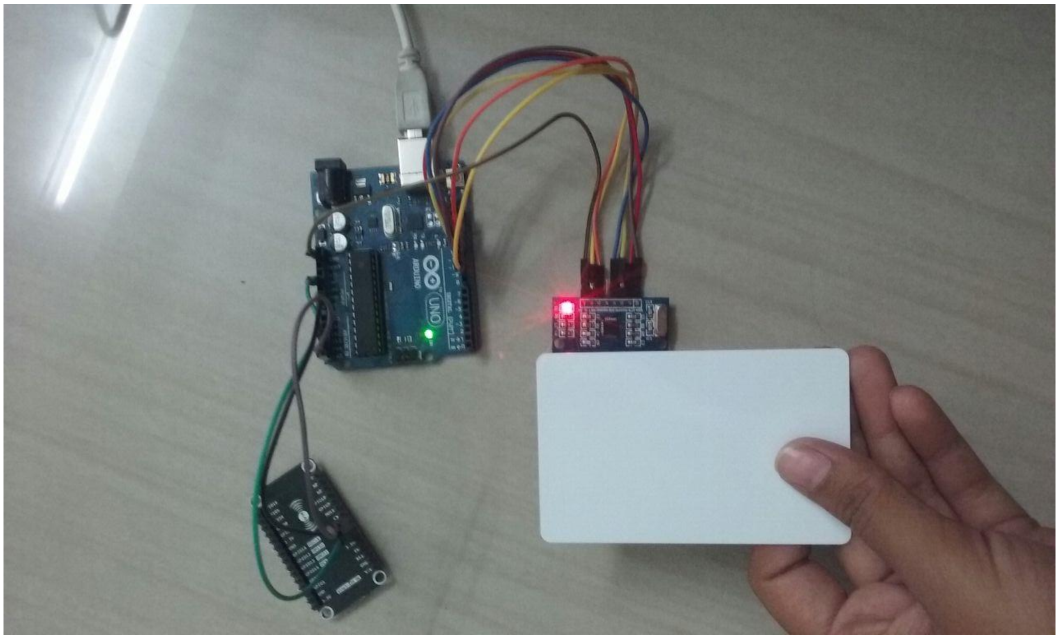


Fig:8.4.1.2 Reading RFID Card

Starting the shopping by reading the RFID card to the RFID reader. When the customer enter the shopping mall he is given a RFID card with which he can start his shopping simply by reading the card given to him to the RFID reader present to the basket.

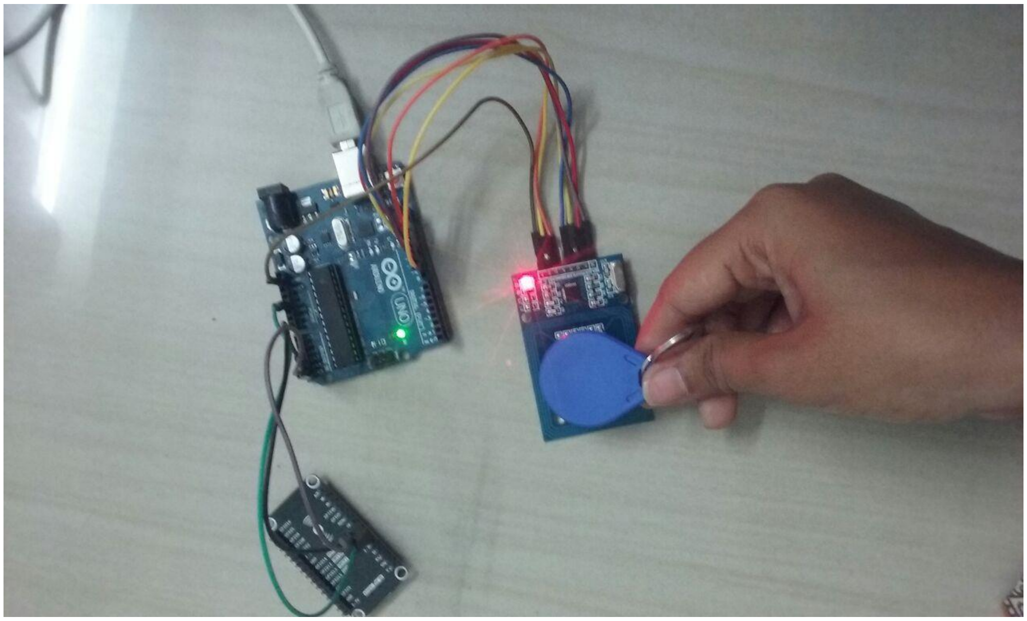


Fig:8.4.1.3 Reading Add Tag

Adding a product to cart by reading the RFID Add Tag. After the customer starts the shopping he can read the add tag of the product that we wants to buy with which the cost of the product is added to his bill.

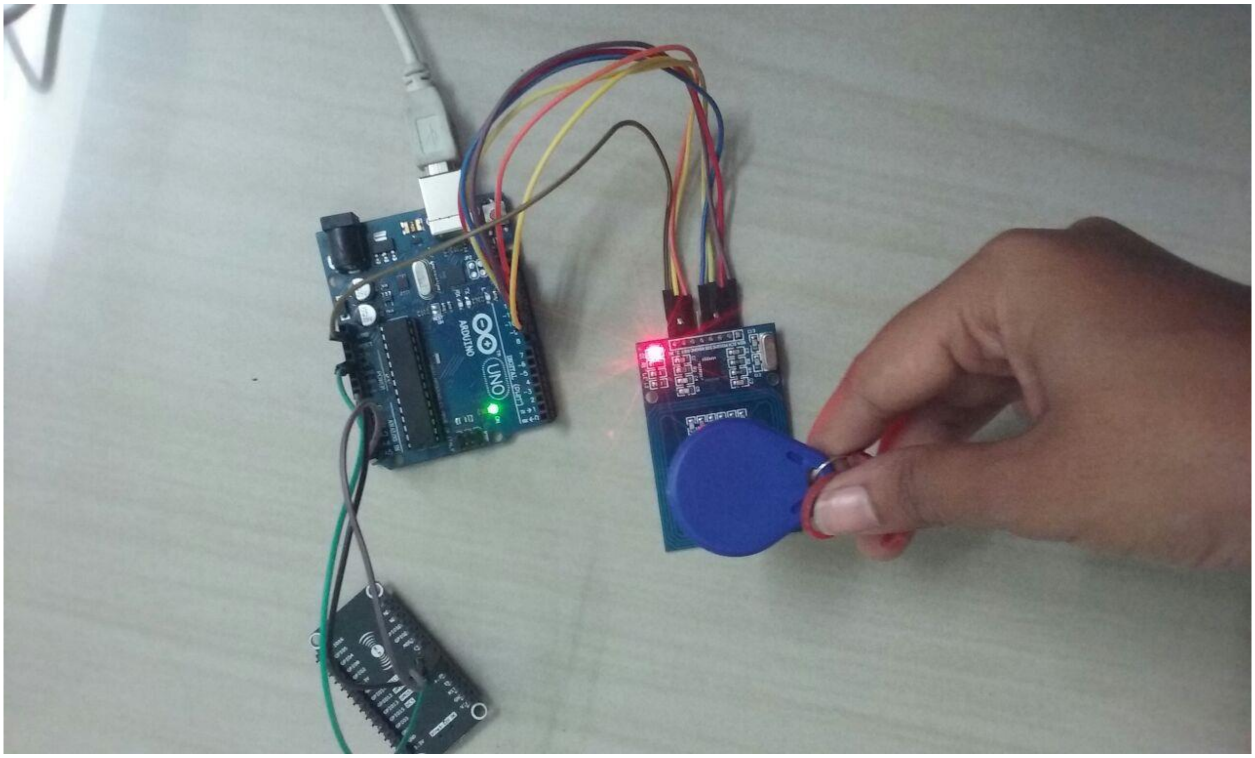


Fig:8.4.1.4 Reading Subtract Tag

Removing a product from the cart by reading RFID Sub Tag. If the customer wants to delete an item from bill or doesn’t want to buy an item that is added to the cart, he can do that by reading the sub tag of that product to RFID reader with which the cost of the item is subtracted from the bill.

9. SOURCE CODE

ARDUIONO:

```
#include <Wire.h>
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
static int bill=0;
MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup() {
 Wire.begin(8);
  SPI.begin();      // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522
  Serial.println("Approximate your card to the reader...");
  Serial.println();/* join i2c bus with address 8 */
 Wire.onReceive(receiveEvent); /* register receive event */
 Wire.onRequest(requestEvent); /* register request event */
 Serial.begin(9600);           /* start serial for debug */
}

void loop() {
   if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //Show UID on serial monitor
  Serial.print("UID tag :");
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
     Serial.print(mfrc522.uid.uidByte[i], HEX);
     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  Serial.println();
  Serial.print("Message : ");
  content.toUpperCase();
  if (content.substring(1) == "C0 99 02 25") //change here the UID of the card/cards that you want to give access
  {
    //Wire.write("a");
    Serial.println("welcome!");

    Serial.println("Your shopping started");

    Serial.println();

    delay(3000);

  }

  else if (content.substring(1) == "B4 02 56 D3") //change here the UID of the card/cards that you want to give access

  {

    Serial.println("name: dairy milk ");

    Serial.println("cost: 25");

    Serial.println("Added");

    Serial.println();

    bill=bill+25;

    delay(3000);

  }

    else if (content.substring(1) == "D5 1F D2 83") //change here the UID of the card/cards that you want to give access

  {

    Serial.println("name: dairy milk ");

    Serial.println("cost: 25");

    Serial.println("Removed");

    Serial.println();

    bill=bill-25;

    delay(3000);

  }
```

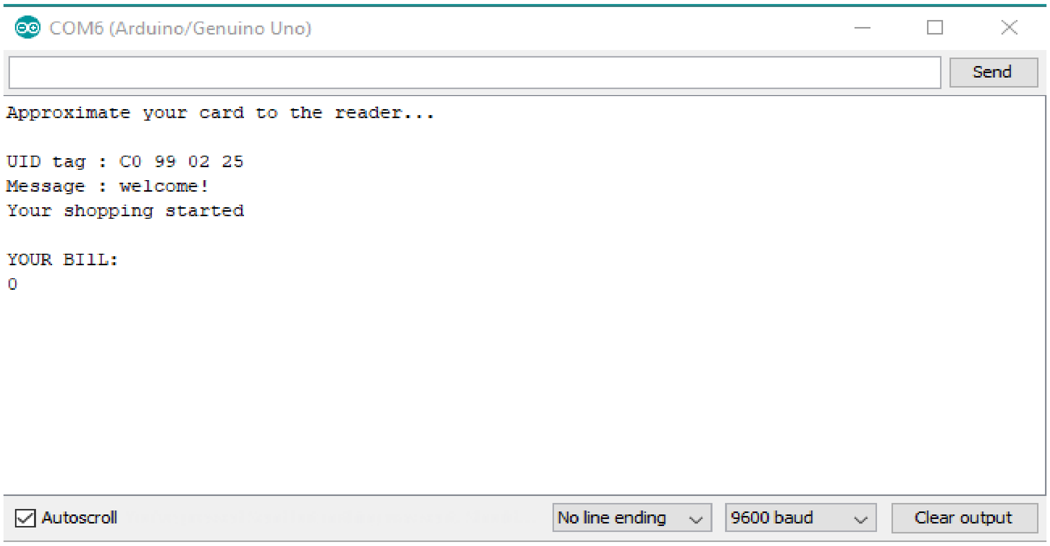
```
 else   {
    Serial.println(" Access denied");
    delay(3000);
  }
  Serial.println("YOUR BIlL:");
  Serial.println(bill);

 delay(100);
}

// function that executes whenever data is received from master
void receiveEvent(int howMany) {
 while (0 <Wire.available()) {
    char c = Wire.read();      /* receive byte as a character */
    Serial.print(c);           /* print the character */
  }
 Serial.println();             /* to newline */
}

// function that executes whenever data is requested from master
void requestEvent() {
 Wire.write(bill);  /*send string on request */
}
```

*********SECOND***********

```
#include <Wire.h>
   // Create MFRC522 instance.


#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#define FIREBASE_HOST "smartshoppingcart-9abcd.firebaseio.com"
#define FIREBASE_AUTH "QPeoWr3AypPB2aaxil0oV6su7USEOvRJBEFOIPE5"
#define WIFI_SSID "PG Lab"
#define WIFI_PASSWORD "Pg@105##"
//int n=0;
void setup() {
  //Serial.begin(9600);   // Initiate a serial communication


 Serial.begin(9600);/* begin serial for debug */
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());


  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
 Wire.begin(D1, D2); /* join i2c bus with SDA=D1 and SCL=D2 of NodeMCU */
}

void loop() {// Look for new cards


   // set value
 int d;
 Wire.beginTransmission(8); /* begin with device address 8 */
 Wire.write("Hello Arduino");  /* sends hello string */
 Wire.endTransmission();    /* stop transmitting */


 Wire.requestFrom(8, 13); /* request & read data of size 13 from slave */
 if(Wire.available()){
    char c = Wire.read();
     d=(int)c;
  Serial.print(d);
 }
  Firebase.setFloat("bill", d);
  // handle error
  if (Firebase.failed()) {
      Serial.print("setting /number failed:");
      Serial.println(Firebase.error());
      return;
  }
 Serial.println();
 delay(1000);
}
```

10.OUTPUT SCREENS

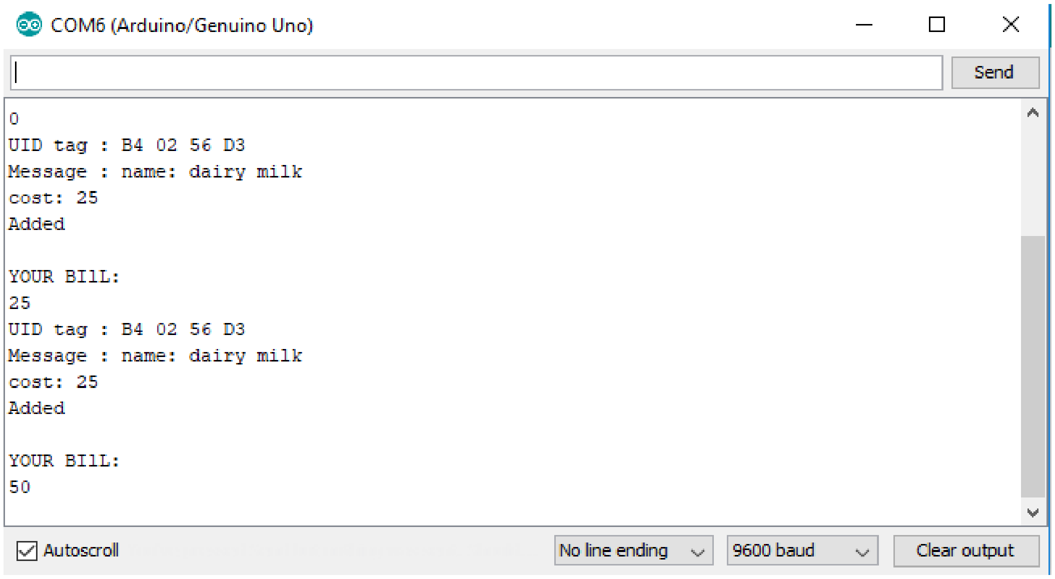
10.1 Serial monitor:



```
Approximate your card to the reader...

UID tag : C0 99 02 25
Message : welcome!
Your shopping started

YOUR BILL:
0
```

Fig:10.1.1 Starting the shopping

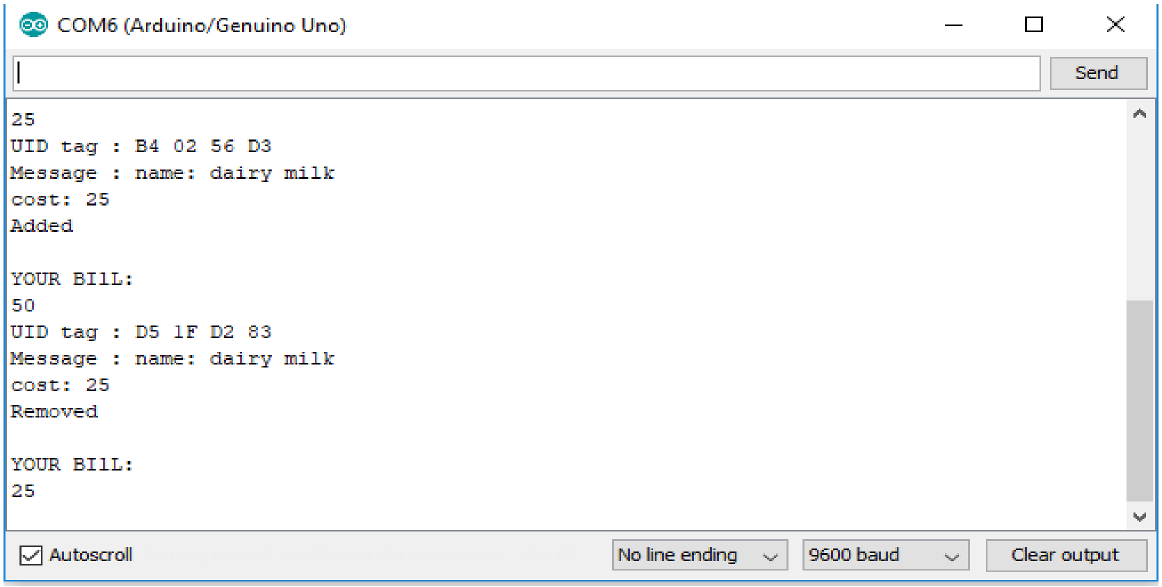
When the user enters and he reads the RFID card his shopping starts and the bill for his shopping is added to that card ID



```
0
UID tag : B4 02 56 D3
Message : name: dairy milk
cost: 25
Added

YOUR BILL:
25
UID tag : B4 02 56 D3
Message : name: dairy milk
cost: 25
Added

YOUR BILL:
50
```

Fig:10.1.2 Adding products to cart

When a product is added to the cart it is added to bill by reading the Add tag against the RFID reader placed to basket



```
25
UID tag : B4 02 56 D3
Message : name: dairy milk
cost: 25
Added

YOUR BILL:
50
UID tag : D5 1F D2 83
Message : name: dairy milk
cost: 25
Removed

YOUR BILL:
25
```

Fig:10.1.3 Displaying the bill after removing item

When a product is removed from the cart it is added to bill. When the customer wants to remove a product from the cart he can and the cost is deducted from the bill.

10.2 Application Screenshots:

First Screen:

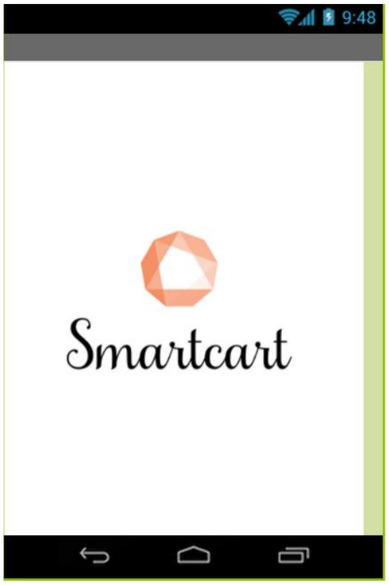
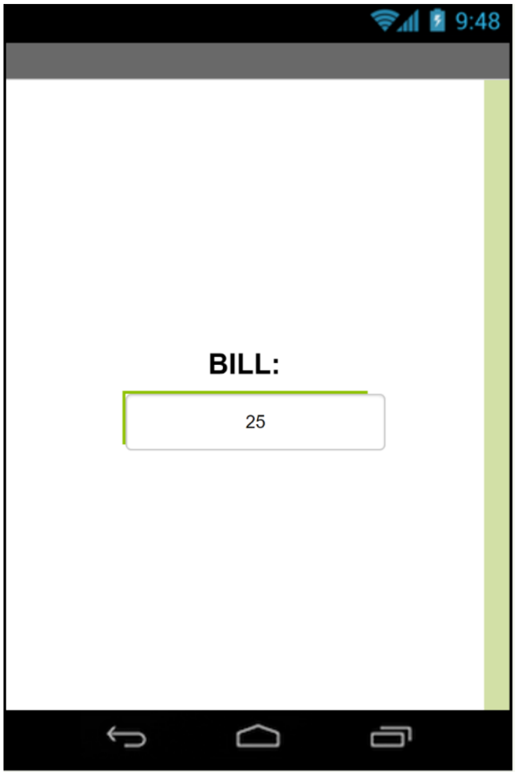


Fig:10.2.2 Home Screen

The first screen of the application shows the logo of the application. Name of our application is "Smartcart"

Second Screen:



BILL:

25

Fig: 10.2.2 Bill Screen

The second screen of the application shows the bill for the products the customer has purchased. This will be updated when the customer buys a product or removes a product from his cart.

11.TESTING

a. Introduction:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

a. Types of Testing:

i. Unit Testing:

Unit Testing is essentially for the verification of the code produced during the coding phase and the goal is to test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not. Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two different phases.

ii. Integration Testing:

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.

iii. Functionality Testing:

The program is successfully loaded and executed with no execution errors. All the modules of whole application are tested. The complete performance of the project “Smart Shopping Cart ” is good.

iv. System Testing:

System Testing is the testing of a complete and fully integrated software product. Usually software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

v. Interface Testing:

A connection that integrates two components are called interface. This interface in a computer world could be anything like API's, web services, etc. Testing of these connecting services or interface is referred as Interface Testing.

Interface is actually software that consists of sets of commands, messages, and other attributes that enable communication between a device and a user.

12. DEPLOYEMENT

12.1 Deployment model:

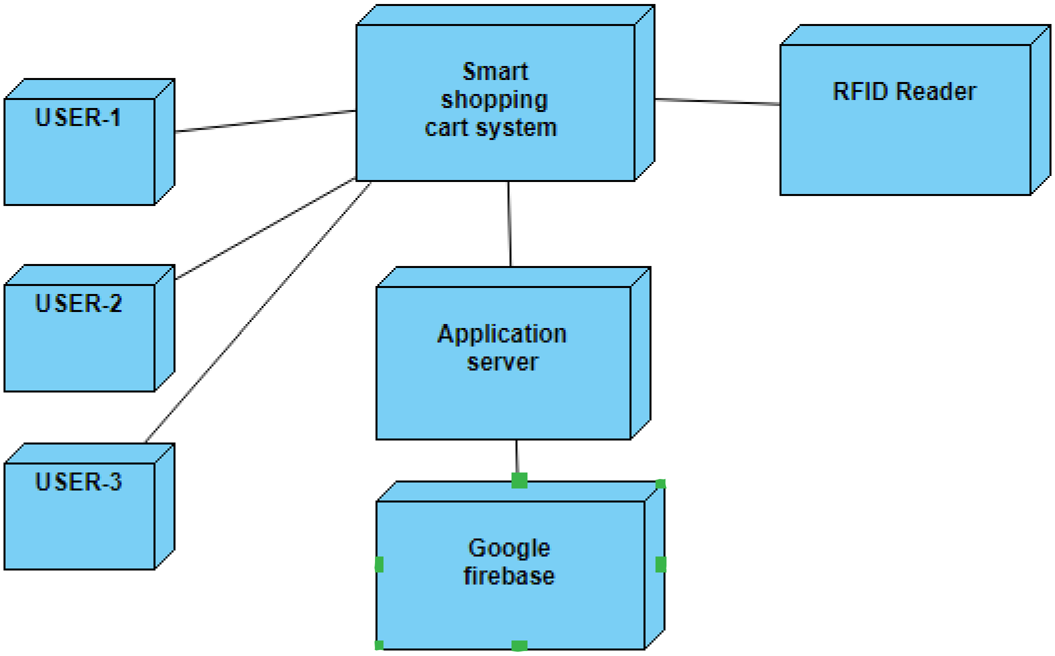


Fig: 12.1.1 Deployment Model

12.1.1 Smart shopping cart:

'Smart Shopping Cart'- The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. We are implementing an RFID based Smart Shopping Cart in the field of retail merchandise.

Our whole shopping experience is often marred by the long checkout lines. Soon we can end this problem by replacing the ubiquitous Universal Product Code (UPC) bar code by smart labels, known as radio frequency identification (RFID) tag. The key idea here is to aid in everyday shopping in terms of reduction in time spent, eliminating the daily hassle of locating the right product and standing in long lines.

The primary goal is to provide a technology oriented, reduced cost, time saving, hassle free, commercially oriented system for an enhanced shopping experience. It can be implemented with Arduino Uno as microprocessor and RFID technology through cloud.

12.1.2 Application server & Google Firebase:

For the cloud storage we are using Google Firebase where the bill details generated dynamically are stored after the processing is done through microcontroller and after the data is sent to cloud using Wi-Fi module

Wi-Fi module takes the data from Arduino and sends the data to google firebase after establishing connection with server.

13. CONCLUSION

Hence the Smart Shopping Cart provides a best solution for the long queue lines the shopping malls. Our proposed system is designed in such a way that it overcomes all the defects of previously available systems. User friendliness is given much importance in our system which makes it more comfortable to use than any other existing systems. Also, it is built with open source hardware which makes it cheaper and more efficient for the users. There are existing systems for the product we have developed but most of them are using embedded-systems which is costly to build, we are using Arduino which is very cost efficient and easy for further updating. Finally, time the most important resource for every individual today, there are a lot of solutions available for these issues like amazon GO and complete solution for this problem is not yet discovered.

14. BIBLIOGRAPHY

- https://www.electronicshub.org/iot-project-ideas/
- https://en.wikipedia.org/wiki/Internet_of_things
- http://www.instructables.com/id/Controlling-Home-Appliances-Using-Node- MCU-Via-MQT/
- https://en.wikipedia.org/wiki/Radio-frequency_identification
- http://www.electronicwings.com/nodemcu/nodemcu-i2c-with-arduino-ide
- References:

  1. Arshdeep Bahga and Vijay Madisetti, Internet of Things A Hands -on approach, Universities Press,2015.

  2. C programming for Arduino by Julien Bayle