



FYP FINAL REPORT

2019/2020

AUTOMATIC SHOPPING CART (CARTIGO)

GROUP MEMBERS:

AHMED ALI RAZA (9718)

BILAL AHMED (14444)

HAMZA ARIF (14446)

SUPERVISOR:

RIDAH FATIMA MUDASSIR.

COORDINATOR:

DR AARIJ MEHMOOD HASSAAN.

Chapter 1:

Introduction

1. Aim & Objectives

The aim and objectives are to develop RFID based billing system for supermarkets in order to make billing process convenient and easy. Implementing an Automatic shopping cart using RFID technology that will be saving time of customers and improving purchasing. In this RFID card is utilized by the RFID reader in the shopping cart when the customer wants to add product the cost of the product will be shown and the total amount of bill will display on the LCD, when the customer wants to remove the product from the Cart, you need to take product out from the Cart, the amount of that product by scanning it again and gets deducted from the total amount. After customer finished shopping, the customer will press send button on the hardware device and the bill will be generated in the database which could be taken by providing the Cart ID trolley number. The main purpose of this system is to make it effectively adaptable for helping the customers, time will be saved at the billing counters avoiding the long waiting queues.

2. Relevance of domain

Today shopping is becoming a time consuming, hectic activity in cities. There are long lines in marts on weekend/events in big cities. Therefore, at different marts because of this, after shopping the customer reach billing counter for bill but since using bar code for adding product and calculating bill that is very time consuming and that increases the waiting queues for Bill. The ultimate goal is to develop a system consisting of a hardware device with a software that can be used in super marts to resolve the long queues at billing counter using RFID Technology using in the super marts with single and multiple cash counters.

3. Research Question:

Q: How do you like our designed Project Trolley?

Good designed, yes brilliant idea, a very good layout but still equipment attached is too big, it depends on its function, if this place is just to scan you can better place under trolley. Is that good innovative, it's cool.

Q: In Your Opinion where in the Trolley we should Fixed the Hardware?

Handle

Front side

Top (at handle)

Q: Do you Want anything to be improved in our Hardware you Can share your Views With us we will try to Improve?

Go for touchscreen technology like iPad
Hide the circuit, improve hardware design means look
Hardware looks good but either it should be compact to place in front otherwise it must be installed down as prototype it's fine, but for commercial purpose you need more durable and neater. Good work, so far the design is good, hardware should be small physically You can wrap your hardware well and fit it like it is the part of the trolley so If the system under the basket than the system surely safe, it is enough to meet requirement

Q: Do you Think that our Project will have any Impact in Our Super marts and Will Reduce Time at Billing Counter?

38.9%

61.1%

Q: Do you think In Supermarts like Chase up, Metro Cash & Carry, Imtiaz super store, Shazz, Bin Hashim, Bin Safeer and others etc. they took more time in Billing counter?

No=11.1%

Yes=83.3%

Q: Do you think In Super marts Our project will work to reduce time in Billing counter?

Yes=94.4%

Q: In Which Supermarket you would Recommend for Using Our Project (CARTIGO) will be Useful in Reducing Time? Name & Reason?

Carrefour, Make more reliable this product. Imtiaz super market and chassio, Imtiaz due to customer inflow, Big ones. In large supermarket like imtiaz there are many crowds came every day, Imtiaz, Metro, chase etc.

Q: How much Percentage (%) In Supermart Using Our Project (CARTIGO) will be Useful in Reducing Time?

20% to 30%

40% to 50%

60% to 70%

1 (5.6%)

4 (22.2%)

13 (72.2%)

4. Report Structure

Chapter 1:

- Introduction.
- Relevance of Domain.
- Research Questions.

Chapter 2:

- Technology Background.
- Detail background of the technology
- Discuss the problem domain with historical background
- similar solutions and compare with yours
- Summary of 4-5 research papers related to the technology.

Chapter 3;

Requirements & Methodology

- What logic, Constraints, platform and strategy you would use to implement requirements?
- Justify why you would use these strategies or platform or language?
- Project Flow Diagram.
- Security Alarm (Project Flow Diagram).
- Project Architecture
- Features
 - Functional requirements.
 - Non-Functional requirements.

Chapter 4;

Project Plan & Initial Design

- Gantt Chart.
- Diagrams
 - Use Case Diagram,
 - Activity Diagram
 - Class Diagram
- Conceptual ERD Diagram,
- Add Prototype (blueprint) or initial design of your UI, which was made before starting the project.

Chapter 5; Project Design & Development

- **Design**
 - Front End Design:
 - Color Theme
 - Logo
 - Template
 - Screenshots of the User Interface with Description.
 - Backend Design: this includes physical ERD of the database designed directly using DBMS after implementing the conceptual design. It must have column types and sizes.
- **Development**
 - Discuss different stages of development, add code or any important logic you want to discuss, discuss code you used to design layouts and implement logic.
 - Libraries
 - Languages
 - Platform used to develop the project,
 - Add important database queries, update database logics.
 - Add/delete/update/bill ID/save total
 - search products /search bills
 - Screenshots of a sample run of your project

Chapter 6;

- 1. Test Plans**
 - Purpose
 - Test Objectives
 - Scope
- 2. Testing Methodologies**
 - Black box testing
 - White box testing
- 3. Level of Testing**
 - Unit testing
 - Modules tested
 - Integration testing
 - Functional testing
 - Regression testing
 - Hardware testing
 - Performance testing
 - Usability testing
 - GUI testing
 - Compatibility testing
 - System testing

4. Test Cases Factors

- Test entry/exit criteria
- Test deliverables
- Test suspension/resumption criteria
- Test environmental/staffing/training needs

5. Test Cases

Chapter 7;

- Conclusion
 - Future Work
-

Chapter 2; Technology Background

Hardware Specification:

Arduino-Kit:

This is an Arduino pro mini AT mega328 5V 16M compatible board. A microcontroller board based on AT mega168 used as Arduino in hardware.

Node MCU:

NodeMCU is an open source firmware, development kit that helps you build prototype or build IoT product. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266.

RFID Reader: A radio frequency identification **reader (RFID reader)** is a device used to read information from an RFID tag associated with object, which is used to read tags no# on objects. RFID reader is like transceiver and receiver with the use of radio frequency signals.

RFID Tags: Tags are important part of RFID system, because they store the information of the object being tracked. Object information, which has UID is stored in the memory of tags and is accessed via the radio signal of RFID readers. There are different types of Tags.

- Active
- Semi-passive
- Passive

Passive RFID Tags have no internal power supply since we are using passive tags. Passive tags signal by backscattering method from reader. Passive tags have practical read distances ranging from about 11 cm up to 10 meters.

Passive Tags (Ranges):

- LF: 125 kHz – 134.2 kHz,
- HF: 13.56 MHz,

- UHF: 860 MHz – 960 MHz,
- SHF: 2.45 GHz.

Frequency Ranges	LF 125 KHz	HF 13.56 MHz	UHF 868-915 MHz	Microwave 2.45 GHz & 5.8 GHz
Read Range (Passive Tags)	Shortest 1"-12"	Short 2"-24"	Medium 1'-10'	Longest 1'-15'
Data Rate	Slower	Moderate	Fast	Faster
Ability to read near metal or wet surfaces	Better	Moderate	Poor	Worse

Advantages of Passive Tags:

- Size is Optimized
- Lesser Cost
- More Flexibility
- Ability of Reading from Longer Range
- Lifelong capability.

a. Discuss the problem domain with historical background.

In Pakistan there is no such system is being implemented yet, but research on this idea has been done. This is a new innovative idea that brings the relief to the people by saving time. we are using RFID reader with Arduino and every product have RFID card number used for product identification which is connected to the Centralized database which will calculate the bill of the customer bill with Cart ID give to the trolley when they reached the counter. The system performance is increased and speed but the weaknesses is about the hardware to get damaged and not work. The bar-code scanning system for billing which is most time consuming. So we are presenting the solution of billing system, replacing it with automatic billing by scanning the product in trolley since every product has its own identity of RFID card number, RFID tags have Unique products ID that will help to increase performance.

b. All similar solutions and compare with your solution.

Literature Review:

- Smart Shopping Trolley Using RFID (International Engineering Research Journal (IERJ) Volume 2 Issue 3 Page 1418-1421, 2016, ISSN 2395-1621).
- SMART SHOPPING TROLLEY USING RFID (International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018, 3783-3786).
- A Review on Automatic Billing Trolley (International Journal of Research in Advent Technology (IJRAT) (E-ISSN: 2321-9637) Special Issue National Conference "CONVERGENCE 2017", 09th April 2017).
- RFID Based Smart Trolley for Automatic Billing System (Volume 7 Issue No.6 , ISSN XXXX XXXX © 2017 IJESC).
- Design and Implementation of a Smart Shopping Cart by RFID Technology (by Nemalidinne Sai Megana A thesis submitted in partial fulfillment of the requirements for the degree of Master of Engineering in Microelectronics and Embedded Systems).
- RFID BASED SMART TROLLEY FOR SUPERMARKET AUTOMATION (International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 04 Issue: 07 | July 2017 www.irjet.net p-ISSN: 2395-0072).
- A SMART TROLLEY WITH RFID IMPLEMENTATION: A SURVEY AMONG CUSTOMERS (VOL. 12, NO. 4, FEBRUARY 2017 ISSN 1819-6608, ARPN Journal of Engineering and Applied Sciences ©2006-2017 Asian Research Publishing Network (ARPN). All rights reserved. www.arpnjournals.com).
- LIFI BASED AUTOMATED SMART TROLLEY USING RFID (International Journal of Scientific & Engineering Research, Volume 7, Issue 3, March-2016, 1026 ISSN 2229-5518 IJSER © 2016 http://www.ijser.org).
- Developing a Multitasking Shopping Trolley Based on RFID Technology (International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-6, January 2014).
- Smart Shopping Trolley using RFID (Volume 8 Issue No.3, ISSN XXXX XXXX © 2018 IJESC).
- SMART SHOPPING TROLLEY FOR SUPERMARKETS USING RECHARGEABLE SMART CARD (International Journal of Scientific & Engineering Research Volume 8, Issue 7, July-2017 ISSN 2229-5518)
- SMART SHOPPING TROLLEY USING RFID (International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018, 3783-3786 ISSN: 1314-3395 (on-line version) URL: http://www.ijpam.eu).

c. Summary of 4-5 research papers related to the technology.

Literature Review:

(1) Smart Shopping System Using RFID:

International Engineering Research Journal (IERJ) Volume 2 Issue 3 Page 1418
1421, 2016, ISSN 2395-1621.

Result and Analysis:

Trolley utility will be for commercial use first of its kind. 2. With the help of appropriate sensors like RFID Tags, this device records the data of the various products. 3. This recorded data provides the shop owner through the machine with detailed analysis of the customer's shopping and their preferences; it is possible to obtain the same.

FUTURE SCOPE:

The proposed Smart Shopping Trolley System aims to assist in-person shopping which will minimize the considerable amount of time spent in shopping as well as the time needed to easily locate the desired product. The customer simply has to type the product name he / she wants to search on the Android device and the cart will automatically guide him / her to the locations of the products.

(2) SMART SHOPPING TROLLEY USING Smart Phone and Arduino:

Reference: Bedi HS, Goyal N, Kumar S, Gupta A (2017) Smart Trolley using Smart Phone and Arduino. J Electr Electron Syst 6: 223. doi: 10.4172/23320796.1000223.

CONCLUSION:

In Smart Trolley System, the customers now don't need to wait in the queue and wait for their turn to check the product products. Especially during the weekends or festivals season, waiting in the queue is not time-waste. The customer only has to pay at the billing counter and only those customers who have a membership card can use the smart trolley where RFID is used.

Future Scope:

We will refresh the product element information in the trolley memory unit from time to time. We take the help of the Internet of Things and some software which will regularly update all details. We will also produce trolleys with the aid of optical

sensors, motors and motor drivers in such a way that they will obey the customer who purchases products and keep them safe.

(3) RFID Based Smart Trolley for Automatic Billing System:

International Research Journal of Engineering and Technology (IRJET)e-ISSN: 2395-0056 Volume: 04 Issue: 07 | July -2017, www.irjet.net, p-ISSN: 2395-0072.

CONCLUSION:

Shopping will benefit from the proposed system for use in supermarkets, as it would make shopping easier. Now, in order to pay the bill, the customer does not need to stand in a queue. The system does automatic billing. This paper's idea was drawn from large queues at the shopping mall and the inconvenience it creates for the customer. This new billing system is quick as the details of the single product are recorded as it fell into the trolley. Working on this product it was noted that RFID technology and Zigbee in the near future are very extensive applications. Also, RFID is better and faster than reading barcode because the later is compact and reliable on line of sight, which is not the case with RFID technology. Zigbee is the wireless network which connects the costumer with the retailer and is very secure with long operating range. This smart shopping device will absolutely alter the way you shop. The most commonly-used RFID and Zigbee technology would definitely find some commercial use. In addition, this smart trolley will be very useful as it will reduce the number of salesmen and billing counters as well as save time for both the customer and the shopper.

(4) Design and Implementation of a Smart Shopping Cart by RFID Technology:

Smart shopping trolley application creates an automated central billing system in malls by using the ZigBee.

RESULT:

The proposal ultimately resulted in an effective result where RFID technology replaced barcode due to its drawback where barcode requires the line of sight and should be placed at its exact boundary while scanning, but RFID's only constraint is its distance coverage. RFID tags are longer lasting than barcodes that are harmful due to temperature, humidity, physical tear, etc. This ensures fast and accurate scanning process. Fig.3 shows system hardware connection and an LCD bill display. The Arduino uno digital input pins 9, 10, 11, 12, 13 are connected to the RST, SDA, MOSI, MOSO, SCK of the 3.3V VCC reader. The 5V LCD with I2C backpack includes SDA and SCL connected to Arduino's A4 and A5 respectively. Zigbee is used to send customer and counter product information. Zigbee S2C's TX-out and RX-in are connected for serial

communication to the microcontroller pin 2, 3. One of the S2C radios is configured as a coordinator for communication and another as a router. Zigbee is running 3.3V.

CONCLUSION:

Shopping will benefit from the proposed system for use in supermarkets, as it would make shopping easier. Now, in order to pay the bill, the customer does not need to stand in a queue. The system makes automatic invoicing. This paper's idea was drawn from large queues at the shopping mall and the inconvenience it creates for the customer. This new billing system is quick as the details of the single product are recorded as it fell into the trolley. It has been noted that RFID technology and Zigbee are very extensive applications in the near future. Also, RFID is better and faster than reading barcode because the later is compact and reliable on line of sight, which is not the case with RFID technology. Zigbee is the wireless network that connects the costumer to the retailer and is extremely secure with a long operating range. A smart shopping device will fully change the way shopping takes place. The most widely used RFID and Zigbee technologies will definitely find some use commercially. Moreover, this smart trolley will be very useful as it will reduce the number of salesmen and billing counters as well as save time for both consumers and shopper.

(5) LIFI BASED AUTOMATED SMART TROLLEY USING RFID:

International Journal of Scientific & Engineering Research, Volume 7, Issue 3, March-2016,1026 ISSN 2229-5518

FUTURE SCOPE:

The LIFI technology can be used in vehicle communication in future. The LCD can be fitted with a shopping market interface that allows consumers to receive accurate information on the items in different aisles. This improves user friendliness. During a shopping trip the smart trolley may interact with customers. For example, passing discount vouchers according to where they are in the supermarket. The trolley movement can be made automatically with the help of different sensors. There's no need to pull this big trolley in this direction.

CONCLUSION:

In the developed prototype model, the intended goal is successfully achieved. The software built is user-friendly and needs no specific training. It has the effective use of LIFI technology and can minimize the queues in the mall by the smart trolley. So, you can save the time of that client.

Chapter 3; Requirements & Methodology

PROJECT DESCRIPTION:

Our Idea is of making an automatic smart cart, assigning RFID tags to the products and RFID reader and with LCD with ESP8266(Wifi-Module) in the purchasing cart. the Customer can see the Cost of each product, number of products and total amount. the quantity of items will be printed in bill and if the customer has to change his mind then just press the Remove button and scan the products you want to remove; the product price and amount will be reduced from Total Amount. The Total Bill amount will be displayed on LCD in the cart. the customer sends the data by clicking on the Send button when completed shopping with the associated ID and from which the Customers bill will get printed, this will save customer time at the billing counter, this will save money and time. the smart cart will make shopping easier for the customers with improvising comfort for customers.

- What logic, constraints, platform and strategy you would use to implement requirements given below and Justify why you would use these strategies or platform or language?
- The logic behind all the project is to reduce long queues that cause headaches which could be possible using arduino with rfid reader with node mcu used in hardware in which product is scanned and stored in node mcu and then send the data to CBU which web form through xml /json and the bill generated at the billing counter must be verified and checked the cashier can check the product information at that time and for every customer there is s unique bill ID which we will use for customer verification, once the customer pays he received card from cashier to exit which will be authenticated card if unauthenticated card scan against rfid reader od security pass the alarm will be on this is generally the idea of the project we know that there are and have many limitation and risk is this project but until you don't catch the problem you could not found solutions and for that you have to dive in the sea and check and we have solved some problems and in future more people will extend it to the next level.
- We chose this platform or language because we are already diving in an ocean and to dive into another ocean of language would be most difficult for us and since everyone cannot have knowledge of everything you gather is in bits and pieces and which sums up knowledge and since we have knowledge that it could be built in lots of other language and since businessman find ease of doing business places, from that we have chosen from all difficulties that ease of using known language is very useful but I know it is helpful but have constraints.

Software requirements:

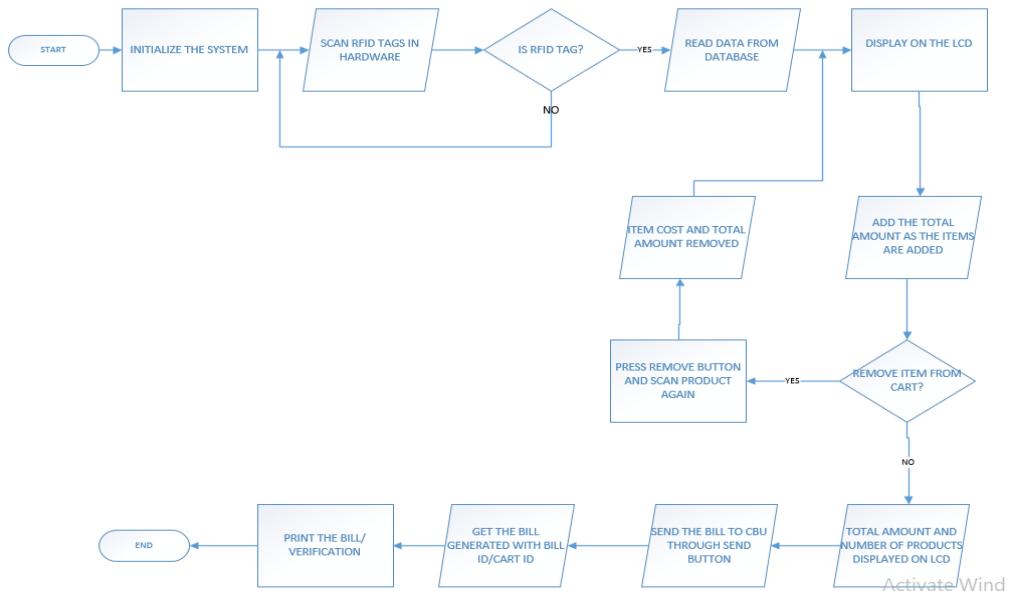
Programming Languages:

- i. Embedded C
- ii. PHP, CSS, JavaScript, HTML
- iii. MYSQL

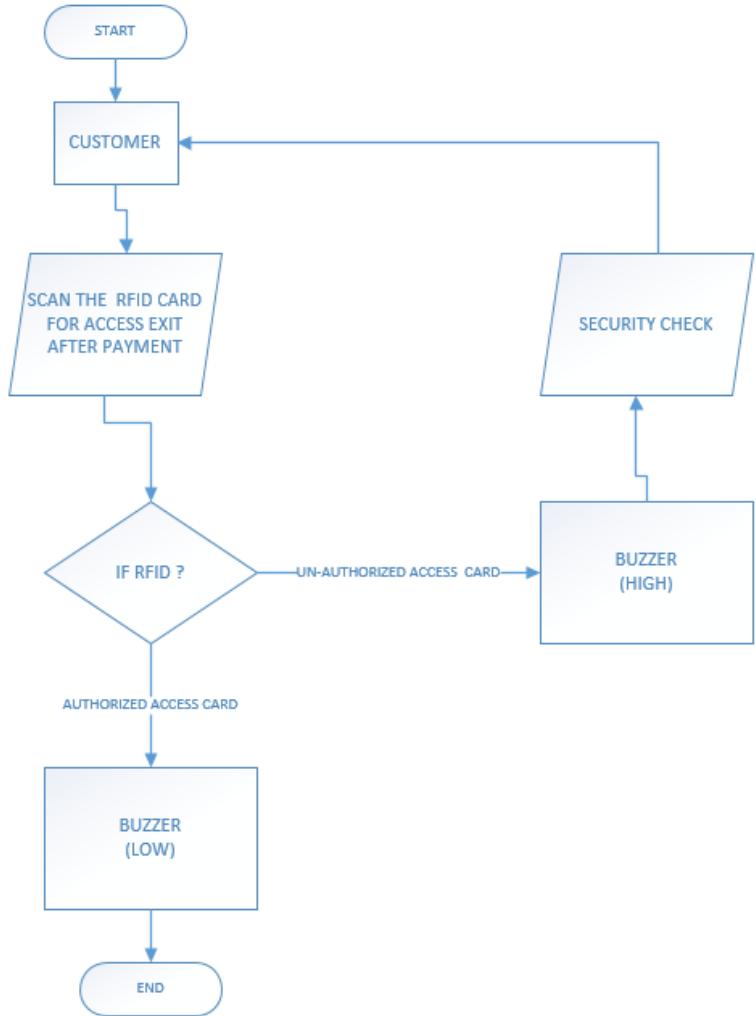
Platform:

- Sublime Text editor (software)
- MySQL (PhpMyAdmin)(database)

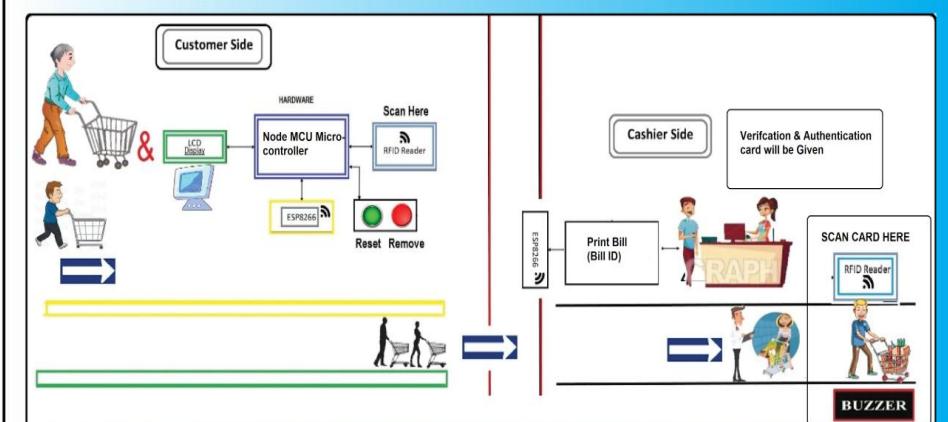
- **Project Flow Diagram :**



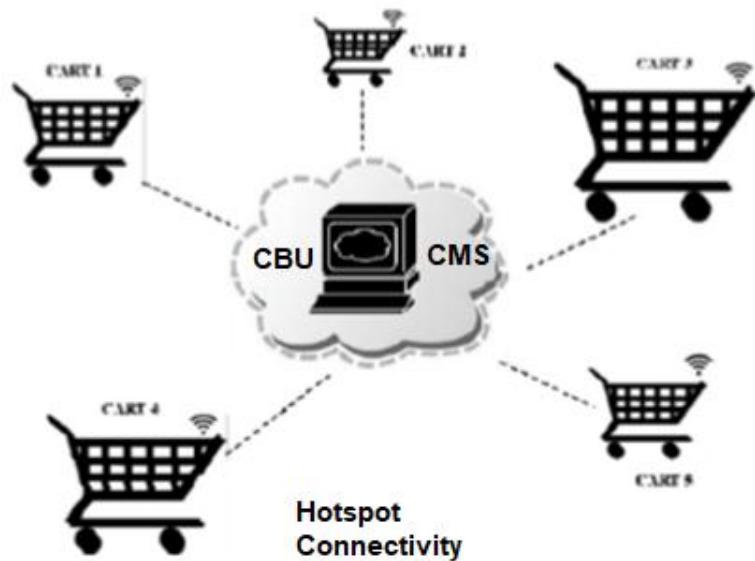
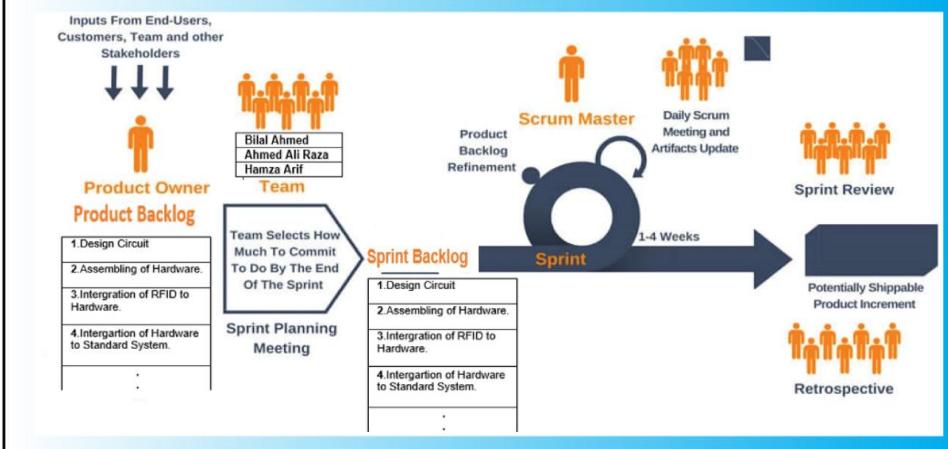
SECURITY ALARM DIAGRAM :



PROJECT ARCHITECTURE:



SOFTWARE DEVELOPMENT LIFE CYCLE:



FEATURES:

- Anti-theft Alarm.
- Interactive LCD.
- Automated Billing System.
- Individuals items description through LCD and RFID.
- Add /Remove Product in the Cart.
- Remove Button to Remove Item from Cart.
- Reset Button to Reset Cart.
- Item name and cost will be displayed on LCD.
- The total bill amount displayed in the Cart.
- The RFID label Can be read and rewritten.
- Time Diminishing.
- The solution is people who come up with a fixed budget.
- Amount will be displayed in the Cart.
- Labor intensive reduced from automated processes

- **Functional Requirements;**

System features:

- Every product in the store will have an RFID tag on it.
- There will be a Centralized Server System and Database which holds the information of the products.
- The product RFID tags should be scanned and added to the bill.
- Cart will have an RFID reader with Arduino and ESP-8266 communication.
- When a customer wants to remove any product from the trolley, then that product needs to be scanned again, displays of items bill on the LCD and centralized database will be updated.
- Display Product price and total Bill amount.
- After the payment of money, the Cart will be reset with the button given in hardware.
- The customer Cart ID will be verified and the product will be counted before paying the bill.
- The customer will get the printed form of bill to pay and then items will be packed and customer can leave.

- **Non-Functional requirements:**

System: Easy/Difficult for the Customers:

This implementation is used to assist a person while shopping and also to avoid standing in long queues and saving time

- Ease of use for customer: since we are using RFID reader and other components it will be new to the customer so it would be difficult for the customer to understand and use for the first time but once they have used it, they will know how to use and will make it easier when they constantly used it.
- The other factor that can affect project is the average time that what time from a customer entering a mart till the customer get bill on the cash counter without any difficulty or error in between.

- The customer can enter many products and remove any at time, since its centralized billing going on the Billing unit the workload of cash counter has been altered and can save 1 person pay and use it for increasing efficiency of product.
- First time customer used the project, some person has ease to understand and some have difficulty but for the first time they will be shown how to use it and complete the task of what project is made for.
- As the multiple user will access the system at the same time, therefore the software needs to be efficient and reliable.
- The accuracy of the project is calculated by the accuracy of calculating the bill of the customer.
- Security requirements should be fulfilled by only the data administrator has the rights to access the database with an ID and password.
- The automatic cart we are making which can add/remove product and update at the same time in database and calculating the total bill of all the customers separately that uses our project in the mart. Performance is that quality that shows you how responsiveness is the system and database updated at equal time and bill is generated in CBU, user interactions

with product decides the performance of product. Poor performance leads to negative user experience. The system security can fail when it gets overloaded.

System Availability:

- Availability of the product is defined that all the hardware components are available that are being used in system the RFID reader, RFID tags, Arduino kit, ESP8266 etc.
- Availability of the services that the RFID reader reads the tags and adds them in the cart which also being edited in database and bill will generate at the CBU and given to the customer, ensures that will work properly.
- Availability of the database and hardware app and CBU work together at the same time.

System Scalability:

It's being defined that the technology can grow with positive performance and since its increasing in the last few years, we can make hardware compatible and software that can take load and database does not fail. The automatic cart we are making which can add/remove product and update at the same time in database and calculating the total bill of all the customers separately that uses our project in the mart.

**Chapter 4;
Project Plan & Initial Design**

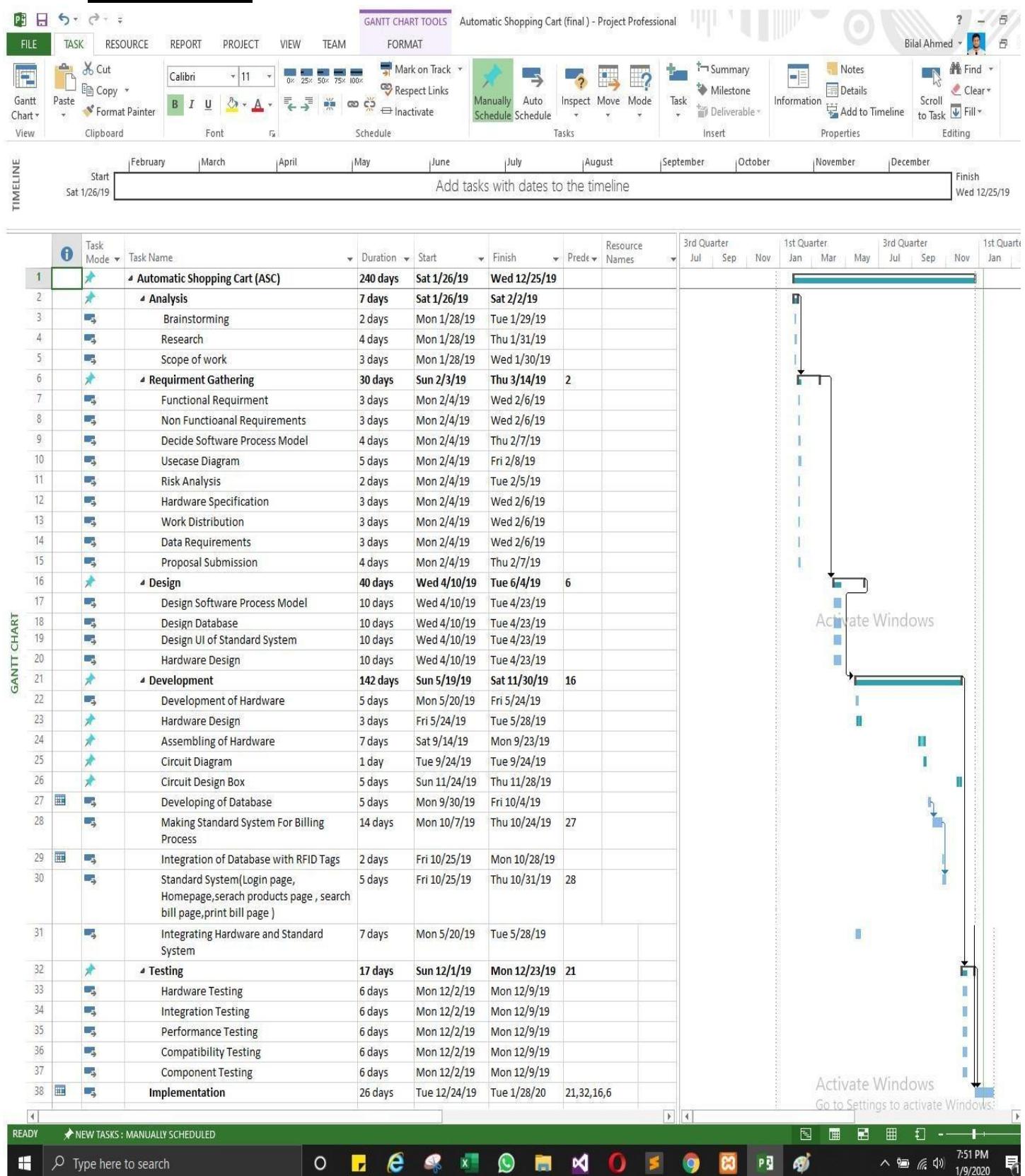
- Work Distribution:

Work Distribution	Member 1	Member2	Member3
Design Circuit of hardware (ASC)		✓	✓
Designing of Cart with Hardware	✓		✓
Assembling of Hardware		✓	
Developing (Coding) of Arduino		✓	
Integrating the RFID tag to Product ID	✓	✓	
Hardware Testing		✓	✓
Design Database	✓		✓
Database with table Entities and attributes	✓		✓
Integrating Database with RFID Tags	✓		
Connecting Database with Web Application with RFID Tags and Cart ID	✓	✓	
Integrating Billing System with Cart(Hardware)	✓	✓	
Testing of all modules (software & hardware)	✓	✓	✓
System Testing and Maintenance	✓	✓	✓
Implementation	✓	✓	✓

- Weekly Chart:**

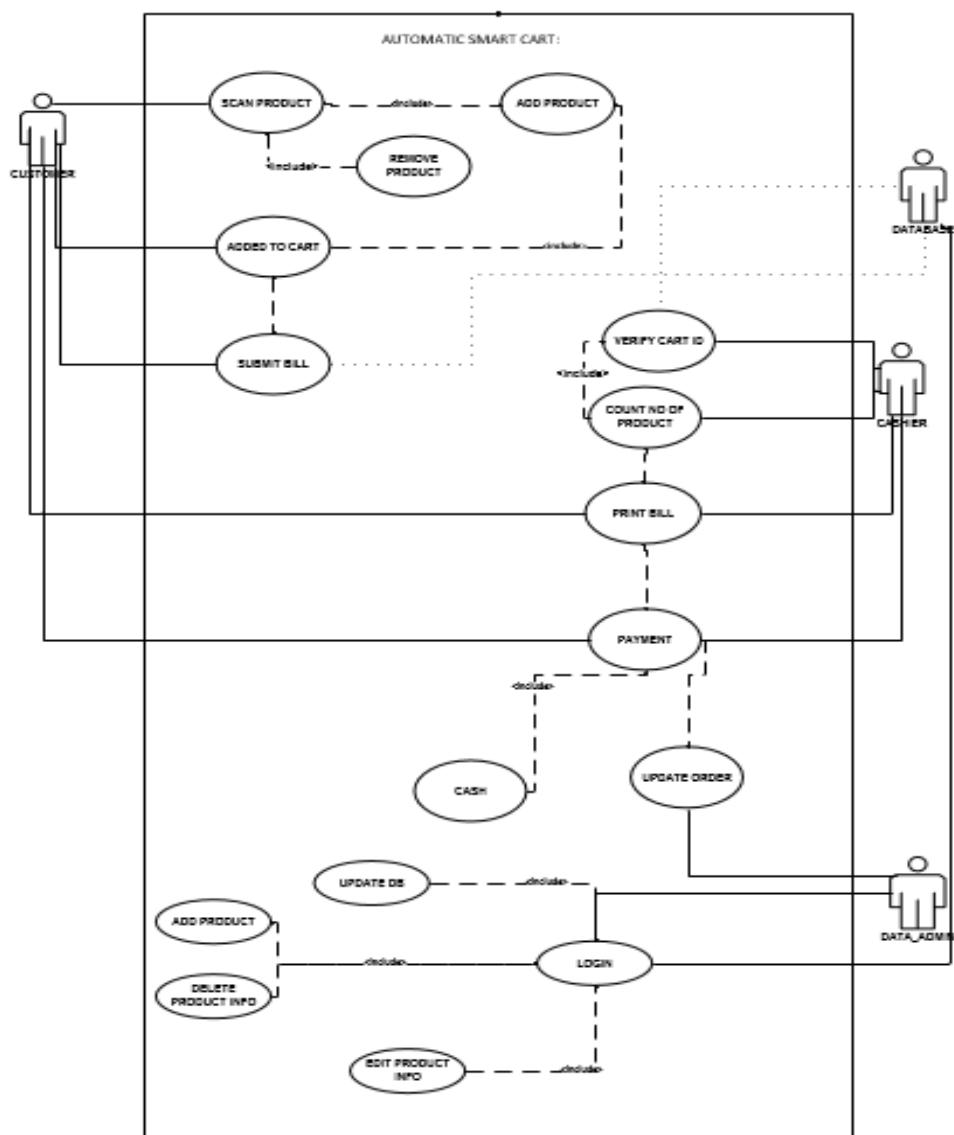
S. No.	Elapsed time since the start of the project	Milestone	Deliverable
1.	Week 01	Research of Suitable ideas	Potential Ideas
2.	Week 02	Brainstorming of Ideas	One specific idea for Project
3.	Week 03	Requirement gathering (Market survey for main Equipment)	Budget and Specifications of hardware
4.	Week 04	Requirement gathering	Software process model
5.	Week 05	Preparation of Use case Diagram	Use case Diagram
6.	Week 06 - 7	Verification of Proposal	Proposal Verified
7.	Week 08	Proposal Feedback	feedback Solution
8.	Week 09	Data requirement work distribution	ERD diagram work distribution
9.	Week 10 - 13	Poster presentation	Presenting the idea
10.	Week 14	Approval of proposal	Proposal accepted
11..	Week 15	Design Circuit of hardware (ASC)	Making Circuit Diagram
12.	Week 16 - 17	Assembling of Hardware	Purchasing of hardware component
13.	Week 18	Developing (Coding) of Arduino	Coding in Arduino to identify product
14.	Week 19	Integrating the RFID tag to Product ID	Assigning RFID tag to Product ID
15.	Week 20 -22	Hardware Testing	Testing Hardware
16.	Week 23	Design Database	Database Design with tables
17.	Week 24	DataBase with table Entities and attributes	DataBase Created in MySQL
18.	Week 25-26	Integrating Database with RFID Tags	DataBase Integrated with RFID Tags
19.	Week 27 – 30	Developing the Web Application	Web Application
20.	Week 27-30	Creating Web page	Web page
21.	Week 28-29	Creating UI for Web Application	UI Interface
22.	Week 27-32	Developing Code for Web Application in HTML,PHP	Working of Web Application
23.	Week 32-33	Tested Desktop Application	Testing and debugging Desktop Application
24.	Week 34-35	Connecting DataBase with Desktop Application with RFID Tags and Cart ID	Integrating Desktop Application with Database with RFID Tags.
25.	Week 36– 38	Integrating Desktop Application with Hardware	Integration the both module
26.	Week 39 – 40	Testing of all modules (software & hardware)	Testing all module
27.	Week 41-44	System Testing and Maintenance	Testing and Performance testing
28.	Week 45-49	Implementation	Optimized Product

● GANTT CHART:



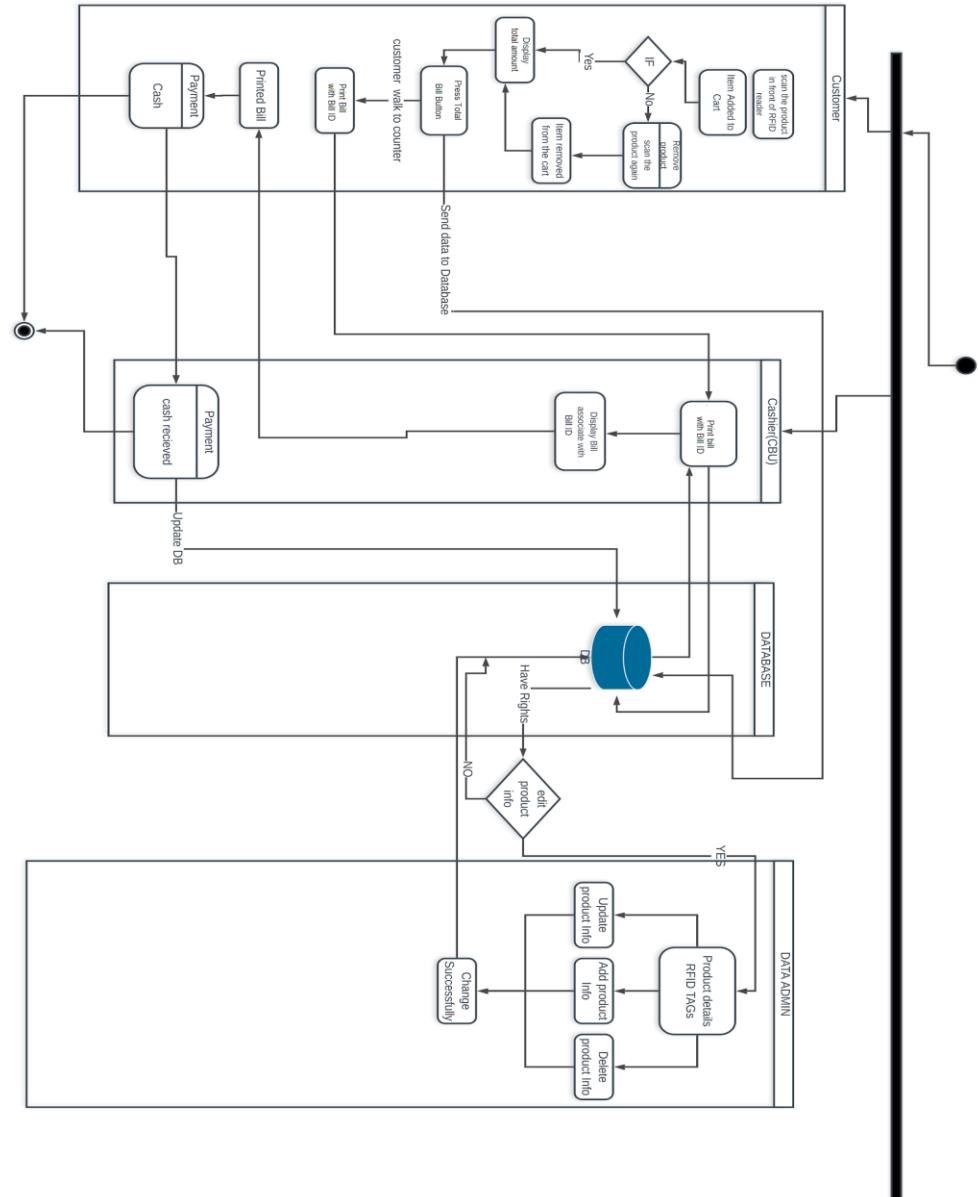
- Add all diagrams including use case, sequence, activity, and DFD (whatever are applicable). You can add it to your FYP1 report.

- **USE CASE DIAGRAM:**



Ricah

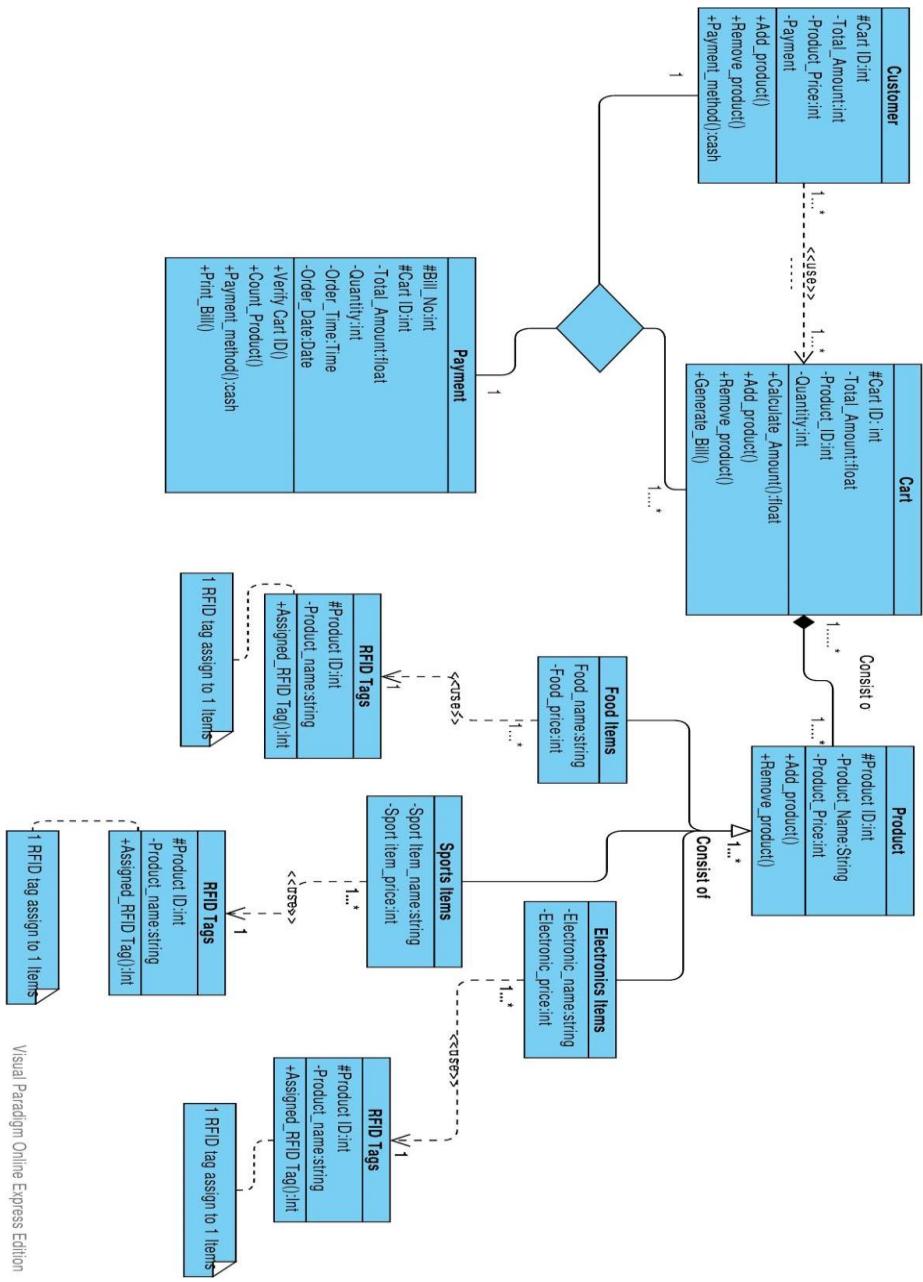
- **ACTIVITY DIAGRAM:**



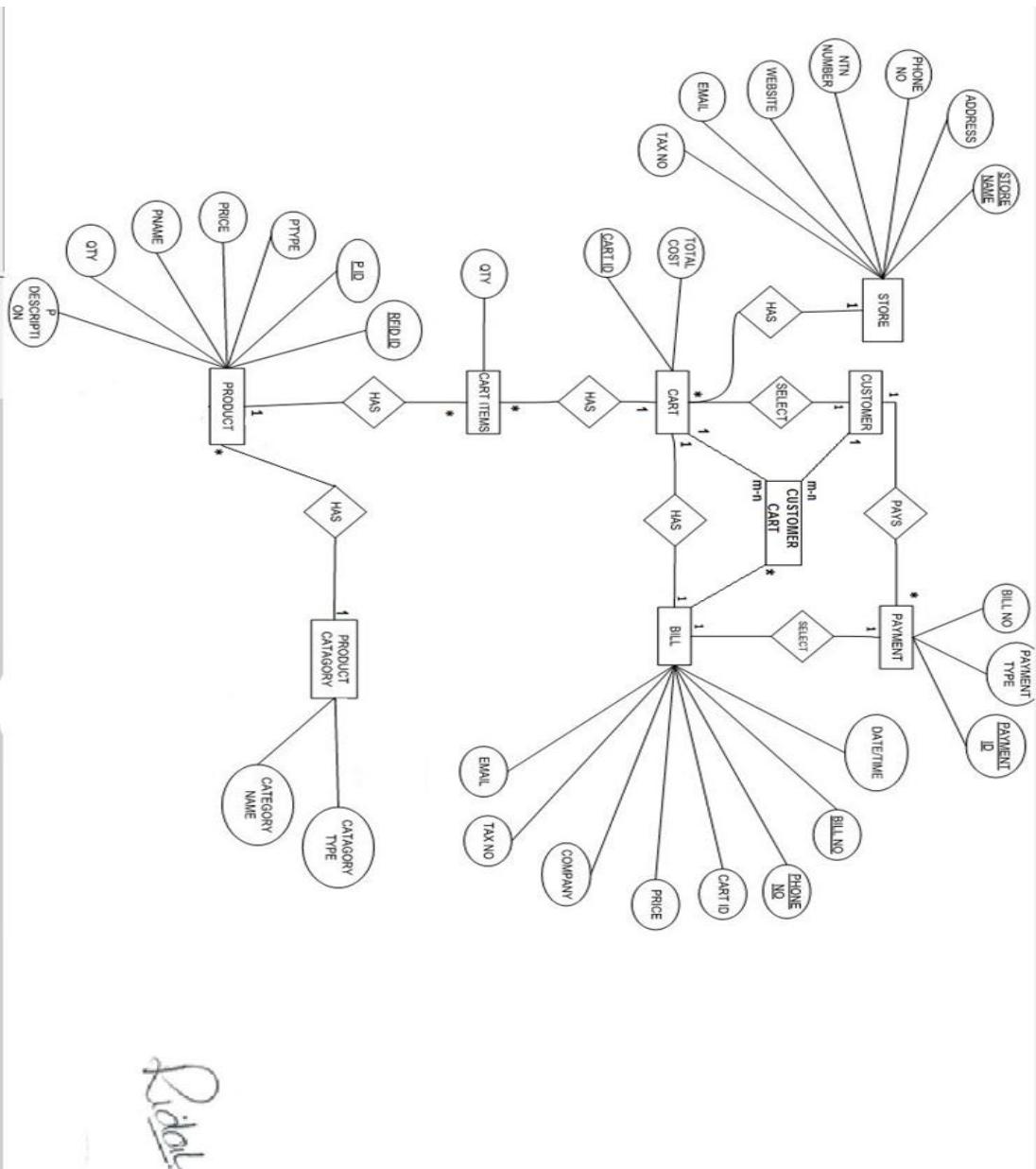
- Class Diagram:

Visual Paradigm Online Express Edition

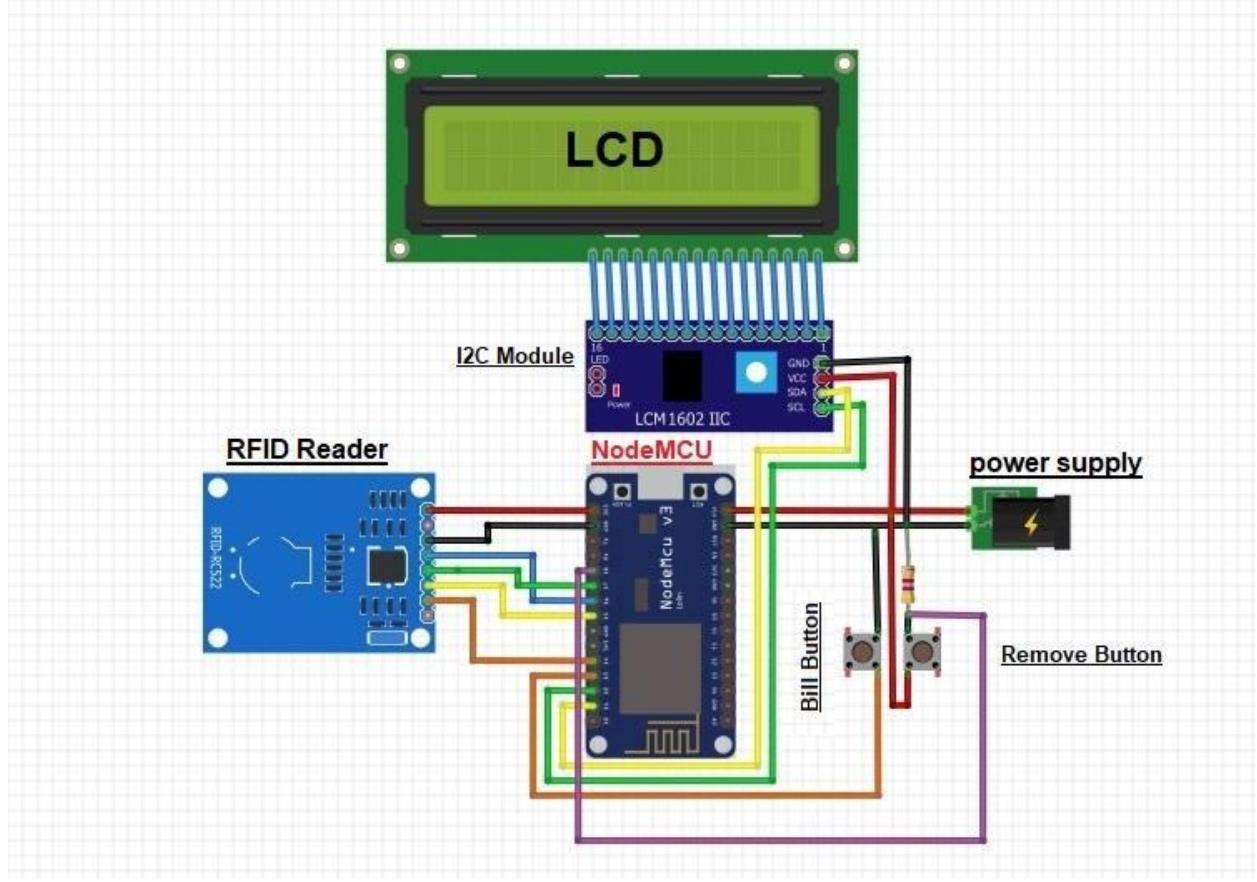
CLASS Diagram(ASC)_2019



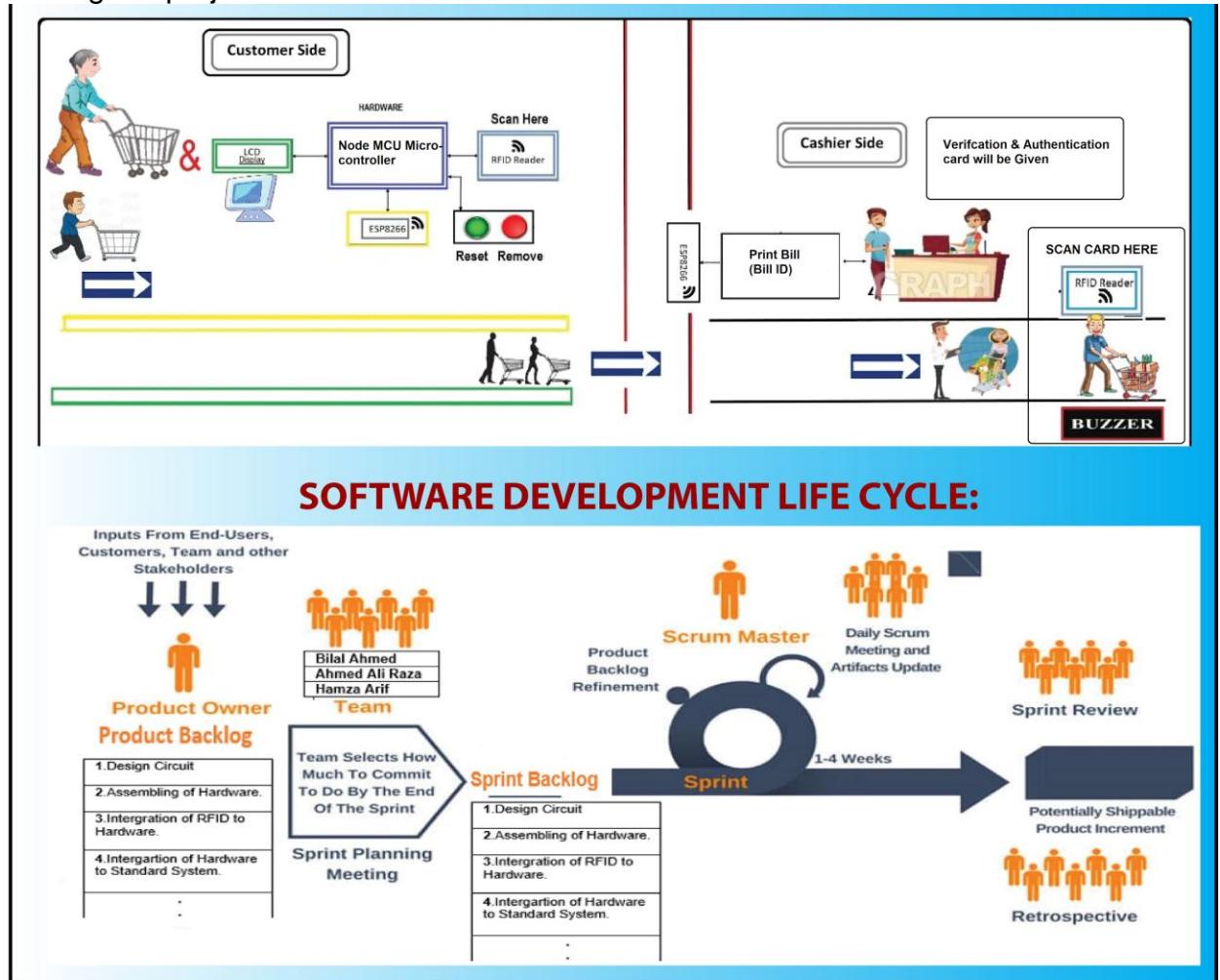
ERD Diagram:



CIRCUIT DIAGRAM:



- Add Prototype (blueprint) or initial design of your UI, which was made before starting the project



• Model Description:

The software model that we are using is the agile software development model in our project because since our project is hardware and software integrated project(Automatic smart Cart), since agile method uses incremental/iterative approach for designing a software, so we can break the product modelling into small modules or different increments which is easy to follow rather than making the whole project phase by phase and testing it after building the whole system, and in agile model every iteration has ended and sprint is tested and we can get feedback at that time. we can manage small modules to make a whole system which includes hardware, since we are clear about the requirements and what is the scope of our project.

- **SCOPE:**

A Shopping Cart that should be possible conceivable by essentially interfacing RFID marks to the things and Display with an LCD in the obtaining Cart. From this customer can add/remove product and can also get information identified with expense of everything which are inside Cart and moreover supreme expense of the thing about the item. This system will spare time and work required for of customers & labor. the system has anti theft alarm at the exit with RFID Reader with Buzzer.

Chapter 5; Project Design & Development

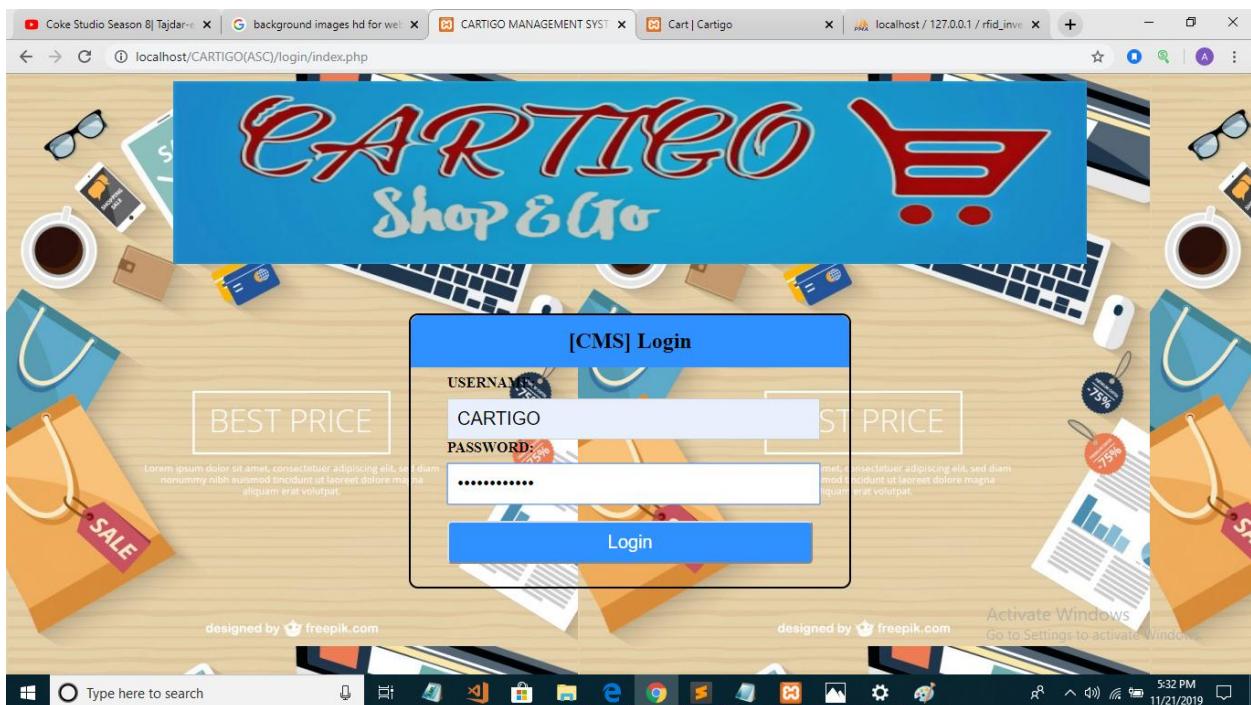
- **Design**
 - Front End Design: This includes color theme, logo, template, screenshots of the user interface of all screens and web pages, etc. with description.
- **COLOR THEME:**
(BLUE, WHITE, LAVENDER, BLACK)
- **LOGO: The LOGO of the Cartigo Management System(CMS)**



SCREENSHOTS OF USER INTERFACES:

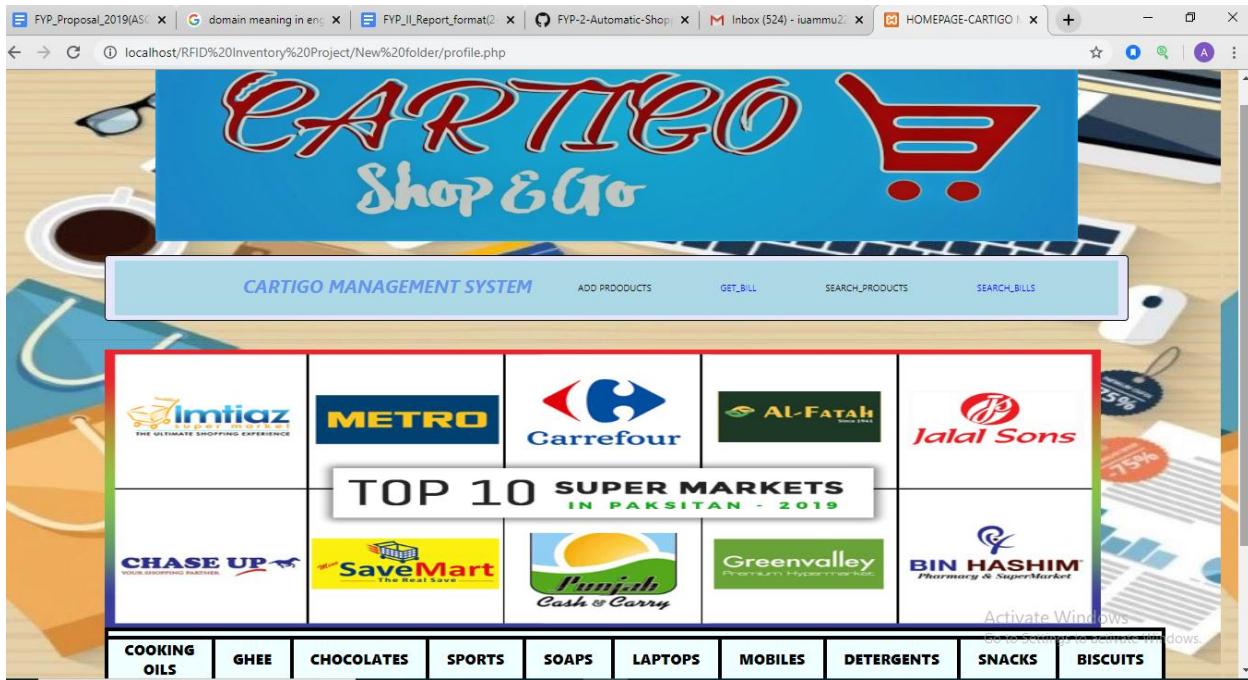
- **LOGIN PAGE:**

The Login page where you enter the username and password and the username and password are saved in the Database. When you enter the Username and password it will fetch it from database and check the validation if its correct it will redirect to Homepage (profile.php) otherwise you would be redirected to the login page.



- **Homepage:**

The Homepage will be displayed when you Login to the web form and provided with all the navigation through different web pages, navigation Bar.



- **ADD PRODUCTS:**

The products can be added through web form when you click add new products button and you enter RFID, product name and products price and click on add products the products created message will be displayed in the web form and products also created in the Database.



The screenshot shows two consecutive views of a web application interface. The top view displays a table of existing products with columns for RFID UID, Name, Price, and Actions (Edit, Delete). The bottom view shows a form for adding new products with fields for RFID UID, Product Name, and Product Price, along with an 'Add' button.

RFID UID	Name	Price	Actions
1052	BADMINTON RACKETS (2R/L)	600.00	Edit Delete
1171	BAKERI BISTIKS(LU)(10pcs)	200.00	Edit Delete

RFID UID:	<input type="text"/>
Product Name:	<input type="text"/>
Product Price:	<input type="text"/>
Add	

Screenshot of a web application interface titled "PRODUCTS(DATABASE)" showing a form for adding a new product.

The form fields are:

- RFID UID: 1202
- Product Name: ahmed
- Product Price: 2000000

An "Add" button is present at the bottom right of the form.

Below the form, a message says "Product was created." followed by the "CARTIGO Shop & Go" logo.

At the bottom, a table displays the added product:

RFID UID	Name	Price	Actions
1202	ahmed	2000000.00	Edit Delete

- **DELETE PRODUCTS:**

The products can be deleted by just clicking the delete button which will delete product all information through ID and the products will no longer be available in the Database.



Screenshot of the CARTIGO Management System showing the product database and a product listing page.

phpMyAdmin Interface:

- Server: 127.0.0.1 > Database: rfid_inventory > Table: products
- Showing rows 0 - 233 (234 total, Query took 0.0016 seconds.) [id: 1... - 280...]
- SQL Query: SELECT * FROM `products` ORDER BY `id` ASC
- Filter rows: vivo
- Sort by key: PRIMARY (ASC)
- Options: id, rfid, name, price, category
- Action buttons: Check all, With selected: Edit, Copy, Delete, Export
- Query results operations: Print, Copy to clipboard, Export, Display chart, Create view
- Bookmark this SQL query: Label: (empty), Let every user access this bookmark

Product Listing Page:

ID	Product Name	Price	Action
1030	TWIX	60.00	Edit Delete
1186	VIVO V7 Pro	23000.00	Edit Delete
1109	WAVYS(BARBIQUE)(1pcs)	50.00	Edit Delete
1105	WAVYS(BARBIQUE)(1pcs)	20.00	Edit Delete
1107	WAVYS(BARBIQUE)(1pcs)	30.00	Edit Delete
1106	WAVYS(BARBIQUE)(24pcs)	480.00	Edit Delete
1108	WAVYS(BARBIQUE)(24pcs)	720.00	Edit Delete
1110	WAVYS(BARBIQUE)(24pcs)	1200.00	Edit Delete
1115	WAVYS(SALT & PEPPER)(1pcs)	50.00	Edit Delete
1111	WAVYS(SALT & PEPPER)(1pcs)	20.00	Edit Delete

Windows Taskbar:

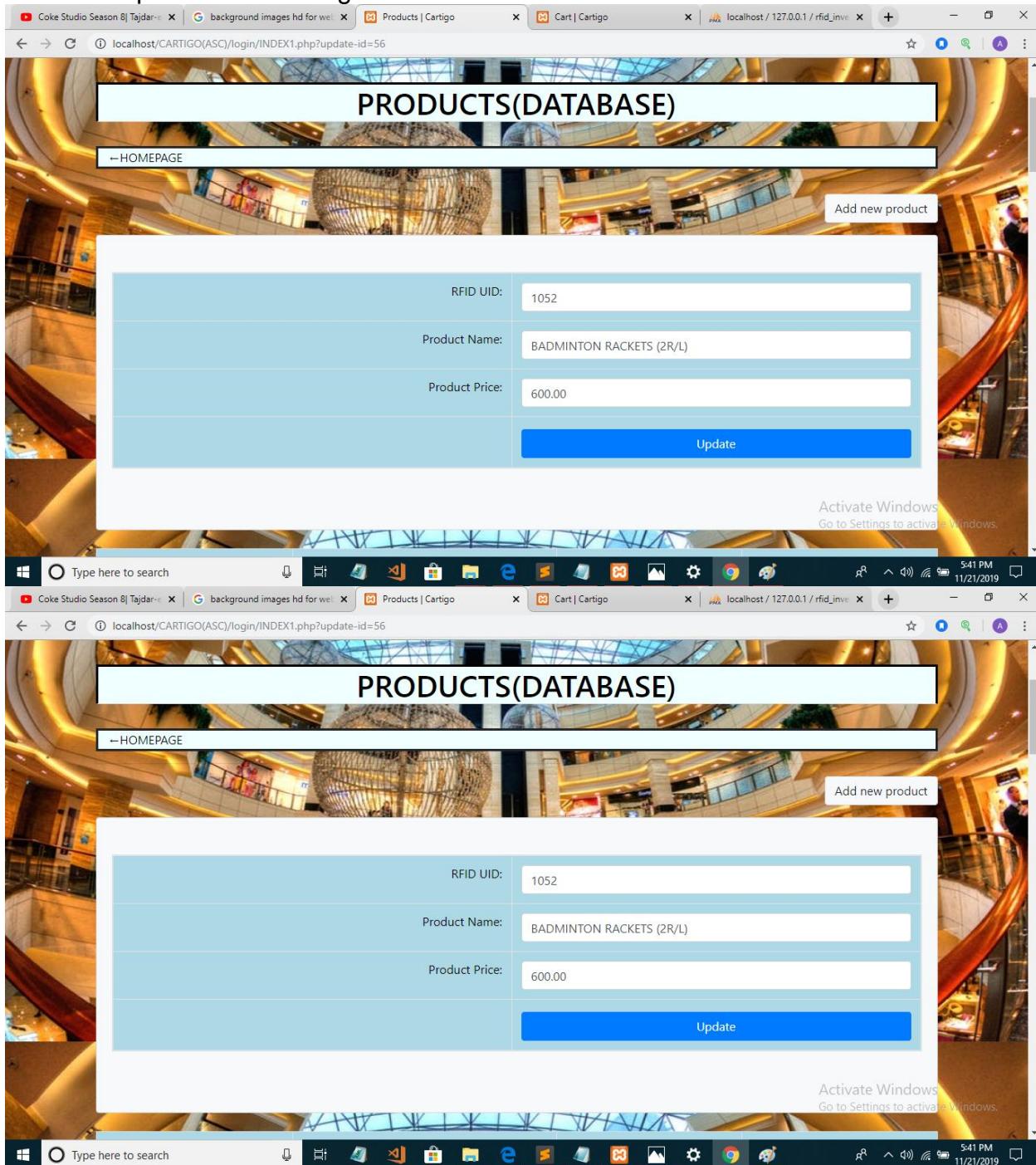
- Type here to search
- Coke Studio Season 8 | Tajdar-e-Hind
- background images hd for web
- Products | Cartigo
- Cart | Cartigo
- localhost / 127.0.0.1 / rfid_inve

Bottom Status Bar:

- 5:40 PM 11/21/2019
- Activate Windows
- Go to Settings to activate Windows.
- 5:38 PM 11/21/2019

- **UPDATE PRODUCTS:**

The product information can be updated through web form which is connected with database updates also changes in the database when edited in the web form.



A screenshot of a web browser window titled "Products | Cartigo". The main content area displays a form titled "PRODUCTS(DATABASE)". The form contains three input fields: "RFID UID:" with value "1052", "Product Name:" with value "BADMINTON RACKETS (2R/L)", and "Product Price:" with value "600.00". Below the form is a blue "Update" button. In the top right corner of the browser window, there is a watermark that says "Activate Windows Go to Settings to activate Windows." The browser tabs show other pages like "Coke Studio Season 8 | Tajdar-e-Hind", "background images hd for we...", and "Cart | Cartigo". The system tray at the bottom shows the date and time as 11/21/2019 5:41 PM.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled 'Products | Cartigo'. A message at the top of the page says 'Product was updated.' Below this is a large banner for 'CARTIGO Shop & Go' featuring a red shopping cart icon. A main heading 'PRODUCTS(DATABASE)' is centered above a search bar. To the left of the search bar is a link to the 'HOMEPAGE'. On the right, there is a button labeled 'Add new product'. The main content area displays a form for updating a product record:

RFID UID:	<input type="text" value="1052"/>
Product Name:	<input type="text" value="BADMINTON RACKETS (2R/L)"/>
Product Price:	<input type="text" value="6.00"/>
Update	

Below this form is a table listing products:

RFID UID	Name	Price	Actions
1052	BADMINTON RACKETS (2R/L)	6.00	Edit Delete
1171	BAKERI BISTIKS(LU)(10pcs)	200.00	Edit Delete
252	BAKERI BISTIKS(LU)(12)	200.00	Edit Delete

RFID UID: 1052

Product Name: BADMINTON RACKETS (2R/L)

Product Price: 600.00

Update

RFID UID	Name	Price	Actions
1052	BADMINTON RACKETS (2R/L)	6.00	Edit Delete
1171	BAKERI BISTIKS(LU)(10pcs)	200.00	Edit Delete
252	BAKERI BISTIKS(LU)(12)	200.00	Edit Delete

Add new product

RFID UID: 1052

Product Name: BADMINTON RACKETS (2R/L)

Product Price: 600.00

Update

RFID UID	Name	Price	Actions
1052	BADMINTON RACKETS (2R/L)	600.00	Edit Delete
1171	BAKERI BISTIKS(LU)(10pcs)	200.00	Edit Delete

- SEARCH PRODUCTS:**

The products could be searched by product ID which is different and you could find out the ID by navigating through the table in web page which has all information about products and their ID and RFID.



The screenshot shows two instances of a web browser displaying the 'CARTIGO Management System' interface. The top instance shows a search form for 'PRODUCT_ID' with a placeholder 'ENTER ID'. Below it is a 'PRODUCTS' section with categories: COOKING OILS(20), GHEE(17), CHOCOLATES(32), SPORTS(15), SOAPS(3), LAPTOPS(2), MOBILES(1), DETERGENTS(1), SNACKS(105), and BISCUITS(32). The bottom instance shows the results of a search for '12'. A table displays the following data:

id	name	rfid	price	category
12	Meezan Banaspati(GHEE)(1KG)	1007	200.00	GHEE

The screenshot shows a web browser window with the title "CARTIGO Management System". The main content area displays a "PRODUCTS" section with a table showing a single item:

id	name	rfid	price	category
280	CAFE(MAYFAIR)(6pcs)	1185	120.00	BISCUITS

This screenshot is identical to the one above, showing the same product details for item ID 280. The interface and data remain consistent.

- Navigation to products name and ID with Price:
- From the search products page you can navigate to pdf file of all products with their name, ID and price.

The screenshot shows the homepage of the CARTIGO Management System. At the top, there is a large logo with the text "CARTIGO Shop & Go" and a red shopping cart icon. Below the logo, a navigation bar contains links for "HOMEPAGE", "GET_BILL", "ADD PRODUCTS", and "SEARCH_BILLS". A search bar labeled "PRODUCT ID:" is present, with the placeholder text "ENTER PRODUCTS ID". Below the search bar, a section titled "PRODUCTS" lists categories: COOKING_OILS(), GHEE(), CHOCOLATES(), SPORTS(), SOAPS(), SNACKS(), and BISCUITS().

cooking_oils.pdf

ID	Name	Price	Category	
1	FATCOOP	10 Cooking Oil	100	COOKING_OILS
26	FAAK90103	1001 DM (5 KG)	1000	COOKING_OILS
27	FAAK90104	Shiro Cooking Oil(5kg)	1000	COOKING_OILS
201	129	Hilma Cooking (5l)kg	200	COOKING_OILS
202	130	Hilma Cooking (5l)kg	1000	COOKING_OILS
203	131	Hilma Cooking (5l)kg	200	COOKING_OILS
204	132	Toko Cooking (5l)kg	1000	COOKING_OILS
205	133	Kastrol Cooking (5l)kg	1000	COOKING_OILS
206	134	Kastrol Cooking (5l)kg	1000	COOKING_OILS
207	135	Kastrol Cooking (5l)kg	200	COOKING_OILS
208	136	Kastrol Cooking (5l)kg	1000	COOKING_OILS
209	137	Dafra Cooking (5l)kg	1000	COOKING_OILS
210	138	Dafra Cooking (5l)kg	200	COOKING_OILS
211	139	Miracle Cooking (5l)kg	1000	COOKING_OILS
212	140	Miracle Cooking (5l)kg	200	COOKING_OILS
213	141	Soya Supreme Cooking (5l)kg	1000	COOKING_OILS
214	142	Soya Supreme Cooking (5l)kg	200	COOKING_OILS
215	143	Miracle's Cooking (5l)kg	200	COOKING_OILS
216	144	Soya Supreme Cooking (5l)kg	1000	COOKING_OILS
217	145	Soya Supreme Cooking (5l)kg	200	COOKING_OILS
218	146	Soya Supreme Cooking (5l)kg	1000	COOKING_OILS
219	147	Soya Supreme Cooking (5l)kg	200	COOKING_OILS

chocolates.pdf

ID	Name	Price	Category	
24	150091510	Dairy Milk	100	
95	23	Cadbury Dairy Milk Chocolate (Box)	10	
96	24	Cadbury Dairy Milk Chocolate (Box)	100	
97	25	Cadbury Dairy Milk Chocolate (Box)	25	
98	26	Cadbury Dairy Milk Chocolate (Box)	50	
99	27	Cadbury Dairy Milk Chocolate (Box)	65	
100	28	Cadbury Dairy Milk Chocolate (Box)	100	
101	29	Cadbury Dairy Milk Chocolate (Box)	35	
102	30	Cadbury Dairy Milk Chocolate (Box)	95	
103	31	Cadbury Dairy Milk Chocolate (Box)	135	
104	32	Cadbury Dairy Milk Chocolate (Box)	25	
105	33	Cadbury Dairy Milk Chocolate (Box)	450	
106	34	KitKat	60	
107	35	KitKat	10	
108	36	Twix	60	
109	37	Reese	60	
110	38	GALAXY (BARK) Chocolate	725	
111	39	BOUNTY Chocolate	60	

SNACKS.pdf

ID	Name	Price	Category	
55	150091510	Lays	10	SNACKS
96	150091510	Chips	10	SNACKS
27	200091510	Doritos	400	SNACKS
138	06	KURLAZZ(MASALA) (Box)	10	SNACKS
139	07	KURLAZZ(MASALA) (Box)	200	SNACKS
140	08	KURLAZZ(MASALA) (Box)	30	SNACKS
141	09	KURLAZZ(MASALA) (Box)	70	SNACKS
142	10	KURLAZZ(MASALA) (Box)	10	SNACKS
143	11	KURLAZZ(MASALA) (Box)	100	SNACKS
144	12	KURLAZZ(MASALA) (Box)	20	SNACKS
145	13	KURLAZZ(MASALA) (Box)	240	SNACKS
146	14	KURLAZZ(MASALA) (Box)	30	SNACKS
147	15	KURLAZZ(MASALA) (Box)	70	SNACKS
148	16	KURLAZZ(MASALA) (Box)	50	SNACKS
149	17	KURLAZZ(MASALA) (Box)	100	SNACKS
150	18	LAYS(LT) (Box)	10	SNACKS
221	79	LAYS(LT) (Box)	20	SNACKS
312	80	LAYS(LT) (Box)	20	SNACKS
93	81	LAYS(LT) (Box)	400	SNACKS
154	82	LAYS(LT) (Box)	10	SNACKS
155	83	LAYS(LT) (Box)	10	SNACKS
156	84	LAYS(LT) (Box)	50	SNACKS
227	85	LAYS(LT) (Box)	100	SNACKS
158	86	LAYS(LT) (Box)	10	SNACKS

- **SEARCH BILL:**

Search the bill through bill ID and you will get the amount and through which you can find out bill saved in the pdf file. you can also give Cart ID to hardware and for multiple cashiers you need multiples hardware which could solve this issue.

cartID	id	amount
1122	2	75.25

cartID	id	amount
1020	3	451.50

localhost/phpmyadmin/sql.php?server=1&db=rfid_inventory&table=bills&pos=0

Showing rows 0 - 2 (3 total, Query took 0.0008 seconds.)

SELECT * FROM `bills`

	cartID	id	amount			
<input type="checkbox"/>	Edit	Copy	Delete	786	1	75.25
<input type="checkbox"/>	Edit	Copy	Delete	1122	2	75.25
<input type="checkbox"/>	Edit	Copy	Delete	1020	3	451.50

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options Edit Copy Delete Check all With selected: Edit Copy Delete Export

Query results operations Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query Label: Let every user access this bookmark Activate Windows Go to Settings to activate Windows. Bookmark this SQL query

Console

Type here to search

CARTIGO Management System(Shop & Go) HOMEPAGE GET_BILL SEARCH_PRODUCTS SEARCH_BILLS

BEST PRICE

BILL_ID: ENTER ID

cartID	id	amount
786	1	75.25

SEARCH BY

Logout

designed by freepik.com

Activate Windows Go to Settings to activate Windows.

- **PRINT BILL:** The bill printed in bill format with all the products, price, quantity and total amount with unique bill ID when you press the total bill button on the hardware side the bill generated in the web form application.

11/25/2019

Cart | Cartigo

[Print Bill](#)



Address: Block D-13 gulshan e iqbal Karachi
Phone: 1232435456
Bill ID: 51

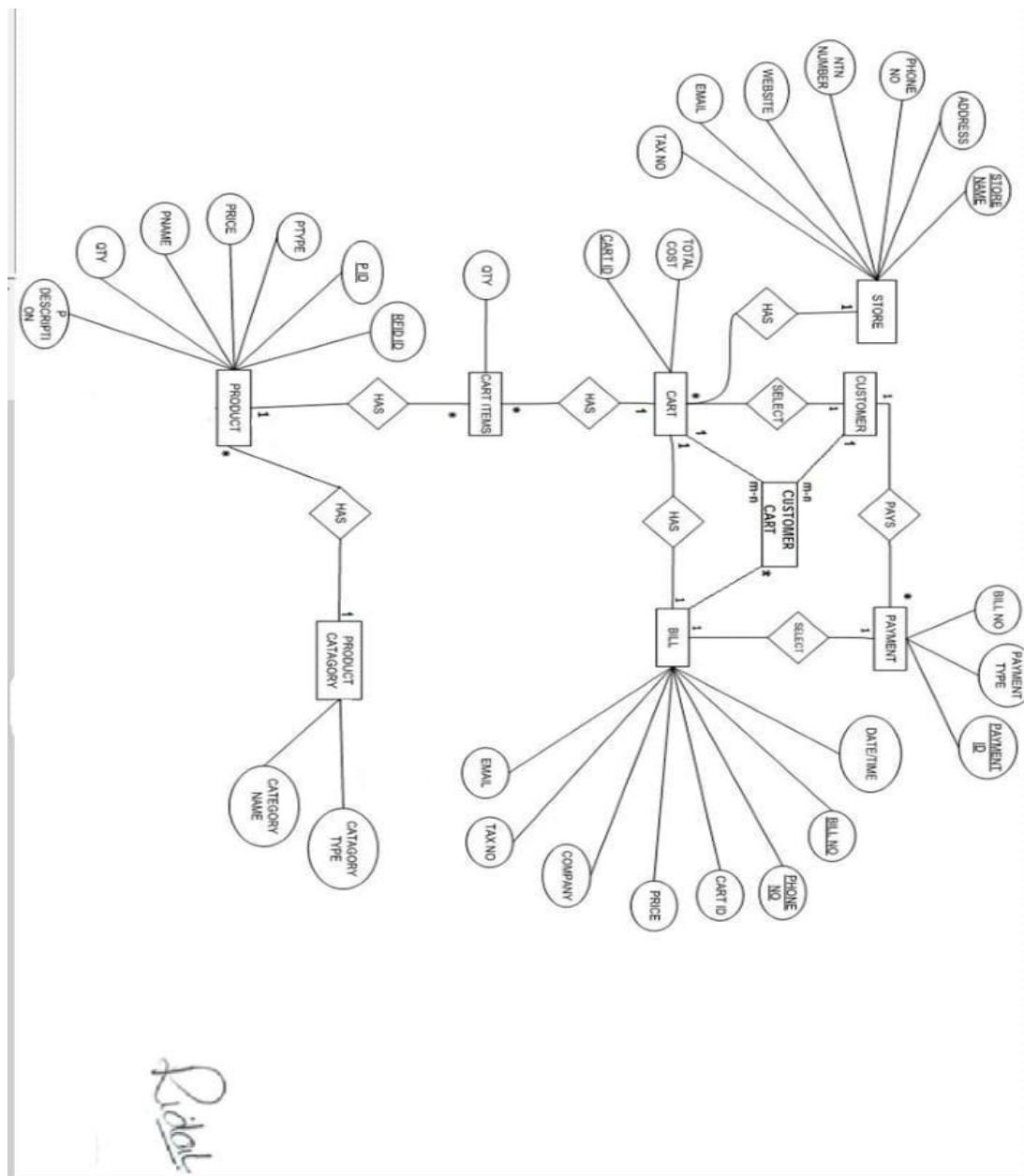
S.No.	Product Name	Quantity*	Product Price	RFID Tag No.	Amount
1	Milk pack	1	40.00	59870E10	40
2	LUX Soap	1	70.00	8916DF0F	70
3	Custard (1/2 kg)	1	120.00	8994C60F	120
4	Head Shoulders	1	95.00	39521E10	95
5	HP Laptop (Pro B	2	86000.00	E97CF70F	172000
6	MOTO E4(ROM 16GB	1	10000.00	199AB80F	10000
7	Dairy Milk	1	100.00	59070310	100
8	Lays	1	10.00	192AFB0F	10
9	Cooking Oil	1	180.00	F9E9C00F	180
10	Ketchup	1	70.00	392FC70F	70
11	Kisan Nimko	1	40.00	B97EC90F	40
12	Cookies	1	60.00	E9D6F50F	60
13	Surf excel	1	90.00	C9570D10	90
14	Cheetos	1	20.00	59D30F10	20
15	Doritos	1	450.00	2988C80F	450
16	Capri Soap	1	40.00	09D4DC0F	40
17	Dettol Soap	1	50.00	F978E20F	50
18	Bisconni Cocomo	1	10.00	0981E40F	10
19	HP Laptop (Probo	1	19000.00	39FBDE0F	19000

Total Amount: Rs 202445

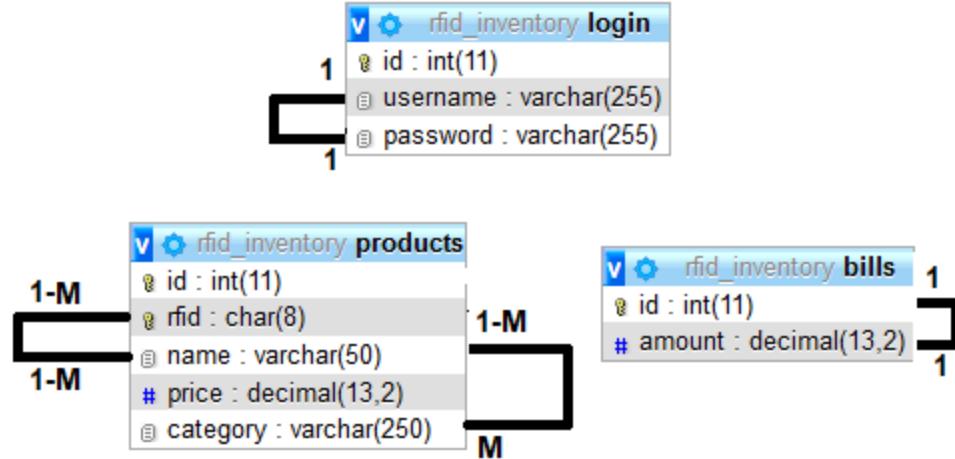
Date & Time: 25-11-2019 (01:41:20 pm)

***** Thank you for Shopping *****

- Backend Design: this includes physical ERD of the database designed directly using DBMS after implementing the conceptual design. It must have column types and sizes.



Ridol



- **Development**

- Discuss different stages of development, add code or any important logic you want to discuss. Your algorithm, running your algorithm and generating results screenshots, screenshots of your project structure, classes, and resources, important files screenshot, discuss code you used to design layouts and implement logic.
- Mention libraries you used also the version of language and platform used to develop the project.

- **Languages:**

- C-Language
- PHP,
- Javascript,
- CSS,
- Bootstrap

- **LIBRARIES:**

1. `<ESP8266WiFi.h>`
2. `<ESP8266HTTPClient.h>`
3. `<ArduinoJson.h>`
4. `<SPI.h>`
5. `<MFRC522.h>`
6. `<LiquidCrystal_I2C.h>`
7. `<Wire.h>`

- **PLATFORM:**
 - Sublime Text Editors
 - MySQL/phpMyadmin
 - Xampp server
 - Arduino IDE

- **Add important database queries, update database logics.**
 - add/ delete/ update/save total/ Get Bill ID query:

```
<?php
class Product{
    // database connection and table name
    private $conn;
    private $table_name = "products";
    // object properties
    public $rfid;
    public $name;
    public $price;
    public function __construct($db){
        $this->conn = $db;
    } // create product
    public function create(){
        //write query
        $query = "INSERT INTO
                  " . $this->table_name . "
                  SET
                  rfid=:rfid, name=:name, price=:price";
        $stmt = $this->conn->prepare($query);
        // posted values
        $this->rfid=htmlspecialchars(strip_tags($this->rfid));
        $this->name=htmlspecialchars(strip_tags($this->name));
        $this->price=htmlspecialchars(strip_tags($this->price));
        // bind values
        $stmt->bindParam(":rfid", $this->rfid);
        $stmt->bindParam(":name", $this->name);
        $stmt->bindParam(":price", $this->price);
        if($stmt->execute()){
            return true;
        }else{
            return false;
        } } // create product
    public function update($id){
        //write query
        $query = "UPDATE
                  " . $this->table_name . "
                  SET
                  rfid=:rfid, name=:name, price=:price

```

```

        WHERE id=$id;";

$stmt = $this->conn->prepare($query);
// posted values
$this->rfid=htmlspecialchars(strip_tags($this->rfid));
$this->name=htmlspecialchars(strip_tags($this->name));
$this->price=htmlspecialchars(strip_tags($this->price));
// bind values
$stmt->bindParam(":rfid", $this->rfid);
$stmt->bindParam(":name", $this->name);
$stmt->bindParam(":price", $this->price);
if($stmt->execute()){
    return true;
}else{
    return false;    }
}

public function readByRfid($id){
$query = "SELECT * FROM
        " . $this->table_name . "
        WHERE rfid='$id'; ";

$stmt = $this->conn->prepare( $query );
$stmt->execute();
return $stmt; }

public function read($id){
$query = "SELECT * FROM
        " . $this->table_name . "
        WHERE id=$id; ";

$stmt = $this->conn->prepare( $query );
$stmt->execute();
return $stmt; }

public function readAll($from_record_num, $records_per_page){
$query = "SELECT id, rfid, name, price FROM
        " . $this->table_name . " ORDER BY name ASC LIMIT
        {$from_record_num}, {$records_per_page}";

$stmt = $this->conn->prepare( $query );
$stmt->execute();
return $stmt; }

public function countAll(){
$query = "SELECT id FROM " . $this->table_name . "''";
$stmt = $this->conn->prepare( $query );
$stmt->execute();
$num = $stmt->rowCount();
return $num;    }

public function getBillID(){

$query = "SELECT id FROM bills order by id desc LIMIT 1";
$stmt = $this->conn->prepare( $query );
}

```

```

$stmt->execute();
$row = $stmt->fetch(PDO::FETCH_ASSOC);
extract($row);
$num = $id+1;
return $num; }
public function delete($id){
$query = "DELETE FROM ".$this->table_name." WHERE id=$id";
$stmt = $this->conn->prepare( $query );
if($stmt->execute()){
    return true;
}else{
    return false; } }
public function saveTotal($amount){
$query = "INSERT INTO bills SET amount=:amount";
$stmt = $this->conn->prepare($query);
// posted values
$this->rfid=htmlspecialchars(strip_tags($amount));
// bind values
$stmt->bindParam(":amount,:products",$amount);
if($stmt->execute()){
    return true;
}else{
    return false; } } }

```

- **search products query:**

```

<?php
include("conn_db.php");
if (isset($_POST['SEARCH'])) {
    $id=$_POST['id'];
$sql = "SELECT * FROM products WHERE `id`=$id && CONCAT(`id`, `name`, `rfid`, `price`, `category`)";
$results=mysqli_query($conn,$sql);
if(mysqli_num_rows($results) >0){
    echo "<table>";
    echo "<tr>";
    echo "<th>id</th>";
    echo "<th>name</th>";
    echo "<th>rfid</th>";
    echo "<th>price</th>";
    echo "<th>category</th>";

    echo "</tr>";
    while($row = mysqli_fetch_assoc($results)){
        echo "<tr>";
        echo "<td>" . $row['id'] . "</td>";

```

```

        echo "<td>" . $row['name'] . "</td>";
        echo "<td>" . $row['rfid'] . "</td>";
        echo "<td>" . $row['price'] . "</td>";
        echo "<td>" . $row['category'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
}
?>

```

- **search bill query:**

```

<?php
include("conn_db.php");
// include("login.php");
// include_once 'config/database.php';
if (isset($_POST['SEARCH'])) {
    $id=$_POST['id'];
    $sql = "SELECT * FROM bills WHERE `id`=$id && CONCAT(`id`, `amount`)";
    $results=mysqli_query($conn,$sql);
    if(mysqli_num_rows($results) >0){
        echo "<table>";
        echo "<tr>";
        echo "<th>id</th>";
        echo "<th>amount</th>";
        echo "</tr>";
        while($row = mysqli_fetch_array($results)){
            echo "<tr>";
            echo "<td>" . $row['id'] . "</td>";
            echo "<td>" . $row['amount'] . "</td>";
            echo "</tr>"; }
        echo "</table>";}
    ?>

```

- **UpdateCart.json Query:**

```

<?php
include_once 'config/database.php';
include_once 'objects/product.php';
$total = 0.00;
$json_data = json_decode(file_get_contents('cart.json'), true);
for ($i = 0; $i < count($json_data); $i++) {
    $total += ($json_data[$i]["price"] * $json_data[$i]["quantity"]);
}
$database = new Database();
$db = $database->getConnection();
$product = new Product($db);
if (isset($_GET["rfid"]) && isset($_GET["action"])) {

```

```

$rfid = $_GET["rfid"];
    $action = $_GET["action"];
if ($action == "1") {
    $exists = false;
    for ($i = 0; $i < count($json_data); $i++) {
        if ($json_data[$i]["rfid"] == $rfid) {
            $json_data[$i]["quantity"] = strval($json_data[$i]["quantity"] + 1);
            $fp = fopen('cart.json', 'w');
            fwrite($fp, json_encode($json_data));
            fclose($fp);
            $exists = true;
            http_response_code(200);
            echo json_encode(dataLCD());
            break; }}
    if (!$exists) {
        $stmt = $product->readByRfid($rfid);
        $num = $stmt->rowCount();
        if ($num == 1) {
            $row = $stmt->fetch(PDO::FETCH_ASSOC);
            extract($row);
            $json_data[count($json_data)] = array(
                "rfid" => $rfid,
                "name" => $name,
                "price" => $price,
                "quantity" => "1");
            $fp = fopen('cart.json', 'w');
            fwrite($fp, json_encode($json_data));
            fclose($fp);
            http_response_code(200);
            echo json_encode(dataLCD());
            return;
        } else {
            http_response_code(400);
            return; }}
} else if ($action == "2") {
    $exists = false;
    for ($i = 0; $i < count($json_data); $i++) {
        if ($json_data[$i]["rfid"] == $rfid) {
            if ($json_data[$i]["quantity"] > 1) {
                $json_data[$i]["quantity"] = strval($json_data[$i]["quantity"] - 1);
                $fp = fopen('cart.json', 'w');
                fwrite($fp, json_encode($json_data));
                fclose($fp);
                http_response_code(200);
                echo json_encode(dataLCD());
                return;
            }
        }
    }
}

```

```

        } else {
            $exists = true;
            break;}}
    if ($exists) {
        $json_new = array();
        $index = 0;
        foreach ($json_data as $item) {
            if ($item["rfid"] == $rfid) {
                continue;}
            $json_new[$index++] = $item;}
        $fp = fopen('cart.json', 'w');
        fwrite($fp, json_encode($json_new));
        fclose($fp);
        http_response_code(200);
        echo json_encode(dataLCD());
        return;}
        http_response_code(200);
        echo json_encode(dataLCD());
        return;
    } else {
        http_response_code(400);
        return;}
} else if (isset($_GET["total"])) {
    $json_total = array(
        "total" => $_GET["total"]);
    if($_GET["total"]=="0"){
        $fp = fopen('cart.json', 'w');
        fwrite($fp, json_encode([]));
        fclose($fp);    }
    $fp = fopen('total.json', 'w');
    fwrite($fp, json_encode($json_total));
    fclose($fp);
    http_response_code(200);
} else if (isset($_GET["getTotal"])) {
    $product->saveTotal($total);
    http_response_code(200);
    echo json_encode(array("total"=>$total));
    return;}
function dataLCD(){
    $database1 = new Database();
    $db1 = $database1->getConnection();
    $product1 = new Product($db1);
    $rfid1 = $_GET["rfid"];
    $total_products1 = 0;
    $total_amount1 = 0.00;
    $json_data1 = json_decode(file_get_contents('cart.json'), true);

```

```

for ($i = 0; $i < count($json_data1); $i++) {
    $total_products1 += $json_data1[$i]["quantity"];
    $total_amount1 += ($json_data1[$i]["price"] *
$json_data1[$i]["quantity"]);
    $stmt1 = $product1->readByRfid($rfid1);
    $num1 = $stmt1->rowCount();
    if ($num1 == 1) {
        $row1 = $stmt1->fetch(PDO::FETCH_ASSOC);
        extract($row1);
        $json_data1 = array(
            "name" => $name,
            "price" => $price,
            "total_products" => $total_products1,
            "total_amount" => $total_amount1 );
    } else {      http_response_code(400);
        return; }
    return $json_data1;
}

```

- **Cart.json Query:**

```

<script>
$(document).ready(function() {
    setInterval(function() {
        $.ajax({
            type: 'GET',
            url: 'total.json',
            dataType: 'json',
            success: function(data) {
                console.log(data["total"]);
                if (parseInt(data["total"]) == 1) {
                    $.ajax( type: 'GET',
                        url: 'cart.json',
                        dataType: 'json',
                        success: function(data) {
                            if (data.length > 0) {
                                var sno = 1;
                                var total = 0;
                                $('#products').html("");
                                $.each(data, function(index, item) {
                                    var totalAmount = parseFloat(item["price"]) *
                                        parseInt(item["quantity"]);
                                            $('#products').append(
                                                '<tr>' +
                                                '<td scope="row">' + (sno++) + '</td>' +
                                                '<td>' + item["name"] + '</td>' +
                                                '<td>' + item["quantity"] + '</td>' +
                                                '<td>' + item["price"] + '</td>' +

```

```

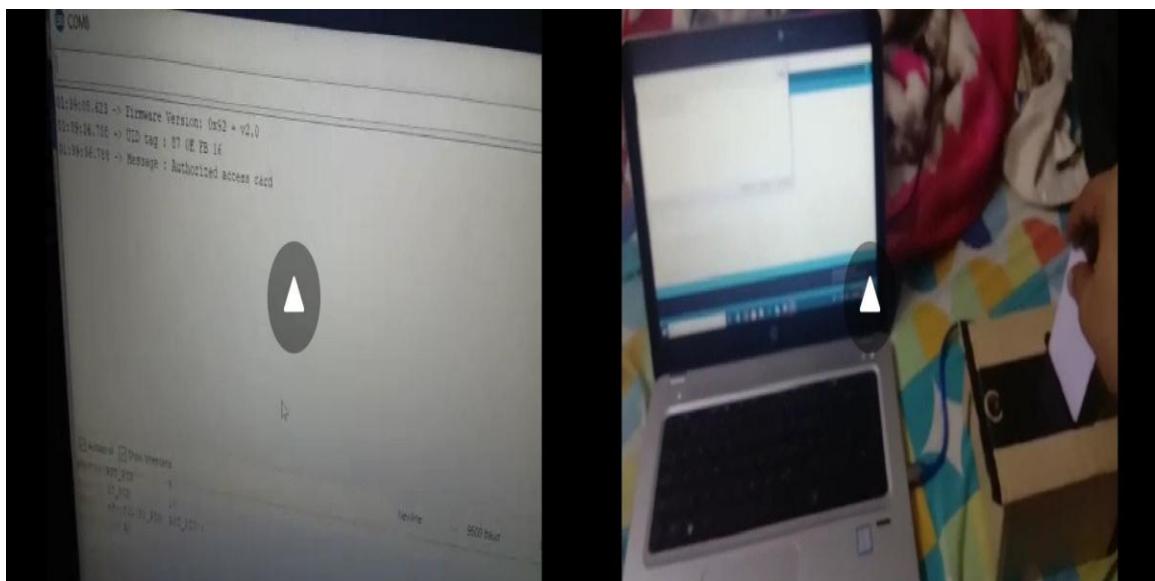
        '<td>' + item["rfid"] + '</td>' +
        '<td>' + totalAmount + '</td>' +
        '</tr>' );
    total += totalAmount;});
    $('#total').html(total);
} else {
    $('#products').html('<tr scope="row"><td
colspan="6">There are no items in the cart!</td></tr>');}
} });
} else { $('#products').html("");
    $('#total').html(""); }
}, 1000);
},
$('#printButton').click(function() {
    window.print();
    $.ajax( type: 'GET',
        url: 'updateCart.php?total=0',
        dataType: 'json',
        success: function(data) {
            } });
    $('#billID').html((parseInt($('#billID').text())+1));
});
</script>

```

- **Screenshots of a sample run of your project**







Chapter 6;

Test Cases

Test Plan

(Automatic Shopping Cart)

Purpose:

The test plan is created during the development or reverse engineering phase and identify all elements that are about to be tested. The test plan should explicitly describe what to test, what to expect, and how to do the testing. Subsequently it should be confirmed what was done, what was the result, and if the result was approved. The main purpose of writing this section is to test every feature of our project, including both the hardware, software aspects and security alarm system using RFID card access system and also performing different testing methodologies to test and check whether they are working optimally or not.

Test objectives:

- Finding problems which came up during the development process of the Cart Hardware.
- To test for Compatibility of the Different components of the Cart Hardware System and Security alarm and Web Based application.
- Gaining Confidence in and providing Information about the level of knowledge of the cart system.
- To Prevent surprise problems before rather than later in the Cart hardware system.
- To Ensure that the finished product met all the user requirements.
- To ensure that the Automatic Shopping Cart fulfill the system requirements specifications.
- To provide training and create confidence among the users of the Automatic Shopping Cart system by creating the impression that was tested and running successfully.
- To check the functionality of security alarm system using RFID card for authorized and Unauthorized access buzzer system

Scope:

The purpose of this report is to find and fix bugs in the project in order to obtain user experience and make our project more effective and error-free. All application functionality has been checked to add products, upgrade products, uninstall items, scan products and search bills to the server feature. This included both the Cart Hardware software and the ASC Web-based program.

Testing Methodologies:

Software testing methodologies are the different approaches and ways of ensuring that a software application in particular is fully tested. Software testing methodologies encompass everything from unit testing individual modules, integration testing an entire system to specialized forms of testing such as security and performance. The purpose of developing these strategies is to ensure completeness when it comes to the project's construction, that it includes all of the system features, is completed within the allotted timeline and budget, and allows for plans to scale the testing efforts even more. Testing strategy will affect test planning, test type, test script development, and test execution tasks.

- **Black Box Testing:**

Black box analysis, also known as behavioral, opaque-box, closed-box testing, is a test method that analyzes the usability of the software / application without knowing much about the internal structure / design of the object being evaluated and compares the input value to the output value. It is a technique which ignores the system's internal mechanism and focuses on the output produced.

- **White Box Testing:**

White Box Testing is a test of the internal coding and functionality of the software solution. It focuses primarily on enhancing security, the flow of inputs and outputs through application, and improving design and usability. White box testing is also known as Clear Box Testing, Open Box Testing, Structural Testing, Transparent Box Testing, Software Based Testing, and Glass Box Testing.

Levels of Testing:

Testing is the method of testing a code object to identify the variations between the input and the predicted output. Also, to determine the functionality of the computer object. Analysis measures the quality of the product. Software testing is a process that should be carried out during the development process. In other words, code testing is a method of verification and validation.

Unit Testing:

It includes testing different modules individually and ensuring the proper functioning of each module. Unit testing is a software testing method by which individual source code units, a set of one or more computer program modules, together with the associated control data, usage procedures and operating procedures, are tested to determine whether they are suitable for use.

Modules Tested:

- WIFI (hotspot) connectivity.
- Products scan function in the Hardware.
- Add product function in the Hardware.
- Remove Button function in the Hardware.
- Reset Button function in the Hardware.
- Send Bill function in the Hardware.
- Total amount function display in LCD.
- No# of products displays in LCD.
- Different Bill ID for each Bill.
- Bill Generated in the System.
- Security alarm Buzzer (authorized access/unauthorized access).

Integration Testing:

This involves testing two or more modules together to ensure the compatibility of the various modules. This is the software testing process in which single software modules are integrated and evaluated as a group. This occurs after device testing and validation testing. Integration testing shall be carried out as its unit-tested input components, grouped into larger aggregates, and tests shall be used to identify the component.

Functional Testing:

This type deals with the functional requirements or specifications of an application. By providing the input and comparing the actual output with the expected output, different actions or functions of the system are being tested here.

Regression testing:

Regression testing is the method of checking computer program improvements to ensure existing programming still functions with the new changes. It is done after software updates, improvements or any other system maintenance to verify that the current code has not been compromised.

Hardware Testing:

Hardware testing is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. Each piece of hardware equipment is also individually tested to see if it works on its own.

Performance Testing:

Quality testing is the method of evaluating a system, network, software program or device's speed, responsiveness, and stability under a workload. Performance testing can require quantitative testing conducted in a laboratory or occur in specific scenarios in the production environment. Typical parameters include speed of transmission, data transfer frequency, network bandwidth, quality of work and reliability.

Usability Testing:

Usability testing is important to develop strategies for finding out how systems work practically when they are delivered to customers.

- Testing Application item like buttons, different tabs.
- Application should be easy to understand
- Instructions provided should be very clear to the end user.

GUI Testing:

Interaction between the various components of an application and the user will affect the product's overall efficiency, so this interaction, i.e. the software must also be checked and confirmed if there are no graphical bugs.

Compatibility Testing:

Compatibility Testing is a form of software testing to check if the code can run on different hardware, operating systems, applications, network environments or different devices. Compatibility testing is one form of non-functional testing.

System Testing:

It is a software test level which tests a complete and integrated software. The purpose of this test is to evaluate the system's compliance with the specified requirements. This involves testing of the whole system. Here we tested whole Automatic Shopping Cart with web-based application and security alarm.

Test Cases Factors:

1. Test Entry / Exit Criteria

This sets out the parameters suggesting a successful completion of a test phase. The exit requirements are the test's expected outcomes, and are required before progressing to the next development phase. Example: it must pass 95 percent of all-important test cases. (LibStudies, n.d.)

2. Test Deliverables

Test Deliverables is a list of all the documents, software and other resources needed to be created and maintained to support the testing effort.

At each step of the software development lifecycle there are different test deliverables.

Test deliverables are provided **before** testing phase.

- Test plans document.
- Test cases documents
- Test Design specifications.

Test deliverables are provided **during** the testing

- Test Scripts
- Simulators.
- Test Data
- Test Traceability Matrix
- Error logs and execution logs.

After the test cycles are over, test deliverables are provided.

- Test Results/reports
- Defect Report
- Installation/ Test procedures guidelines
- Release notes

3. Test Suspension / Resumption Criteria

Specify the criteria critical for a test suspension. If the criteria for the suspension are met during the test, the active test cycle will be suspended until the criteria have been resolved. For example, if our team members claim that 40 percent of test cases have failed, you will postpone testing until all the failed cases are resolved by the development team.

4. Test Environmental / Staffing / Training Needs

A testing environment is a software and hardware system that the testing team must run test cases on. The research environment consists of real business environment and user etc.as well as physical environments such as server, front-end running environment.

Test Cases (Automatic Shopping Cart)

Unit Testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information about all of the main components or building blocks of the application and hardware.

3. Test Cases / Scenarios

4.

- Checking Connectivity WIFI (Hotspot)

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Node MCU Module	Assign SSID Username and Password	Device should connect	Hotspot connected	Pass

- Checking RFID Reader

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Scan the RFID Tags	RFID Tag	RFID Tags detail	RFID Tag Number	Pass

- Checking Products details of RFID tags Displayed in LCD

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Scan the RFID Tags	Product Name, Product Price,	Device should Display NOP, POP	Device should Display NOP, POP	Pass

- Checking Total Amount of Products Added Displayed in LCD

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Scan the RFID Tags	Adding Products	Device should Show Total Amount	Total Amount Displayed	Pass

- **Checking Number of Products Displayed in LCD**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Scan the RFID Tags	No of Products	Device should Show No of Products	No of Products Displayed	Pass

- **Checking Complete Working of Remove Button**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Press Remove button	Mode (Negative)	Device should Mode (-) on LCD	Mode (-) on LCD	Pass
2	Scan the RFID Tags	Mode (-) Product Name, Product price	Device should Remove Product	Remove Product from Device	Pass

- **Checking Complete Working of ADD Button**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Press Remove button	Mode (Add)	Device should Mode (+) on LCD	Mode (+) on LCD	Pass
2	Scan the RFID Tags	Mode (+) Product Name, Product price	Device should ADD Product	ADD Product from Device	Pass

- **Checking Complete Working of Total Bill Button**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Scan the RFID Tags with Product Detail	Product Name, Product Price	Device should Display NOP, POP	Device should Display NOP, POP	Pass
2	Adding Products	Product Name, Product Price	Device should Show Total Amount	Total Amount Displayed	Pass
3	Press Total Bill button	Total Bill	Device should Show Total Bill	Device should Show Total Bill	pass

5. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry.

6. Test Environmental / Staffing / Training Needs

Proper environment should be set up, PC, laptop should be up to date and bug free.

Integration Testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information about the integration and communication of main components of the application and hardware.

3. Test Cases / Scenarios

- Checking Complete Pairing of Devices**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Node MCU Module	Assign SSID Username and Password	Device should connect	Hotspot connected	Pass
2	Pairing with the module	Pairing local IP address	Pair should be successful	Devices paired	Pass

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
4	Transferring commands to Node MCU	Commands sent via Node MCU ESP2866 connection	Node MCU ESP2866 should acknowledge command and act	Node MCU ESP2866 acts	Pass

4. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry.

5. Test Environmental / Staffing / Training Needs

Proper environment should be set up, PC, laptop should be up to date and bug free.

Functional Testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information about all of the main components and the functionalities of the application and hardware.

3. Test Cases / Scenarios

- Checking Functionality of Node MCU ESP8266 With Web Page

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Connecting Web page with NodeMCU via Esp2866	WIFI module and signals	WIFI should connect devices	Devices Paired	Pass
2	Signals send via Web Page	Node MCU should recognize Local host IP address	Commands should execute by RFID Reader	Command executed properly	Pass

- Checking failed Functionality of Bluetooth Command Recognition

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Connecting Web page with NodeMCU via Esp2866	WIFI module and signals	WIFI should connect devices	Devices Not Paired	Fail
2	Signals send via Web Page	Node MCU should recognize Local host IP address	Commands should execute by RFID Reader	Command not executed properly	Fail

4. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry

5. Test Environmental / Staffing / Training Needs

Proper environment should be set up, PC, laptop should be up to date and bug free

Regression testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information about all of the main components their updates and condition after the upgrades.

3. Test Cases / Scenarios

- Testing Device After Upgrading Battery/ Power Bank

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Replacing the 7V battery with a higher capacity 5V one	Re-establishing connections via the wires	Node MCU should power up	LED powers up	Pass
2	Battery should give a longer duration of function	High capacity of battery	The hardware should stay powered up for longer	The hardware endures for longer	Pass

4. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry

5. Test Environmental / Staffing / Training Needs

Proper environment should be set up, PC, laptop should be up to date and bug free. Safety measures should be taken as we are dealing with hardware.

Hardware Testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information about all of the main hardware components and their condition.

3. Test Cases / Scenarios

- **Testing Device Connection via Wires**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Powering up the Node MCU via battery/ power Bank	Establishing connections to the Node MCU via Soldering with Wire	Node MCU LED should power up	LED power ups	Pass

- **Checking Connectivity WIFI (Hotspot)**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Node MCU Module	Assign SSID Username and Password	Device should connect	Hotspot connected	Pass

- **Checking RFID Reader**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Scan the RFID Tags	RFID Tag	RFID Tags detail	RFID Tag Number	Pass

- **Testing Node MCU Code Uploading**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Powering up Node MCU	Establishing connections to the Node MCU via wires	Node MCUs LED should power up	LED power ups	Pass
2	Connecting the Node MCU with a laptop via USB cable	USB ports and cable	The laptop should recognize Node MCU	Node MCU recognized	Pass
3	Using the Arduino IDE to interact with the board	Uploaded code to the Arduino IDE	Code should be uploaded to the Arduino IDE	Code uploaded	Pass
4	Checking functionality of uploaded code	Commands sent to the Arduino to check uploaded code	Node MCU should perform its programmed functionalities	Functionalities performed	Pass

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Connecting RFID Reader to a preprogrammed Node MCU with an LED	Connecting the RFID Reader to via wires	RFID Reader should be connected properly	RFID Reader connected	Pass

- **Testing Battery Rating**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Using a multimeter to measure specification of battery	Battery	The voltage and ampere rating of battery should be revealed(5V)	Specifications are revealed completely	Pass

4. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry

5. Test Environmental / Staffing / Training Needs

Safety measures should be taken as we are dealing with hardware.

Performance Testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information about responsiveness and load bearing capabilities of the application and hardware.

3. Test Cases / Scenario

- **Testing Load Time for Login Web Page:**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Launching web page	(CARTIGO)login page with username and password	login web page should launch	login web page launches	Pass
2	Measuring time taken for application to launch	Time taken to load	Web page should launch faster	The launch is fast	Pass

- Testing for Username and Password is Valid:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	enter a valid username and password in the login page	username and password	logged in to Homepage	login to Homepage	Pass

Testing for Username and Password is not valid:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	enter incorrect username and password in the login page	username and password	Redirect to Login page	Redirect to Login page	Pass

- Testing add products to Database:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to Add product from Homepage	add new products	display add product box	add product box displayed	Pass
2	Enter RFID, Name, price	RFID, Name and Price	product should be added to database	product added to Database	Pass

- Testing Update products to Database:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to Add product from Homepage	update products	display update product box	update product box displayed	Pass
2	Update RFID, Name, price	RFID, Name and Price	product should be Updated to database	product Updated to Database	Pass

- Testing Delete products from Database:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to Add product from Homepage	delete products	product should be deleted from database	product deleted	Pass

- Testing search products through ID from Database:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to Search product from Homepage	product ID	display product information	Products information displayed	Pass

- Testing Searching Bill through ID from Database:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to Search Bill from Homepage	Bill ID	display ID and amount of Bill	Bill ID and amount displayed	Pass

- Testing Get Bill with Bill ID from via total bill Button in Hardware:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to get Bill from Homepage	bill ID, total amount, time/date	should display bill format	Bill displayed	Pass

4. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry

5. Test Environmental / Staffing / Training Needs

Proper environment should be set up, PC, laptop or smartphone should be up to date and bug free. Safety measures should be taken as we are dealing with hardware.

Usability Testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information on how it is easy for a user to use the application and hardware.

3. Test Cases / Scenarios

- Checking Apparel of Navigation of web page:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Checking Interface of web page	Login, Navigation tab and buttons on webpage	It should be easy to understand	Interface is simple	Pass

- Checking Apparel of Navigation pages:

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Checking Navigation of pages on web page	Text used on buttons or tabs	Text should be plenty and not obscured	Text are clear to understand	Pass

- Checking Relevancy of Information

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Webpage having data from database in search product and search Bill	product details and bill details	Application should show relevant information	Information is relevant	Pass

4. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry

5. Test Environmental / Staffing / Training Needs

Proper environment should be set up, PC, laptop or smartphone should be up to date and bug free.

GUI Testing:

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information about the interface, design and style of the application.

3. Test Cases / Scenarios

- Checking Alignment of Elements**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Checking Alignment of buttons	Buttons on pages	Buttons should be of same size and alignment	Buttons are properly aligned	Pass

- Checking Alignment of tabs**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Checking Alignment of tabs	tabs on Homepage, search and search Bill	tabs should be of same size and alignment	Tabs are properly aligned	Pass

- Checking Visibility of Elements**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Checking Visibility of buttons	login, homepage, search products, search bill	Buttons should be clear	Buttons are properly visible	Pass

- **Checking Visibility of Elements**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Checking text display of tabs used for switching screens	Tabs on the bottom and top of the application	text tabs should be clear and appropriate	tabs are proper	Pass

4. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry

5. Test Environmental / Staffing / Training Needs

Proper environment should be set up, PC, laptop or smartphone should be up to date and bug free.

Compatibility Testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information regarding the devices that can support the application and hardware.

3. Test Cases / Scenarios

- **Testing Application working and failure on Various Devices**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Open webpage on Google chrome	Login form	Google chrome should run without trouble	Login form Displays Application runs	Pass

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Open webpage on Internet explorer	Login form	Internet explorer should run without trouble	Login form Displays Application runs	Pass

- **Checking Working Slow and Success**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Open webpage on Opera Mini	Login form	Google chrome should run without trouble	Login form Displays Application runs slowly	Pass

4. Test Suspension / Resumption Criteria

If any scenario fails, it will be suspended or sent for resumption or for retry

5. Test Environmental / Staffing / Training Needs

Proper environment should be set up, PC, laptop or smartphone should be up to date and bug free. Safety measures should be taken as we are dealing with hardware.

System Testing

1. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

2. Test Deliverables

Information about all of the main components working together as an end product.

3. Test Cases / Scenarios

- Checking Overall Navigation

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	login to Webpage	logo image on Home-screen	Web Page should load without any issues	loads successfully	Pass
2	Navigating through screens	Symbols/text or tabs for all three interfaces	Navigation should be without any issue	Navigation is successful	Pass
3	adding products	product name and price and RFID tags	should add products	products added	Pass
4	update products	product name and price and RFID tags	should update products	products updated	Pass
5	delete products	product name and price and RFID tags	product deleted	products deleted	Pass
6	search products	product ID	should display product information	products information displayed	Pass
7	Search Bill ID	Bill ID	should display bill information	Bill ID and amount displayed	Pass
8	get Bill	get bill with products, price and quantity and RFID tags	should print bill format	print bill format	Pass
9	Logout	Logout button	should logout from webpage	logged out from webpage	Pass

- Checking Software and Hardware Reliability

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Loading webpage with hardware in another device	Login form and connectivity with Hardware	webpage and Hardware should connect without any issues	connectivities problems	Fail

- Checking overall Status of components

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Supplying power to the Automatic Shopping Cart via power bank	Connecting the power bank to the hardware	Hardware should power up	Hardware powers up	Pass
2	Checking supply to Cart	LED on of ASC Cart	LED should light up	LED lights up	Pass
3	Checking supply to Node MCU	LED on of WIFI node MCU module	LED should light up	LED lights up	Pass
4	Checking status of adding products to Cart	scanning the products to add	products should be added by scanning	Products added to Cart	Pass
5	Checking status of removing products to Cart	press the remove button then scan products	product name and price should be reduced accordingly	Product and product prices are reduced from the amount and cart	Pass
6	Checking status of total bill to the Cart	products, price quantity, total amount and Bill ID	get the print bill with bill ID accordingly	print bill with Bill id and products	Pass

RFID Security Alarm System:

Hardware Testing:

6. Test Entry / Exit Criteria

All of the components should be working for passing entry criteria. Exit criteria will only happen if the scenario passes tests.

7. Test Deliverables

Information about all of the main hardware components and their condition.

8. Test Cases / Scenarios

- **Checking security alarm system working (Idea):**

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Connecting the USB cable connection	Connecting the USB cable to the hardware	Hardware should power up	Hardware powers up	Pass
2	Checking Arduino IDE port connection	connect with port which will let LED & Buzzer on	LED should light up and buzzer initialize	LED light up and Buzzer initializes	Pass
3	Giving RFID Card authorized Number are in the program	RFID Number Access	RFID number given in authorize access should run on authorize condition accordingly	Buzzer run in authorize conditions applies	Pass
4	Giving RFID Card Unauthorized Number that are not in the program	RFID Unauthorized Card	RFID number has no access should run on Unauthorized condition accordingly	Buzzer runs in Unauthorized condition applies	Pass

- Checking Overall Security Alarm System (Idea);

Step No.	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Connecting the USB cable connection	Connecting the USB cable to the hardware	Hardware should power up	Hardware should power up	Pass
2	Checking Arduino IDE port connection	connect with port which will let LED & Buzzer on	LED should light up and buzzer initialize	LED light up and Buzzer initializes	Passl
3	Verifying the Bill	Bill with Bill ID and products	Bill should have the products added in the cart	Bill with associated cart products have authorize card	Pass
4	Driving Cart through the person given Card should be scan in hardware	Authorized access	Buzzer will buzz normal	Buzzer buzz 2(low) and message: access	Pass
5	Driving Cart through the person given Card should be scan in hardware	Unauthorized access	Buzzer will not buzz normal	Buzzer buzz 3(HIGH)and alert message (not-access)	Pass
6	Driving Cart through the person can scan any RFID Card in hardware	Unauthorized access	Buzzer will not buzz normal	Buzzer buzz 3(HIGH)and alert message (not-acces)	Pass
7	ARduino IDE is connected with Hardware	Arduino IDE, Port, Serial Monitor	System connected and Buzzer should run when Card scanned alert message generated in arduino IDE	System connected and Buzzer run when Card scanned and alert message generated in arduino IDE	Pass

Chapter 7; Conclusion and Future Work:

CONCLUSION:

This proposed system works on improving customer satisfaction as on customer enter the supermart first customers take a cart which has a Cart ID number, Cart have hardware application with an RFID reader with Node MCU and LCD. Customer wants to add products in the Cart by scanning tags of the product that will be read by the reader. If the Tags matches with the product ID, then the cost of that product will be displayed on the LCD in the Cart. If the customer wants to remove product from the cart, the customer can take away that product from Cart and scanned again and cost of that product will be reduced from the total amount, and after shopping the products data with total amount gets transferred to the CBU through ESP-8266. The database at the central billing system will be updated when add/remove from the cart. And at the end the customer sends the data by pressing the send button on hardware, which generate the bill associated with trolley Cart ID. The cashier will verify Cart ID and customer will pay the bill without hectic of queues.

- ❖ **Good things:**
 - A potential solution to the problem was built.
 - Basic Solution for the cashier was also built.
 - Most of the issues with RFID based totaling were resolved.

- ❖ **The following things could be done in future to improve project:**
 - Integration with a real super mart.
 - if you have addressed real hardware design issues.
 - If you have conducted a survey with supermarkets to discuss the issues with your solution.
 - If you have solved the issue of theft and reduce the risk factors.
 - If you have solved the issue of transferring data from the cart to a cashier where there is more than one cashier.
 - If you have integrated your system with a real-life system.