

Electronic Trolley

Saba Mustafa¹, Saleem Raza²

^{1,2}Department of Computer Engineering in Cyber Security, Middle East College, Muscat, Oman.

Abstract

Nowadays shopping requires a lot of time and effort to search for each aisle for a specific product, trying to read the small information on the product even though some of the products have no further information just the barcode and the country or the region of the product. Also most of the time there is no price tag on the products which makes customers keep all the products in the trolley until the checkout comes and then get shocked by the bill as they cannot keep track of the product's prices, the boring and long-lasting process of keeping products inside the trolley, unlimited time-wasting throughout the checkout lanes in which sometimes it encourages customers to abandon the cart and leave.

To ease the shopping process for customers we are developing an electronic trolley that provides several benefits for both customers and retailers. By enabling anytime, anyplace shopping without the limitations of actual store hours or locations, they improve convenience for customers. Easier for customers to compare items, prices, and features by allowing them to do so quickly and easily. Electronic shopping carts help retailers by streamlining processes, automating inventory control, and enabling data analytics-driven targeted marketing campaigns. Also, offers a smooth checkout experience so that clients may quickly and safely assess their options, select their preferred payment option, and complete their purchases.

Keywords: Electronic Trolley, Arduino, Sensors, physical Diagram, Logical Diagram, Network Diagram.

1. INTRODUCTION:

Our lifestyle structure needs to adapt to the growth of technology. In 2024, our approach needs to be focused on merging modern technologies into every aspect of what we do. accepting advances in technology is essential to staying relevant in our industry. A culture of constant improvement is what we must dedicate ourselves to. This involves turning traditional, manual shopping into an electronic trolley shopping experience.

The environmental impact of our daily activities especially when it comes to shopping has become a crucial factor in the lives of people, and the traditional purchasing of groceries experience plays a major role. My bachelor project offers a ground-breaking solution in reaction to this: a cutting-edge, environmentally responsible electronic trolley designed to maximize shopping enjoyment.

2. LITERATURE REVIEW

A) A.Sami "intelligent trolley"

The Intelligent Trolley has improved efficiency and convenience, revolutionizing the retail shopping experience. Long checkout lines are eliminated, and integrated RFID (Radio Frequency Identification) cuts down on the amount of time spent on product hunting. By integrating into the store's inventory system, RFID technology enables trolleys to automatically recognize and count things without physical contact,

directing customers to the locations of products. The shopping experience is further streamlined by additional features like internet access for organizing shopping lists.

B) Clark, L., Wright, P “off their trolley- understanding grocery shopping behaviour”

This study of the literature explores the complex and multifaceted world of online consumer behavior, with an emphasis on grocery shopping in the UK. This article presents the framework, which provides a thorough method of modeling consumer behavior that is more accurate and detailed than traditional models. The survey's findings highlight important interaction design factors, such as the need to enhance search and scan features, support loyalty card initiatives, and find a way to balance user autonomy with experimentation. All things considered, this PDF is a priceless tool for anyone trying to understand and plan for the intricacies of online consumer behavior, especially when it comes to grocery shopping.

C) D.Urankar, A.Giri, V.Raiputu, A.Dhavalikar, “Smart shopping trolley system”

The Smart Shopping Trolley System offers a novel way to prolong shopping hours, shorten checkout lines, and raise consumer satisfaction levels all while enriching the shopping experience. This approach makes purchasing easier and faster by utilizing technology and providing a variety of payment choices. Because of its adaptability, it may be used in a variety of retail environments, such as supermarkets, grocery stores, clothes showrooms, and shopping centres. All things considered, the Intelligent Buying Trolley System proves to be an intelligent, economical, scalable, and durable solution that may greatly improve consumers' shopping experiences in a variety of retail settings.

3. PROBLEM DEFINITION

By offering an online webpage, the electronic cart risks solve many of the issues present in traditional shopping experiences. Using the latest technology, a shopping cart solves problems like restricted product information and manual inventory management, while also improving accessibility for clients. The challenges with traditional cash-based transactions are overcome using secure payment gateways, which guarantee a safe transaction environment. In addition, and leverages consumer information to enable specific advertising, all of which contribute to the creation of a quick and customized shopping experience. By lowering the environmental effect of traditional shopping techniques, the shopping cart project not only satisfies the needs of the modern market but also advances goals for sustainability.

4. OBJECTIVES

This project is designed to provide a modern shopping experience, an Electronic Trolley serves as a modern shopping partner. One of its most important characteristics is the addition of barcode reading technology, which simplifies the checkout process by enabling customers to easily scan things while they purchase. The trolley focuses on energy efficiency by utilizing low-power components. To improve user engagement, a smart screen on the trolley provides up-to-date information on products, discounts, and promotions. An integrated payment system reduces the need for traditional checkout counters by enabling users to make purchases instantly through the trolley for smooth transactions.

5. THE DESIGN OF THE TROLLEY

The below figure represents the flow chart of the proposed design of the electronic trolley.

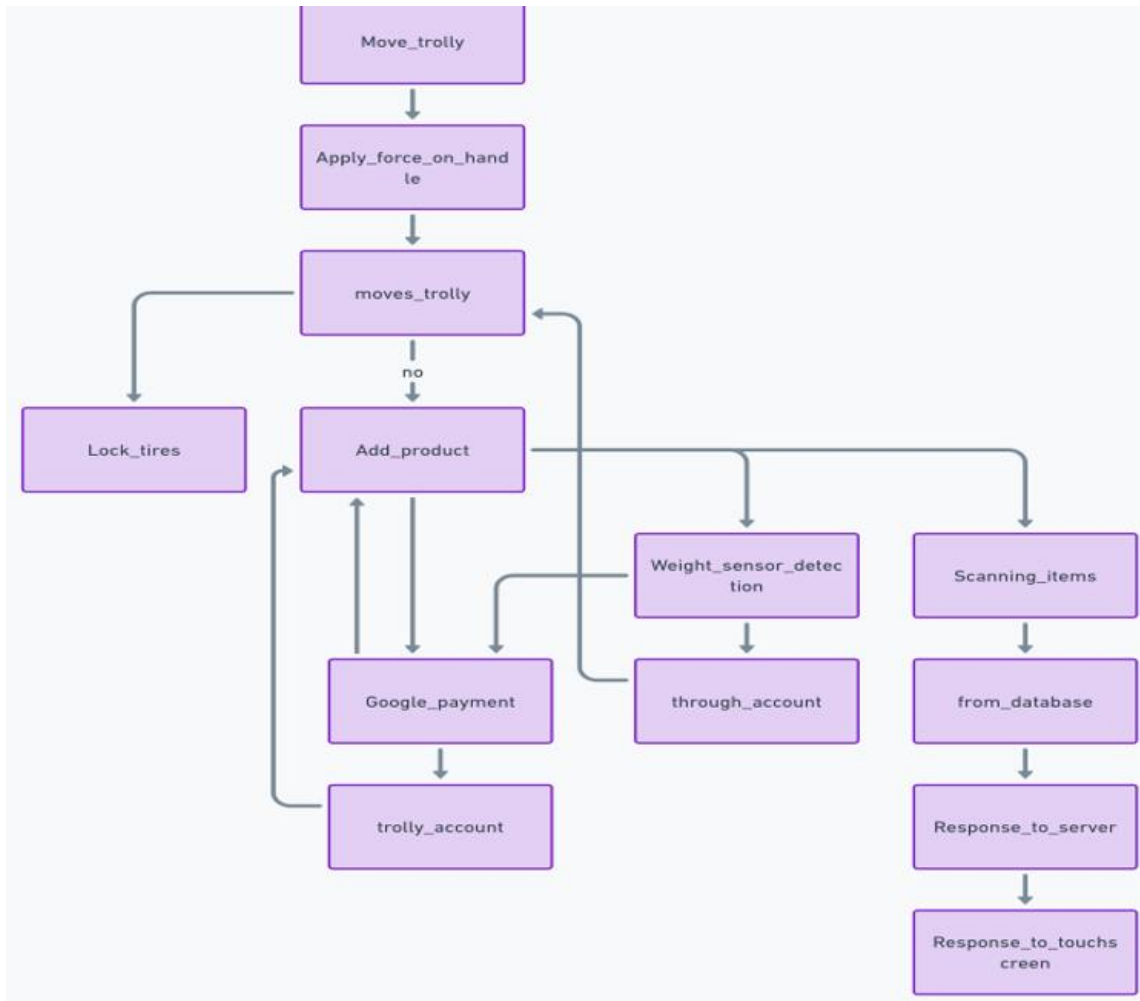


Figure 1

Start applying force on the handle of the trolley to move it forward. If a different customer moves the trolley, face ID recognition will detect a different face from the one that is using the current trolley, send commands, and automatically lock the tires of the trolley. Customers collect the products.

The customer scans the product on the touch screen and then places it inside the trolley. The item is added by the first network request which is an HTTPS request for the server which is a code kept on the computers of the supermarket. The server will request the database and take information from it depending on the type of request it has got, from the database it will go back to the server, and from the server, it will go back as a response to the screen on the trolley with the changes that happen which in this case adding an item to the trolley. All the communication through the network and HTTPS is secured and the information and data between them is encrypted. then you can simply keep the product inside the trolley as it is updated on the website, now all the items are added successfully to the database and the cart. It is updated on the website, and now all the other items are added successfully to the database and the cart. The trolley will confirm the product with the weight sensor, which will measure the weight of the product if it matches what has been scanned. The products that have been added will be shown on the touch screen that is attached to the handle of the trolley, so the customer will be able to see the product information, code, price, and quantity. Customers will also be able to add or remove the product from the touchscreen first and then remove it from the trolley as the weight sensor will detect the same amount of weight that

has been removed from the touch screen. When the customer finishes the shopping experience he will click on the finish button that is clearly shown on the touch screen so now the QR code has popped on the screen, the customer will scan the code on his phone, bill will be shown on the phone, so now the customers can easily pay with all of the information stored on the user account that he first signed in when using the trolley for the first time.

6. SEQUENCE DIAGRAM

Below is the diagram of the proposed sequence of the working of the trolley.

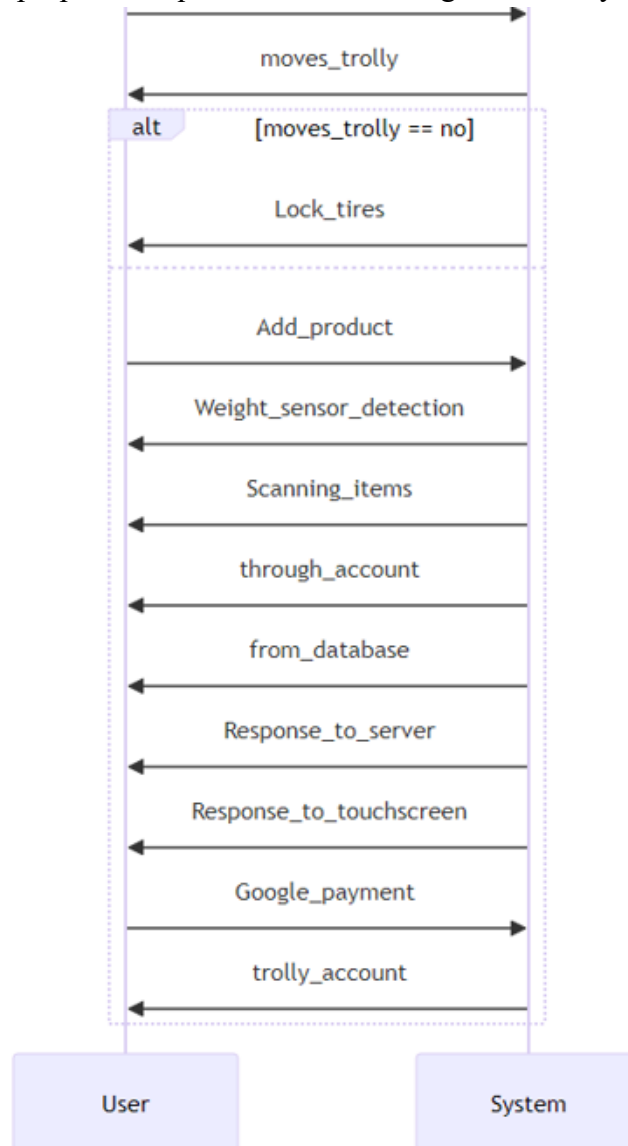


Figure 2

The sequence diagram illustrates when a customer moves a trolley and then it detects other customers face ID it will lock the tires of the trolley. Then when adding an item or a product inside the trolley, by scanning the product then weight sensor detects it there is a match between the scanned product with the same weight detected through the account that has been signed in the information and data is transferred from the database to the server, from the server back to the touchscreen on the handle of the trolley. Last step is finalizing the payment by using google pay by a QR code on the screen to be scanned from the customers phone through using their bank account application.

7. SIMULATION

The below figure represents the simulation of our project:

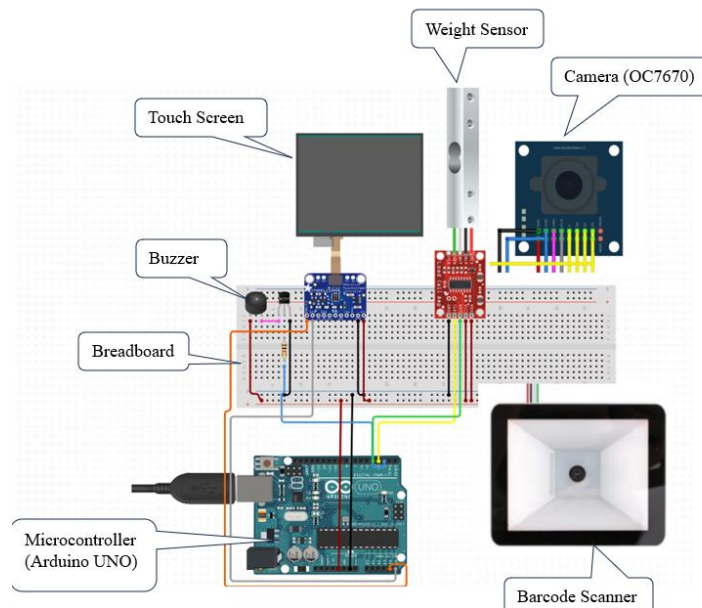


Figure 3

The simulation of the project undergoes the different parts connected on the breadboard and connected with the help of Arduino to form a system that will be used for the working of the electronic trolley.

This simulation illustrates the different parts that will be done inside of the trolley, and it will not be shown to the customers. The above simulation illustrates how all the parts are connected to the Arduino on the motherboard, the parts that are barcode scanner, camera (OC7670), weight sensor, touchscreen, and buzzer, all of these parts work because of the Arduino, if the Arduino will be having any error the whole system will be affected and the process will have trouble or in some cases, it will stop working.

The advantages of simulating an electronic trolley include **cost savings**.

saves time and money by eliminating the requirement for physical prototypes as well as testing.

Risk mitigation **lowers the chance of failures** by spotting possible problems before the design phase.

Performance optimization: Assists in adjusting the design for maximum effectiveness and performance.

8. CONCLUSION

The project is created at a minimal cost and power usage. Customers may enjoy shopping without moving a typical shopping cart thanks to this technology. Utilizing sensors on the shopping cart to facilitate the operation, track the total, and maximize the capacity to add and identify items. Additionally, this computerized trolley benefits those who dislike interacting with employees.

9. FUTURE IMPROVEMENTS

The following are some potential future features that could be included on the website or the trolley itself:

1. Obstacle Detection:

The trolley may combine sensor data from several sources to improve its obstacle-detection abilities. Combining information from the infrared ultrasonic, and optical sensors, for instance, can offer a more thorough grasp regarding the trolley's surroundings and increase the accuracy of obstacle recognition. It is possible to use machine learning methods to gradually improve the trolley's obstacle-detecting skills.

Through the examination of historical sensor data and the trolley's related movements, these algorithms can adjust and gain knowledge from past experiences, enhancing performance in a range of situations. Predictive analytics could also be used by the trolley to foresee possible roadblocks based on past performance and environmental variables. Through proactive trajectory adjustment before to obstacle meeting, the trolley achieves higher levels of efficiency (*Improve Store Operations with Smart Carts, n.d.*).

2. Route Planning:

To dynamically modify routes in response to changing conditions, sophisticated route planning algorithms may include real-time feedback loops. For instance, the cart might redirect itself in real time to reduce delays and maximize efficiency if an unexpected obstacle or traffic jam occurs. Algorithms for collaborative route planning can let many trolleys operating in the same area coordinate with one another. Trolleys can prevent crashes and traffic jams by exchanging details about their proposed routes and present positions, which facilitates smoother operations overall. In addition to these limitations and goals, route design algorithms may take into account factors like electric trolley battery life and energy efficiency. The technology can extend trolley runtime and lower overall operating costs by improving routes that reduce energy usage while still meeting operational requirements.

3. Integration with Store Inventory Systems:

Demand projections and restocking planning are two examples of more advanced inventory optimization techniques that may be made possible by the integration of the trolley with store inventory systems. Through an examination of past sales information and market patterns, the system is better equipped to forecast future demand and modify inventory levels appropriately. Product groups or larger inventory categories can be included in real-time inventory tracking in addition to individual products. More strategic decision-making is made possible by the system's ability to spot trends and patterns in inventory levels across categories that might not be obvious when looking at individual products.

Customizing of shopping trips based on individual consumer desires and purchase history is made possible by integration with CRM (customer relationship management) systems. Retailers can improve customer satisfaction and loyalty by offering customized rewards and recommendations based on customer data gathered by the trolley.

10. ACKNOWLEDGEMENT

I would like to express my thanks especially to Mister Saleem Raza and in general to all the faculties that helped in the completion of this project. Mister Saleem has guided me from the beginning on all the requirements and illustrations I needed till the end of the project. I would also like to thank the Department of Computer Engineering (cyber security) for contributing and helping me to succeed in the project.

11. RESULTS

```

39 scale.set_scale(calibration_factor);
40 scale.tare(); //Assuming there is no weight on the scale at start up, reset the scale to 0
41
42 if (! touch.begin()) Serial.println("Touch controller not found!");
43 menuOption = menu();
44
45 }
46
47 // Main logic of your circuit. It defines the interaction between the components you selected. After setup, it runs over and over
48 void Loop()
49 {
50
51
52 if(menuOption == '1') {
53 // Buzzer - Test Code
54 // The buzzer will turn on and off for 500ms (0.5 sec)
55 buzzer.on(); // 1. turns on
56 delay(500); // 2. waits 500 milliseconds (0.5 sec). Change the value in the brackets (500) for a longer or shorter
57 buzzer.off(); // 3. turns off.
58 delay(500); // 4. waits 500 milliseconds (0.5 sec). Change the value in the brackets (500) for a longer or shorter
59 }
60 else if(menuOption == '2') {
61 // SparkFun HX711 - Load Cell Amplifier - Test Code
62 float scaleUnits = scale.get_units(); //scale.get_units() returns a float
63 Serial.print(scaleUnits); //You can change this to lbs but you'll need to refactor the calibration_factor
64 Serial.println(" Kg"); //You can change this to lbs but you'll need to refactor the calibration_factor
65 }
66 else if(menuOption == '3') {
67 // Resistive Touch screen - 3.7" Diagonal - Test Code
68 uint16_t x, y;
69 uint8_t z;
70 if (touch.touched()) {
71 // read x & y & z;
72 while (! touch.bufferEmpty()) {
73 Serial.print(touch.bufferSize());
74 touch.readData(x, y, z);
75 Serial.print("->");

```

Figure 4

```

113
114 if(c == '1')
115     Serial.println(F("Now Testing Buzzer"));
116 else if(c == '2')
117     Serial.println(F("Now Testing SparkFun HX711 - Load Cell Amplifier"));
118 else if(c == '3')
119     Serial.println(F("Now Testing Resistive Touch screen - 3.7" Diagonal"));
120 else
121 {
122     Serial.println(F("illegal input!"));
123     return 0;
124 }
125 time0 = millis();
126 return c;
127 }
128 }
129 }
130
131 //*****

```

Figure 5

12. REFERENCES:

1. Sarala, T., Sudha, Y. A., Sindhu, K. V., Suryakiran, C., & Nithin, B. N. (2018). Smart Electronic Trolley for Shopping Mall. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. <https://doi.org/10.1109/rteict42901.2018.9012466>
2. Zemic weighing sensors integrated into the Easy Shopper shopping trolley. (n.d.). www.zemiceurope.com. Retrieved November 11, 2023, from
3. <https://www.zemiceurope.com/en/case-studies/Zemic-weighingsensors-easy-shopper>
4. Terrapass. (2022, January 18). *The Impact of World Energy Consumption and Solutions*. Terrapass. <https://terrapass.com/blog/impact-world-energy-consumption-and-solutions/>

5. *Smart shopping trolleys improve the customer journey* | Wanzl. (2021, October 21). [www.wanzl.com](http://www.wanzl.com/en_DE/wanzl-inside/press-and-news/smart-shopping-trolleys-improve-the-customer-journey~n5505). https://www.wanzl.com/en_DE/wanzl-inside/press-and-news/smart-shopping-trolleys-improve-the-customer-journey~n5505
6. Soori, P. K., Abraham, K. M., & Osman, M. A. M. A. I. (2021). Smart and Sustainable Shopping Cart for the Physically Challenged: International Conference on Intelligent Manufacturing and Energy Sustainability 2020. *Intelligent Manufacturing and Energy Sustainability*, 195–206. https://doi.org/10.1007/978-981-33-4443-3_19
7. Collins, T. (2023, April 7). *What is User Interface (UI)? (Types & Features)*. BrowserStack. <https://www.browserstack.com/guide/what-is-user-interface>
8. *Web page editor [2023]*. (n.d.). Mobirise.com. Retrieved January 6, 2024, from <https://mobirise.com/builder/web-page-editor.html>
9. Mechanical engineering department diploma of mechanical engineering session 2 2021/2022 *REPORT PROJECT SMART GROCERY CART PREPARED BY*. (n.d.). Retrieved December 8, 2023, from http://repository.psa.edu.my/bitstream/123456789/4273/1/SMART%20GROCERY%20CART%20LAPORAN%20AKHIR_2.pdf
10. Sarai, R. (2017, November 18). *Solar Panel Equipment | Solar Energy*. LA Solar Group. <https://la-solargroup.com/how-to-choose-the-right-solar-equipment/>
11. Mostafa, M. (2018, February 12). *What to consider before choosing a payment gateway for your business*. PayTabs. <https://site.paytabs.com/en/top-3-things-to-consider-while-choosing-a-payment-gateway-for-your-business/>
12. Bruijn, L. de. (2021, June 22). *Seven Tips To Clean Code With Python*. Analytics Vidhya. <https://medium.com/analytics-vidhya/seven-tips-to-clean-code-with-python-24930d35927f>
13. Editor, A. (n.d.). *Meet Tiny11, a fine-tuned Windows 11 build that barely needs RAM*. PCWorld. <https://www.pcworld.com/article/1505699/1505699.html>
14. *Laptop Offers – Buy Laptops of at Best Price in Oman*. (n.d.). Retrieved December 9, 2023, from <https://oman.sharafdg.com/c/computing/laptops/>
15. Clark, L., & Wright, P. (2007). Off Their Trolley — Understanding Online Grocery Shopping Behaviour. *Springer EBooks*, 157–170. https://doi.org/10.1007/978-0-387-73697-6_12
16. Urankar, D., Giri, A., Rajput, V., & Dhavalikar, A. (2022). Smart Shopping Trolley System. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4109035>
17. Alotaibi, S. (2012). Intelligent trolley. *Uobrep.openrepository.com*. <https://uobrep.openrepository.com/handle/10547/302716>
18. *An implementation of electronic shopping cart on the Web system using component-object technology / IEEE Conference Publication / IEEE Xplore*. (n.d.). ieeexplore.ieee.org. Retrieved January 6, 2024, from <https://ieeexplore.ieee.org/abstract/document/945116>
19. Lohse, G. L., & Spiller, P. (1998). Electronic shopping. *Communications of the ACM*, 41(7), 81–87. <https://doi.org/10.1145/278476.278491>
20. El Mahboul, A. (2014). *Smart shopping cart system : a new innovation for Grocery Industry*. www.theseus.fi. <https://urn.fi/URN:NBN:fi:amk-2014060612041>
21. *Boost Revenue With Smart Shopping Cart Conversion Strategies*. (n.d.). www.linkedin.com. Retrieved January 7, 2024, from <https://www.linkedin.com/pulse/boost-revenue-smart-shopping-cart-conversion-chris-krystalowich>

22. *Ahold Delhaize Q1 2023 Growth: Success in Local Brands*. (n.d.). www.grocerydoppio.com. <https://www.grocerydoppio.com/articles/grocers-smart-cart-elevating-the-customer-experience-and-driving-efficiency>
23. Danziger, P. N. (n.d.). *Smart Carts May Be The Disruptive Technology Grocery Stores Need Now*. Forbes. Retrieved January 7, 2024, from <https://www.forbes.com/sites/pamdanziger/2023/08/03/smart-carts-may-be-the-disruptive-technology-grocery-stores-need-now/?sh=4bd645ff4588>
24. *Cloud-based Centralized Smart Cart and Contactless Billing System | IEEE Conference Publication | IEEE Xplore*. (n.d.). ieeexplore.ieee.org. Retrieved January 7, 2024, from <https://ieeexplore.ieee.org/document/9835856>
25. *Smart Shopping Cart | IEEE Conference Publication | IEEE Xplore*. (n.d.). ieeexplore.ieee.org. Retrieved January 8, 2024, from <https://ieeexplore.ieee.org/abstract/document/8909680>
26. *Radio Frequency Identification Based E-Shopping Cart | IEEE Conference Publication | IEEE Xplore*. (n.d.). ieeexplore.ieee.org. Retrieved January 8, 2024, from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10266032>
27. *Implementation of Smart Shopping Carts for Large Retail Chain*. (n.d.). www.linkedin.com. Retrieved January 8, 2024, from <https://www.linkedin.com/pulse/implementation-smart-shopping-carts-large-retail-chain-mohindroo->
28. *Waterfall vs Agile models: SWOT analysis*. (2017, June 16). Welcome to Sidra's Blog. <https://sidraarooj.wordpress.com/2017/06/16/waterfall-vs-agile-models-swot-analysis/>
29. Dam, R. (2023). *5 stages in the design thinking process*. The Interaction Design Foundation. <https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process>
30. Lalband, N., & Kavitha. (2020, August). *Software Development Technique for the Betterment of End User Satisfaction using Agile Methodology* [Review of *Software Development Technique for the Betterment of End User Satisfaction using Agile Methodology*]. ResearchGate. https://www.researchgate.net/publication/344127540_Software_Development_Technique_for_the_Betterment_of_End_User_Satisfaction_using_Agile_Methodology
31. *How is Walmart revolutionizing its shopping carts?* (n.d.). <https://www.retail4growth.com/>. <https://www.retail4growth.com/viewpoints/how-is-walmart-revolutionizing-its-shopping-carts-582>
32. Alotaibi, S. (2012). *Intelligent trolley*. uobrep.openrepository.com. <https://uobrep.openrepository.com/handle/10547/302716>
33. Urankar, D., Giri, A., Rajput, V., & Dhavalikar, A. (2022, April 9). *Smart Shopping Trolley System*. Papers.ssrn.com. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4109035
34. Clark, L., & Wright, P. (2007). *Off Their Trolley — Understanding Online Grocery Shopping Behaviour*. *Springer EBooks*, 157–170. https://doi.org/10.1007/978-0-387-73697-6_12

All the codes:

```
// Include Libraries
```

```
#include "Arduino.h"
```

```
#include "Buzzer.h"
```

```
#include "HX711.h"
```

```
#include "Adafruit_STMPE610.h"
```

```
#include "Wire.h"
```

```
// Pin Definitions
#define BUZZER_PIN_SIG 2
#define SCALE_PIN_DAT 4
#define SCALE_PIN_CLK 3

// Global variables and defines

// object initialization
Buzzer buzzer(BUZZER_PIN_SIG);
HX711 scale(SCALE_PIN_DAT, SCALE_PIN_CLK);
#define calibration_factor 2280 //This value is obtained using the SparkFun_HX711_Calibration sketch
https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-
guide?_ga=2.77038550.2126325781.1526891300-303225217.1493631967
Adafruit_STMPE610 touch = Adafruit_STMPE610();

// define vars for testing menu
const int timeout = 10000; //define timeout of 10 sec
char menuOption = 0;
long time0;

// Setup the essentials for your circuit to work. It runs first every time your circuit is powered with
electricity.
void setup()
{
  // Setup Serial which is useful for debugging
  // Use the Serial Monitor to view printed messages
  Serial.begin(9600);
  while (!Serial) ; // wait for serial port to connect. Needed for native USB
  Serial.println("start");

  scale.set_scale(calibration_factor);
  scale.tare(); //Assuming there is no weight on the scale at start up, reset the scale to 0
  if (! touch.begin()) Serial.println("Touch controller not found!");
  menuOption = menu();
}

// Main logic of your circuit. It defines the interaction between the components you selected. After setup,
it runs over and over again, in an eternal loop.
void loop()
{
  if(menuOption == '1') {
```

```
// Buzzer - Test Code
// The buzzer will turn on and off for 500ms (0.5 sec)
buzzer.on(); // 1. turns on
delay(500); // 2. waits 500 milliseconds (0.5 sec). Change the value in the brackets (500) for a
longer or shorter delay in milliseconds.
buzzer.off(); // 3. turns off.
delay(500); // 4. waits 500 milliseconds (0.5 sec). Change the value in the brackets (500) for a
longer or shorter delay in milliseconds.
}
else if(menuOption == '2') {
// SparkFun HX711 - Load Cell Amplifier - Test Code
float scaleUnits = scale.get_units(); //scale.get_units() returns a float
Serial.print(scaleUnits); //You can change this to lbs but you'll need to refactor the calibration_factor
Serial.println(" Kg"); //You can change this to lbs but you'll need to refactor the calibration_factor
}
else if(menuOption == '3') {
// Resistive Touch screen - 3.7" Diagonal - Test Code
uint16_t x, y;
uint8_t z;
if (touch.touched()) {
// read x & y & z;
while (! touch.bufferEmpty()) {
Serial.print(touch.bufferSize());
touch.readData(&x, &y, &z);
Serial.print("->");
Serial.print(x); Serial.print(", ");
Serial.print(y); Serial.print(", ");
Serial.print(z);
Serial.println("");
}
touch.writeRegister8(STMPE_INT_STA, 0xFF); // reset all ints
}
delay(10);
}

if (millis() - time0 > timeout)
{
menuOption = menu();
}
}
// Menu function for selecting the components to be tested
// Follow serial monitor for instructions
```

```
char menu()
{

    Serial.println(F("\nWhich component would you like to test?"));
    Serial.println(F("(1) Buzzer"));
    Serial.println(F("(2) SparkFun HX711 - Load Cell Amplifier"));
    Serial.println(F("(3) Resistive Touch screen - 3.7\" Diagonal"));
    Serial.println(F("(menu) send anything else or press on board reset button\n"));
    while (!Serial.available());

    // Read data from serial monitor if received
    while (Serial.available())
    {
        char c = Serial.read();
        if (isAlphaNumeric(c))
        {

            if(c == '1')
                Serial.println(F("Now Testing Buzzer"));
            else if(c == '2')
                Serial.println(F("Now Testing SparkFun HX711 - Load Cell Amplifier"));
            else if(c == '3')
                Serial.println(F("Now Testing Resistive Touch screen - 3.7\" Diagonal"));

            else
            {
                Serial.println(F("illegal input!"));
                return 0;
            }
            time0 = millis();
            return c;
        }
    }
}

class TrolleySystem:
def __init__(self):
    self.user_logged_in = False
    self.trolley_locked = False

def start(self):
    print("Process started by user.")
    self.second_time()
```

```
def second_time(self):
    print("System acknowledges start.")
    self.account_log_in()

def account_log_in(self):
    self.user_logged_in = True
    print("User logged into account.")
    self.web_page()

def web_page(self):
    print("System displays web page for user.")
    self.start_shopping()

def start_shopping(self):
    print("User starts shopping.")
    self.move_trolley()

def move_trolley(self):
    if self.apply_force_on_handle():
        print("Trolley moves successfully.")
        self.add_product()
    else:
        self.lock_tires()

def apply_force_on_handle(self):
    # Simulate force application and movement detection
    force_applied = True # This can be a sensor input in real implementation
    return force_applied

def lock_tires(self):
    self.trolley_locked = True
    print("Trolley tires locked.")

def add_product(self):
    print("User adds product to trolley.")
    self.weight_sensor_detection()

def weight_sensor_detection(self):
    print("System detects added product using weight sensors.")
    self.scanning_items()
```

```
def scanning_items(self):
    print("System scans the items added to the trolley.")
    self.through_account()

def through_account(self):
    print("System processes scanned items through the user's account.")
    self.from_database()

def from_database(self):
    print("System retrieves product information from the database.")
    self.response_to_server()

def response_to_server(self):
    print("System sends processed information to the server.")
    self.response_to_touchscreen()

def response_to_touchscreen(self):
    print("System updates the touchscreen interface.")
    self.google_payment()

def google_payment(self):
    print("User makes a payment via Google Payment.")
    self.trolley_account()

def trolley_account(self):
    print("System updates trolley account with payment information.")
    print("Transaction complete.")

# Simulate the process
trolley_system = TrolleySystem()
trolley_system.start()
```