

# IntelliCart Project

## Overview

IntelliCart is a smart shopping cart system designed to transform the shopping experience by automating the checkout process and minimizing queue times. Using an RFID card, customers can start a shopping session with just a tap. As they add items to the cart, IntelliCart utilizes an integrated camera to detect and recognize products visually, which are then displayed on the mobile app in real time. This app also provides a live view of the cart contents, allowing customers to keep track of selected items, view product details, and manage their shopping list seamlessly. At checkout, the cart automatically calculates the total, making the payment process smooth and efficient.



## Appendix: Component Requirements

- Microcontrollers: Maixduino (K210), ESP32
- Modules: RC522 RFID, SD Card, OLED display
- Communication Protocols: UART (Maixduino → ESP32), SPI (ESP32 → RFID)
- Storage: SD Card (CSV data file for product prices)

## 1. System Design and Architecture

## System Flow

1. Session Start:
  - The customer scans the RFID card to begin the shopping session.
  - The cart is linked to the customer's profile.
2. Product Detection:
  - The Maixduino (K210) module identifies items added to the cart through image processing.
  - It sends the detected product's name to the ESP32 via UART.
3. Product Details Display:
  - ESP32 retrieves the product details (name and price) from an SD card.
  - It displays the information on an OLED screen and updates the cloud via Wi-Fi.
4. Real-Time Cart Monitoring:
  - The customer can view cart details, including products and prices, on their smartphone via a local HTML server hosted by ESP32.
5. Session End and Checkout:
  - Customer scans their RFID card again to finish shopping.
  - ESP32 calculates the total cost, displays it on the OLED, and sends it to the HTML server for viewing on the mobile app.
6. Payment:
  - Payment is completed at the checkout desk.

## 2. Hardware Integration

### Modules and Communication Protocols

1. Maixduino (K210) and ESP32 Communication:
  - Establishes UART communication for data transfer.
  - Maixduino detects and identifies products through image processing, then sends the product name to ESP32.
  - Data Format: Product name is sent in plain text (e.g., "item\_name").
2. RFID for Customer Identification:
  - RFID module (RC522) connects to ESP32 via SPI or I2C.
  - Customer scans their RFID to start/stop a shopping session.
  - Process:
    - First scan: starts a session and links the cart to the customer.
    - Second scan: ends the session and triggers total calculation.
3. SD Card for Product Data Storage:
  - Stores a CSV file with product names and prices for retrieval by ESP32.

- CSV Format Example:

Apple,1.50

Bread,2.00

Milk,1.20

#### 4. OLED Display:

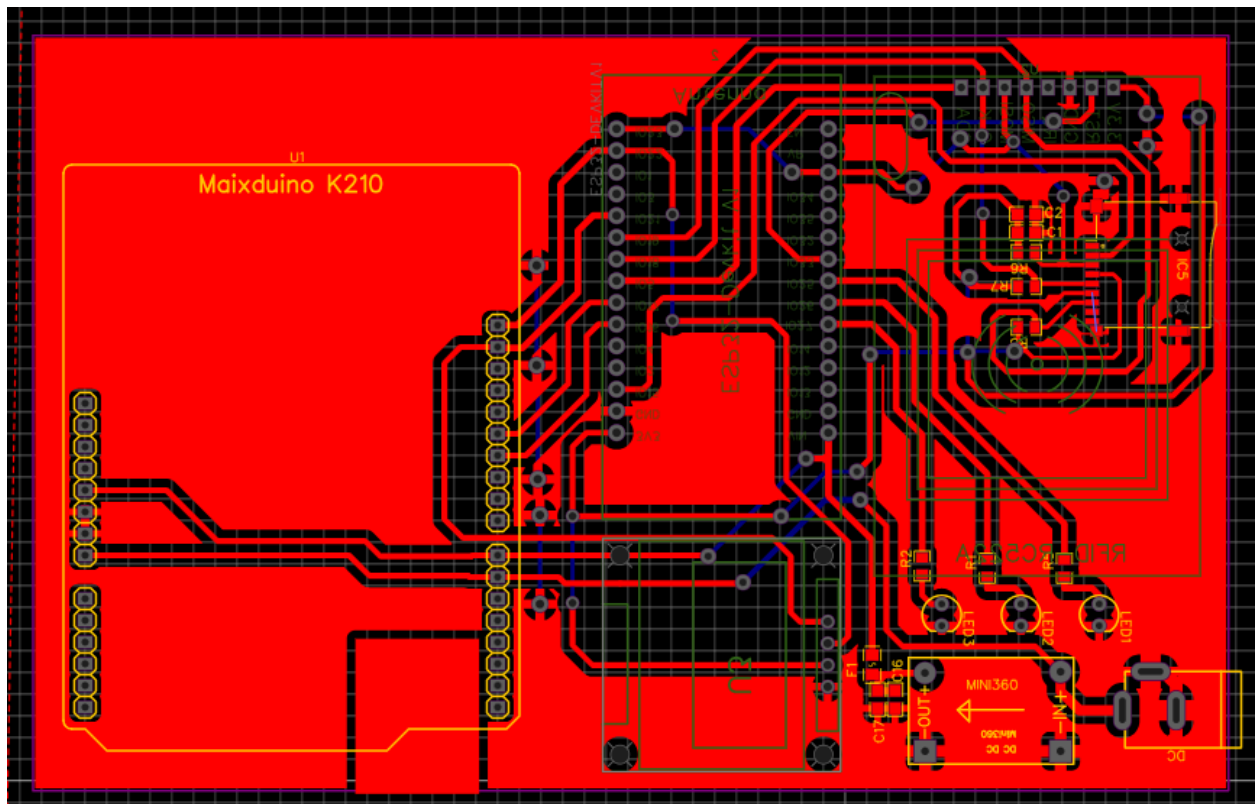
- Displays product details and total cost.
- Example Display:

Product: Bread Price: ₹20.00

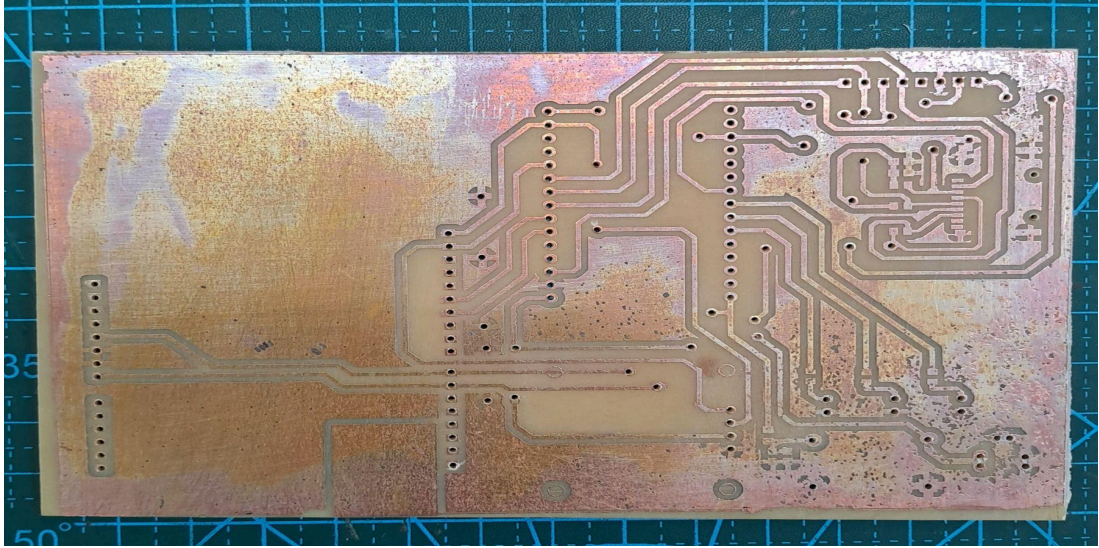
#### 5. HTML Server on ESP32:

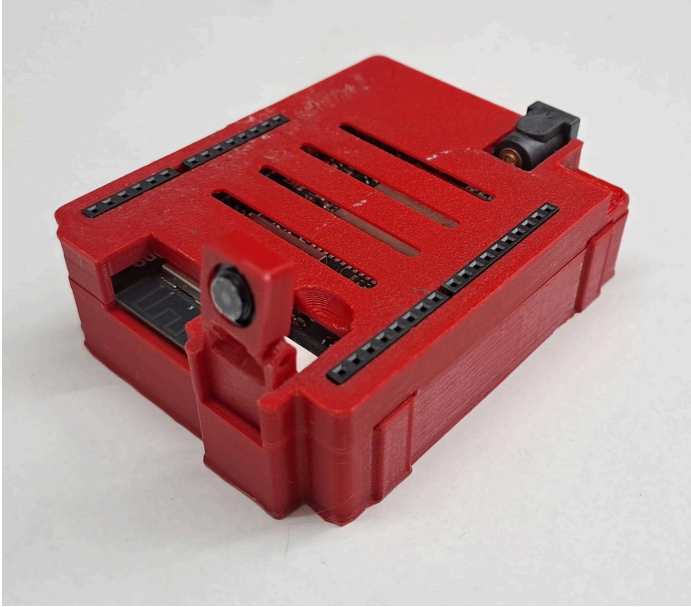
Hosts a local HTML server accessible on smartphones via the network (e.g., <http://192.168.x.x>).

- Displays:
  - Current list of products and prices.
  - Total amount due at checkout.









•

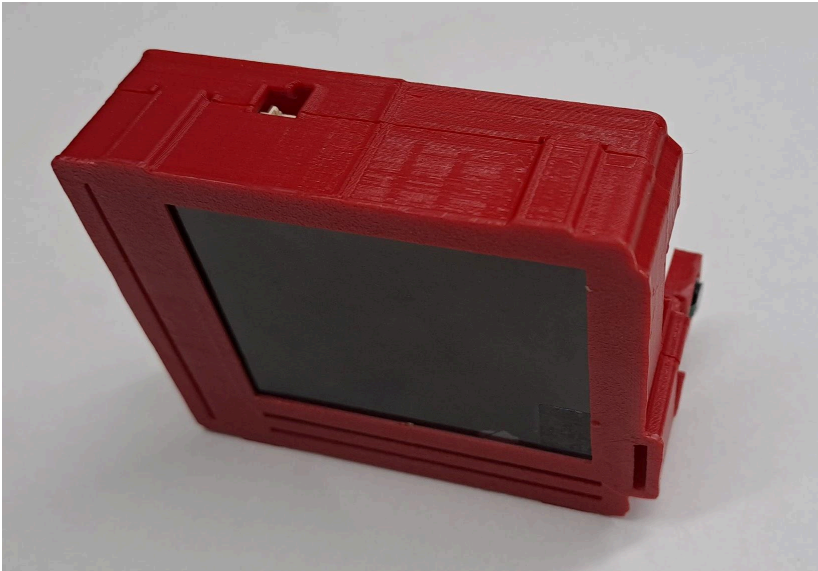


•





•



•

### 3. Software Development

For code you can checkout

GitHub:<https://github.com/Khushi4502/Intellicart>

#### ESP32 Code

- RFID Scanning (Customer Identification):
  - On scanning RFID, initiates or ends a session and stores customer information.

- UART Communication (Maixduino → ESP32):
  - Receives product name from Maixduino via UART.
  - Searches SD card for product price and displays it on the OLED.
- SD Card Lookup:
  - Reads product data from CSV file.
  - Retrieves the price for each item detected by Maixduino.
- Cloud and Local Server Integration:
  - Synchronizes cart data with the cloud and local HTML server.
  - Customers can view their shopping cart in real time.
- Checkout and Total Calculation:
  - Calculates total amount when the customer scans their RFID again to end the session.
  - Displays total on OLED and sends it to the local HTML server.

### HTML Interface

- Hosted by ESP32, the local HTML page displays:
  - Dynamic list of scanned products with prices.
  - Total amount due at checkout.

### Maixduino Code

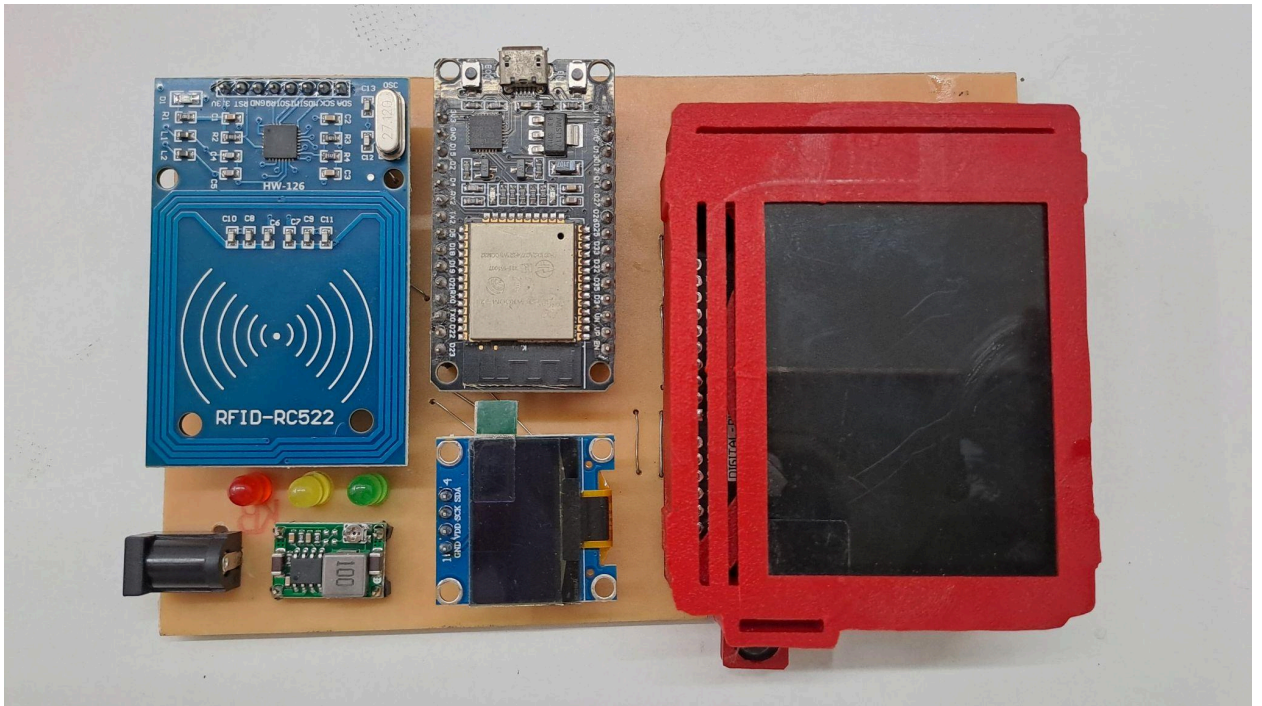
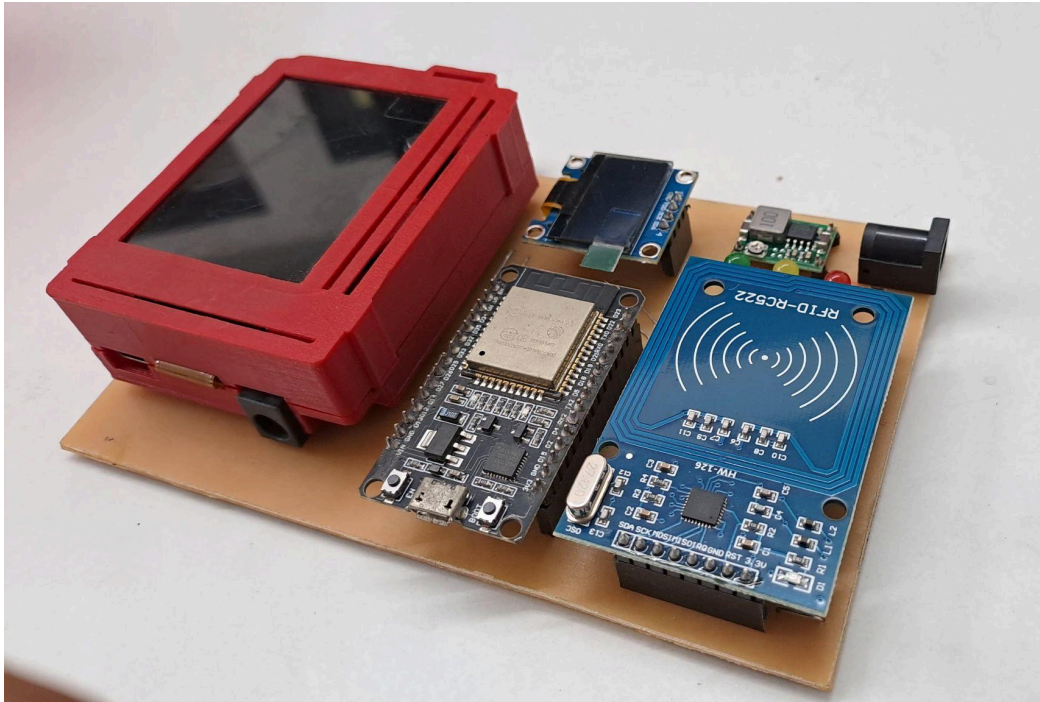
- Product Detection:
  - Identifies products through image processing.
  - Sends product name to ESP32 via UART in the format "product\_name\n".

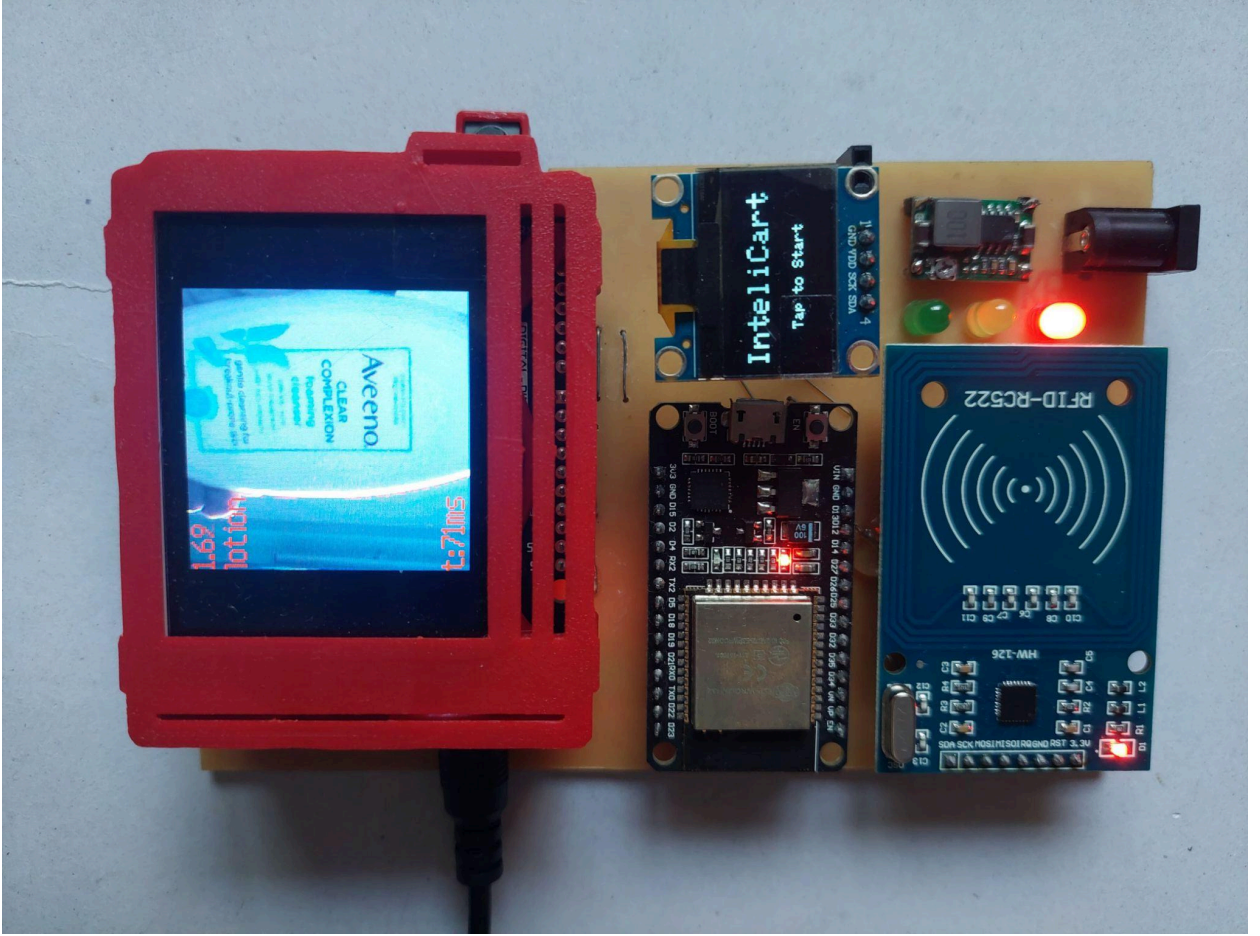
## 4. IntelliCart Hardware Design and Final Product Showcase

- **Schematic Design Overview**  
 This section includes detailed schematic diagrams of IntelliCart's hardware components, illustrating the connections and integrations between Maixduino, ESP32, RFID reader, SD card, OLED display, and any additional modules.









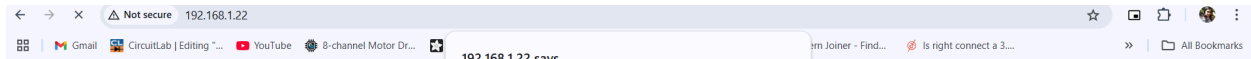


## Intellicart Shopping List

Number	Item Name	Quantity	Price	Value
1	envelope	7	5	35
2	lotion	2	110	220

Total Amount: Rs.255

Complete Shopping



Number	Item Name	Quantity	Price	Value
1	envelope	7	5	35
2	lotion	2	110	220

Total Amount: Rs.255

Complete Shopping

## 5. Testing & Validation

### Initial Testing

- RFID: Verify correct detection of customer ID and session initiation/termination.
- UART Communication: Ensure successful product name transfer from Maixduino to ESP32.
- SD Card Reading: Check accurate product-price retrieval from the SD card.
- OLED Display: Test display functionality for product and price details and checkout total.

### User Experience Testing

- Mobile App/HTML Page: Ensure customers can view their cart details in real time.
- Simulated Shopping Session: Test the workflow end-to-end, including:
  - RFID scanning.
  - Product identification and price lookup.
  - Real-time cart updates.
  - Checkout and total calculation.

# Project Workflow Summary

1. Customer RFID Scan:
  - Begins shopping session; cart links to customer profile.
2. Product Detection:
  - Maixduino identifies products and sends names to ESP32.
3. Data Processing on ESP32:
  - Retrieves prices from SD card.
  - Displays product info on OLED and updates the HTML server.
4. Real-Time Cart Updates:
  - Cart updates in real-time on the mobile HTML page.
5. Checkout:
  - Customers re-scan RFID to complete shopping.
  - ESP32 calculates the total and updates the OLED and HTML pages.

## Future Enhancements and Scalability

In future iterations, IntelliCart could incorporate advanced features to improve functionality, user engagement, and store integration. These updates are intended to make IntelliCart even more interactive, efficient, and competitive for modern retail environments.

### 1. Enhanced Customer Interaction

- **Real-Time Notifications:** In later versions, we can introduce real-time notifications sent to the customer's smartphone. These notifications would provide instant updates on products added, prices, and any available promotions.
- **Personalized Recommendations:** By integrating customer profile data, IntelliCart could offer tailored recommendations, like showing seasonal deals or popular items based on the customer's purchase history. This personalization would enhance the shopping experience and increase sales opportunities.
- **Interactive OLED Display with Additional Details:** Future updates to the OLED display could include additional product details, such as product images, nutrition facts, and suggested pairings. This would create a more interactive shopping experience directly on the cart.

### 2. Advanced Hardware Integration



- **Upgraded Image Processing on Maixduino:** To improve product recognition, we could use a more advanced model on Maixduino. This would enhance detection accuracy, especially in stores with diverse or similar-looking products. With a refined model, IntelliCart could handle complex product ranges with greater precision.

### 3. Enhanced Mobile and Web Integration

- **QR Code Display for Easy Mobile Access:** Displaying a QR code on the OLED screen for easy access to the HTML server is another valuable addition. By simply scanning the code, customers could instantly connect to the cart view on their smartphone, enhancing convenience and connectivity.
- **Dedicated Mobile App:** In the future, a dedicated mobile app could replace or complement the HTML interface. An app would allow for a smoother interface, saved shopping lists, product wishlists, and real-time push notifications. It could also provide loyalty points, promotions, and discounts tailored to the customer's preferences.

### 4. Security and Anti-Theft Measures

- **Verification at Exit:** Future versions could integrate a checkout confirmation QR code. Once the customer completes their checkout on IntelliCart, the OLED could display this code. Store staff would then scan it to verify the purchase, enhancing security and reducing the risk of accidental or intentional theft.
- **Tamper Detection and Alerts:** Another possible addition would be tamper detection mechanisms. If any discrepancies are detected (such as unmatched items between detected products and actual items), the system could alert store personnel immediately.