ISSN 2616-5902 Vol. 7, Issue 6: 52-56, DOI: 10.25236/AJBM.2025.070608

Shared Shops Based on an Intelligent Container System

Sili He[1,*], Xingyu Hui[1], Xiaoyu Shang[2]

[1]*Blockchain Engineering, Sanya University, Sanya, China*
[2]*Saxo Fintech Business School, Sanya University, Sanya, China*
**Corresponding author*

Abstract: *With the rapid development of the sharing economy and IoT technologies, traditional retail models face increasing demands for intelligent transformation. This paper proposes a shared store solution based on a smart cabinet system, integrating Android client development, RFID radio frequency identification, and IoT technologies to achieve unmanned commodity management and automated sales processes. The system adopts a layered architecture design supporting multi-role collaboration (customers, stock clerks, operators, and factory testers), incorporating functions such as product identification, remote hardware control, real-time inventory management, and video surveillance. Test results demonstrate that the system significantly improves cabinet operation efficiency, reduces labor costs, and addresses product misidentification through a discrepancy checking mechanism, achieving an inventory accuracy rate exceeding 98%. The project validates the feasibility of smart cabinets in shared retail scenarios, providing technical support for promoting small-scale distributed business models.*

Keywords: *Shared Store; Smart Cabinet; Android Client; Internet of Things (IoT)*

1. Introduction

Shared stores represent an innovative retail model enabled by IoT and smart terminals, aiming to reduce operational costs and enhance service efficiency through resource sharing and automated management. The core concept involves transforming traditional physical stores into a distributed network of smart cabinets that support 24/7 unmanned operation. Users complete product selection and payment via mobile scanning, while the system monitors inventory and environmental conditions in real time to achieve fully digitized end-to-end management.

Smart cabinets, as IoT-powered automated retail terminals, integrate RFID radio frequency identification, environmental sensing, and video surveillance technologies to create unmanned intelligent shopping scenarios. Their modular design allows flexible deployment in high-traffic areas such as communities and office buildings, with expandable functional units (e.g., refrigeration or heating modules) based on demand. By leveraging remote hardware control and intelligent scheduling systems, labor and maintenance costs are significantly reduced. Users complete purchases through contactless QR code scanning for door unlocking and automatic settlement. The system collects real-time sales data and behavioral insights to dynamically optimize product placement and marketing strategies. Integrated technical solutions, such as discrepancy checking mechanisms and adaptive temperature/humidity regulation, ensure operational safety and reliability at the hardware level, delivering efficient services to consumers while enabling merchants to achieve refined operational management.

2. Shared Store System Design Based on Smart Cabinet Technology

This system targets university scenarios, focusing on high-frequency essential goods for students (e.g., beverages, snacks, stationery) by deploying smart cabinets in dormitory buildings, teaching buildings, and libraries. It establishes a lightweight "space-sharing + long-term collaboration" business model. Cabinets are divided into fixed zones by product category (e.g., beverage zone, snack zone, stationery zone), with each zone leased long-term by different brands (e.g., a beverage zone operated annually by a tea chain brand). Merchants pay fixed shelf rental fees (e.g., 2,000 yuan/month/section) and 3%-5% sales commissions, eliminating the need for frequent product rotations or time-based bidding. The system analyzes student consumption patterns (e.g., 80% beverage sales in teaching building cabinets during lunch hours) to guide merchants in optimizing product selection (e.g., adding energy drinks during exam

ISSN 2616-5902 Vol. 7, Issue 6: 52-56, DOI: 10.25236/AJBM.2025.070608

weeks or restocking instant noodles late at night). In a dormitory pilot, the beverage zone was contracted by a milk tea brand, the snack zone partnered with a convenience store supply chain, and the stationery zone collaborated with campus cultural clubs. Single cabinets achieved stable daily orders exceeding 150 transactions, a 60% repurchase rate, and 40% lower merchant operational costs, validating the sustainability of the "fixed zoning + data-driven" model in university settings.

3. Core Technologies Overview

3.1. Android Platform Technologies

The Android platform is an open-source mobile operating system led by Google, designed based on the Linux kernel and widely used in smartphones, tablets, and other smart devices. Its open nature allows manufacturers and developers to customize the system freely. Since its release in 2007, Android has undergone continuous iterations, with Version 10 supporting 5G technology by 2020. Market data indicates that Android holds 87% of the global smartphone market share, making it the dominant mobile OS [1].

Architecturally, Android adopts a layered structure: the Linux kernel at the bottom layer manages hardware drivers, system runtime libraries provide core component support, the application framework layer offers SDK interfaces for developers, and the application layer implements functional applications. This layered design ensures system stability while improving development efficiency. Developers primarily use Java or Kotlin with the Android Studio IDE, built on the IntelliJ IDEA platform. The tool's rich plugin ecosystem significantly enhances development efficiency.

The Android client for the smart cabinet system leverages these core technologies combined with RFID integration. It fully utilizes Android's openness, layered architecture advantages, and mature development toolchain to ensure client stability and scalability [2].

3.2. RFID Technology Principles and System Architecture

RFID technology serves as the core identification method for the smart cabinet system, enabling non-contact data exchange via radio waves. RFID tags contain embedded chips and antennas that store unique product codes and rewritable information. When a tag enters the scanning range of a reader (typically within a few meters), radio frequency signals activate the tag to generate induced currents and transmit data back. The reader decodes the information through middleware and transmits it to an industrial PC for inventory updates and order generation. Compared to traditional barcode technology, RFID offers advantages such as penetration scanning and batch processing. Combined with a discrepancy checking mechanism (enhanced signal revalidation), it effectively resolves misread issues. The reprogrammable nature of RFID tags significantly reduces maintenance costs, providing a cost-effective solution for unmanned retail. The hardware system comprises tags, readers (including RF modules and power circuits), and middleware, forming a closed-loop process from data acquisition to business processing. This achieves second-level inventory checks and over 99% recognition accuracy in smart cabinets [3][4].

4. Requirements Analysis

4.1. Role **Analysis** *and Requirements Overview*

4.1.1. The Smart Cabinet System Primarily Serves Four User Groups

Customers complete the entire shopping process, including door unlocking, item retrieval, and automatic payment, via QR code scanning.

Stock clerks manage restocking, inventory control, and hardware status monitoring (e.g., temperature and lighting adjustments).

Operators analyze customer behavior through video surveillance to optimize anti-loss algorithms.

Factory testers utilize self-check functions to perform hardware diagnostics on assembled cabinets, ensuring proper device operation.

4.1.2. Shopping Process Analysis

When a user scans a QR code via a WeChat Mini Program, the system first verifies the cabinet's

ISSN 2616-5902 Vol. 7, Issue 6: 52-56, DOI: 10.25236/AJBM.2025.070608

online status. The server sends an unlock command to the industrial PC. After the user retrieves items and closes the door, an automatic locking mechanism is triggered. The RFID system then performs a product inventory check and uploads data to the server for order generation. The process includes multiple exception handling steps, such as offline cabinet alerts and unlock failure feedback, to ensure a seamless shopping experience.

4.1.3. Requirements Overview

Functional requirements cover six core modules: IoT communication, command control, RFID inventory checks, hardware device control (door locks, lighting, compressors), self-diagnostics, and video surveillance. The hardware control module must support remote adjustments, while the video surveillance module requires behavioral analysis capabilities. Non-functional requirements emphasize system security, operational reliability, modular design for maintainability, and scalability. The system employs layered architecture to enable coordinated operation of all functional modules.

4.2. Functional Requirements

The system's functional requirements focus on IoT communication, hardware control, and anomaly monitoring. A persistent Socket connection ensures real-time interaction with the server, including automatic reconnection upon disconnection. Hardware status snapshots (door lock status, temperature, RFID device status) are reported every minute, while network/IP/battery information is synchronized every five minutes. RFID tag data is uploaded after each product scan for server-side order generation.

For hardware management, the system supports atomic commands (e.g., unlock, turn on lights) and composite command sets (e.g., "unlock-inventory-lock" sequences). Door lock voltage detection enhances security, while environmental control modules manage lighting and compressor operations (maintaining a 0-10°C refrigeration environment). RFID technology enables three inventory modes:

Routine inventory checks: Full scans are conducted every five minutes.

Restocking inventory updates: Inventory synchronization after restocking.

Discrepancy checks: Enhanced signal revalidation for suspected unretrieved items.

To ensure reliability, the system includes built-in hardware self-test functions. Before deployment, it automatically checks the status of door locks, RFID recognition, compressors, and fans, generating fault reports. The video surveillance module captures photos before and after door opening, records the entire shopping process, compresses files for storage, and periodically deletes outdated footage. It also supports server-side retrieval of videos for manual review of anomalies (e.g., label tampering), forming a closed-loop risk control system from data collection to decision-making.

5. Key Process Designs

5.1. Shopping Process Design

A user initiates shopping by scanning a QR code:

The server verifies the cabinet's online status and issues an unlock command.

The client pauses other hardware operations to execute unlocking.

After the user retrieves items and closes the door, the client automatically locks the cabinet and initiates a restocking inventory check.

The server triggers a discrepancy check if needed, prompting the client to perform enhanced RFID scans on specific tags to confirm item status (e.g., whether items were taken).

Final results are reported to the server for order generation, ensuring accurate inventory updates and order reliability.

5.2. Persistent Connection Service Design

During system startup, the client reads the device's MAC address from configuration files as a unique identifier. It resolves the server's IP address via DNS and establishes a Socket connection. SSL certificate validation ensures secure communication. If any step fails (e.g., network interruption or certificate error),

ISSN 2616-5902 Vol. 7, Issue 6: 52-56, DOI: 10.25236/AJBM.2025.070608

the client automatically triggers reconnection attempts until a stable link is established. This design guarantees real-time synchronization of commands and data between the server and cabinets.

5.3. Command Control Process Design

Command execution follows strict priority rules:

If the system is uninitialized, hardware initialization takes precedence.

If a command is already executing, the current task is interrupted to prioritize new commands (e.g., emergency door closure or temperature adjustments).

During execution, the client provides real-time feedback on command status (success/failure) to the server, ensuring traceability of the command chain. This mechanism applies to critical operations such as door lock control and compressor management.

5.4. Video Surveillance Process Design

When a user opens the cabinet door, the client first checks storage space. If the video folder exceeds capacity, the oldest files are deleted to free space. The system then captures two photos (before and after door opening) to document initial and final product states, while continuously recording the entire shopping process. After the door closes, recording stops, and the video is compressed and stored on the industrial PC. The system periodically purges old videos and allows servers to retrieve files on demand for manual review of suspicious activities (e.g., label removal), supporting compensation decisions.

6. System Testing

6.1. Test Environment

The test environment for the smart cabinet system is configured as shown in Table 1:

Table 1: System Test Environment.

Master control	RK3288 Industrial computer
Types of shelves	Metal shelves
Reader type	ultrahigh frequency
Container capacity	278L
Reader brand	Self-developed
Antenna brand	Self-developed
gate lock	KOBL
cellular network	4G
WiFi	exist
camera	exist
Android Version	Android 5.0

6.2. White-Box Testing

6.2.1. Complete Shopping Process

When a user triggers the shopping process via mobile scanning, the system first verifies the cabinet's online status. If online, the server sends an unlock command to the industrial PC; if offline, the user is notified of the failure. Upon receiving the command, the industrial PC pauses other hardware tasks and executes unlocking. After the user retrieves items and closes the door, an automatic locking mechanism is triggered, followed by a full RFID inventory check. Product change data is reported to the server for order generation. Exception scenarios (e.g., door unlock failure, network interruption) are communicated to users via real-time feedback, ensuring process transparency. Tests confirmed that both normal and post-self-test shopping paths accurately complete order processing with zero inventory synchronization errors [5][6].

6.2.2. Persistent Connection Service Communication

At system startup, the client reads the device's unique identifier (MAC address) from configuration

ISSN 2616-5902 Vol. 7, Issue 6: 52-56, DOI: 10.25236/AJBM.2025.070608

files, resolves the server's IP via DNS, and establishes a Socket-based persistent connection. During connection, bidirectional SSL certificate verification is performed. If network fluctuations or certificate validation failures occur, an automatic retry mechanism is triggered until successful connection. Tests demonstrated that the module recovers communication within 10 seconds under unstable network conditions (e.g., frequent disconnections) and terminates connections in case of configuration file errors to prevent resource waste, achieving overall communication stability exceeding 95%.

6.2.3. Command Priority Control Logic

The command control module adopts a dynamic priority scheduling strategy. Upon receiving a new command, hardware initialization is prioritized if the device is uninitialized. If a command is executing (e.g., temperature adjustment), it is immediately interrupted to respond to the new command (e.g., emergency door closure). Test cases covered multiple scenarios, including device initialization and command conflicts. Results showed the system completes command switching within 200 milliseconds and synchronizes real-time status to the server, ensuring command chain integrity and traceability.

6.2.4. Video Surveillance and Storage Management

When a user opens the door, the system first checks the industrial PC's storage space. If remaining space falls below a threshold (e.g., 10 GB), the oldest video files are deleted to free capacity. The system then captures two photos (pre- and post-unlock) and records the entire shopping process. After door closure, videos are compressed and stored locally, while the server can retrieve files on demand for anomaly reviews (e.g., label removal). Tests confirmed the module prioritizes retaining data from the past 24 hours under low storage, achieves over 60% compression rates, and maintains stable video recording at 25+ fps, meeting behavioral analysis requirements.

6.2.5. Test Result Integration

Module-specific control flow designs and use case validations confirmed system robustness in complex scenarios. For example, the command control module showed no command loss during concurrent operations, the video module-maintained functionality under extreme storage pressure, and the persistent connection service remained highly available in weak network conditions. The overall test pass rate reached 100%, with key metrics (e.g., inventory accuracy, command response latency) exceeding design requirements, ensuring reliability for commercial deployment.

7. Conclusions

This paper presents a smart cabinet shared store system based on IoT, RFID, and Android technologies. Through layered architecture and modular design, the system achieves automated product identification, remote control, and multi-role collaborative management, effectively improving operational efficiency and reducing labor costs. Tests demonstrated 98% inventory accuracy, command response latency below 200 milliseconds, and significant error mitigation via the discrepancy checking mechanism. The project explores diversified revenue models through B2B2C partnerships, offering a replicable solution for distributed retail. Future work will enhance AI algorithms for user behavior analysis, integrate supply chain digitization, and build an end-to-end smart retail ecosystem. This research provides technical support for traditional industries' intelligent transformation, delivering both social value and market potential.

References

[1] IDC new report: Android phones accounted for 87% of the market share in 2019, iPhone only 13%[OL]. https://www.expreview.com/70381.htm, 2019-09-10.
[2] Bi Xiaopeng. Master of Android Studio[M]. Beijing: Tsinghua University Press, 2017.
[3] Wang Ansheng, Software Engineering [M]. Beijing: Tsinghua University Press, 2014.
[4] Ma Biao. Design of Smart Home System Based on ARM and RFID [D]. Anhui University of Science and Technology, 2018.
[5] Black boxtesting and white box testing [OL]. https://blog.csdn.net/ruthywei/article/details/80484685
[6] Guo Qian. A Brief Discussion on the Application of White Box Testing Technology in Dynamic Software Testing [J]. Mobile Information,2016 (6):00152-00153.