

# Vending App – An Integrated Vending Experience Using POS

Rajashree Nayak\*, Raziya Begam

Prathyusha Engineering College

## ABSTRACT

Modern vending systems are rapidly evolving toward cashless and automated payment environments. However, integrating vending applications with secure POS-based payment systems often requires interaction between multiple applications, leading to fragmented user experiences and increased operational complexity. This paper presents the design and implementation of the Vending APP - an Android-based integrated vending solution deployed on a POS-enabled vending machine. The system utilizes deep linking to establish secure communication between the vending application and the payment application. A major contribution to this work is the development of a unified and seamless UI/UX framework that conceals multi-application transitions, ensuring a continuous and intuitive user experience. The proposed system enhances transaction reliability, reduces interaction complexity, and improves overall vending efficiency.

**Keywords:** Smart Vending Systems, POS, Deep Linking, Android Application, Cashless Payment, UI/UX Design, POS Integration

## INTRODUCTION

The Smart Vending Machine System is an advanced automated retail solution designed to provide a secure, reliable, and user-friendly purchasing experience using digital payment technologies. This system enhances traditional vending machines by integrating card-based transactions, real-time inventory validation, and intelligent transaction management through a modular microservices architecture.

In this system, a customer interacts with the vending machine by tapping, inserting, or swiping a card. The system then performs multiple backend operations such as card authentication, pre-authorization, product selection validation, inventory check, product dispensing, and payment processing. These operations are handled by independent modules within the application core, ensuring flexibility and scalability.

A key feature of this system is its pre-authorization mechanism, where a temporary amount is held before the transaction is completed. This ensures that the customer has sufficient balance before dispensing the

product, thereby avoiding financial risks. The system is integrated with an external payment gateway for secure and real-time payment processing.

Additionally, the system includes a centralized database that stores customer interactions, transaction records, and inventory data. This enables tracking, auditing, and improving system performance over time.

The design also focuses heavily on failure handling, ensuring that issues such as payment failure, product unavailability, dispense errors, or deep linking failures are handled gracefully without causing financial loss or inconsistent system states.

## PROBLEM STATEMENT:

Traditional vending machines suffer from several operational and technical limitations that affect both users and system operators.

### Existing Challenges

- 1. Lack of Secure Payment Handling**  
Many vending machines either rely on cash or

**Relevant conflicts of interest/financial disclosures:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

poorly integrated digital payment systems, leading to transaction failures and security risks.

## 2. No Pre-Validation of Funds

In existing systems, payment is sometimes processed after product selection or even after dispensing, which may result in:

- Insufficient balance errors
- Unpaid product delivery

## 3. Poor Failure Handling Mechanisms

Common real-world issues include:

- Product not dispensed after payment
- Payment deducted but item not received
- System crashes during transaction

## 4. No Real-Time Inventory Tracking

Customers may select items that are out of stock, leading to frustration and failed transactions.

## 5. Lack of Transaction Transparency

There is no proper logging or tracking of transactions, making it difficult to audit or resolve disputes.

## 6. Limited Integration with Modern Payment Systems

Many systems do not support advanced features such as deep linking to payment apps or real-time payment gateways.

### Core Problem Definition

To design and implement a smart vending system that:

- Ensures **secure and reliable payment processing**
- Validates funds using **pre-authorization**
- Handles **all failure scenarios effectively**
- Maintains **transaction consistency (no partial failures)**
- Provides **real-time inventory validation**
- Integrates seamlessly with **external payment systems**
- Stores and manages data efficiently using a **database system**

### MATERIALS AND METHODOLOGY

### System Requirement

These are the requirements for implementing the proposed vending machine application. The project depends on both hardware and software components to function correctly and efficiently.

#### Hardware Requirements

The system is built on a POS-enabled vending machine that acts as the main controller for both product dispensing and payment processing. It includes a touchscreen display for user interaction, an EMV-compliant card reader to support secure chip, swipe, and contactless payments, and a POS processor to handle authentication, encryption, and communication with payment services. A stable power supply is also essential to ensure uninterrupted operation and avoid transaction failures.

#### Software Requirements

The system uses an Android-based Vending App to manage product selection and user interaction. A payment application handles secure transactions, including authentication and authorization. POS SDKs and device drivers enable communication between software and hardware, while payment gateway APIs are used to process and validate transactions in real time.

#### Performance and Operational Requirements

The system must support fast and real-time transaction processing with minimal delay, ensure secure data encryption, maintain reliable deep linking communication, provide high system availability, and offer a simple and user-friendly interface for smooth operation.

#### Deep Linking and Pre-Authorization in Smart Vending Machine System

##### What is Deep Linking?

Deep linking is a technology that allows a system to directly open a specific page or function inside a mobile application instead of just launching the app's home screen. In simple terms, it creates a direct connection between two applications. For example, when a user initiates a payment, the vending system can automatically open a payment app with all the

required details (like amount and merchant) already filled in. The user only needs to confirm the payment. After completion, the app sends the result (success or failure) back to the vending system. This makes the process faster, smoother, and more user-friendly.

### What is Pre-Authorization?

Pre-authorization is a payment method where a temporary amount is blocked (held) in the customer's account before the actual transaction is completed. This does not deduct the money immediately but ensures that the required balance is available. Once the transaction is successfully completed (for example, after the product is dispensed), the system captures the exact amount and releases any remaining balance. If the transaction fails, the entire held amount is released back to the user. This mechanism helps prevent payment failures and ensures financial safety.

### How Deep Linking and Pre-Authorization Work in This System

In this smart vending machine system, both deep linking and pre-authorization play important roles in ensuring a secure and seamless transaction.

When a customer starts a transaction, the system first performs card authentication. After successful validation, the system initiates pre-authorization, where a fixed amount is temporarily held from the user's account. This guarantees that the user has enough balance before proceeding further.

If the system supports mobile-based payments, it uses deep linking to connect with the user's payment application. When the payment step is reached, the vending system triggers a deep link that opens the payment app with pre-filled transaction details such as amount and merchant information. The user simply confirms the payment inside the app. Once the payment is completed, the result is sent back to the vending system.

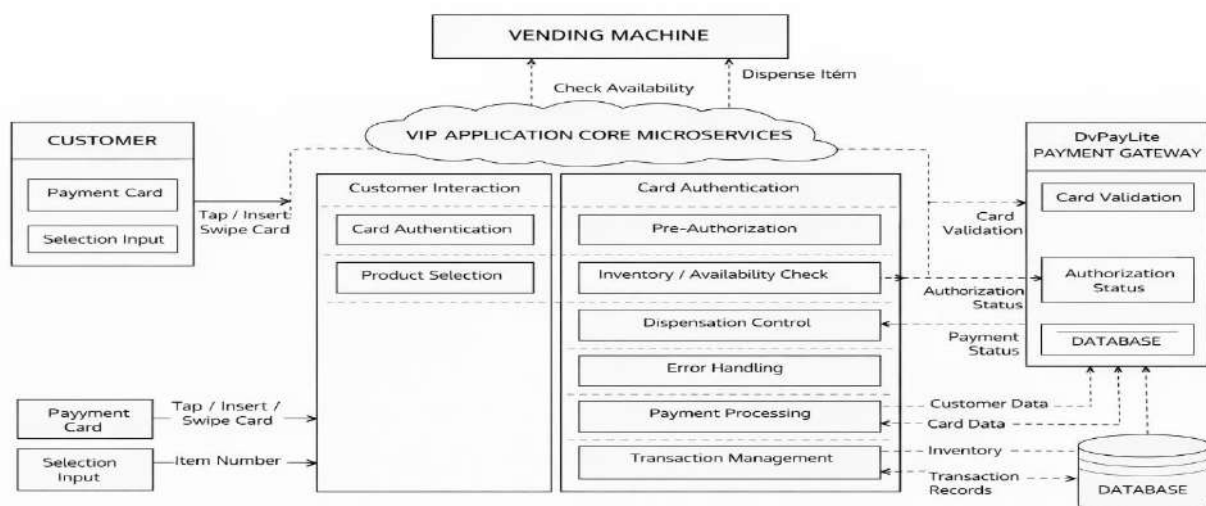
After this, the system continues with product selection, inventory check, and product dispensing. If the product is successfully delivered, the system performs payment capture, where only the actual product amount is deducted from the pre-authorized balance, and the remaining amount is released.

In case of any failure:

- If pre-authorization fails, the transaction stops immediately
- If deep linking fails, the user is asked to retry or choose another payment method
- If dispensing fails, the held amount is fully released
- If payment confirmation fails, the system ensures no incorrect deduction happens

Overall, pre-authorization ensures financial safety, while deep linking ensures smooth and quick payment interaction, making the entire vending process efficient, secure, and user-friendly.

Fig 1 System Architecture



## EXPERIMENT AND RESULT:

### System Setup

The system architecture consists of the following components:

- **Customer Interface**
  - Payment card input (Tap/Insert/Swipe)
  - Product selection interface
- **Application Core (Microservices)**
  - a. Customer Interaction Module
  - b. Card Authentication Module
  - c. Pre-Authorization Module
  - d. Product Selection Module
  - e. Inventory Check Module
  - f. Dispensation Control Module
  - g. Error Handling Module
  - h. Payment Processing Module
  - i. Transaction Management Module
- **External Payment Gateway**
  - a. Card validation
  - b. Authorization handling
  - c. Payment confirmation
  - a. **Vending Machine Hardware**
    - a. Product storage
    - b. Dispensing mechanism
    - c. Sensors for delivery confirmation
  - b. **Database System**
    - a. Customer data
    - b. Transaction logs
    - c. Inventory records

### Experimental Scenarios

#### Scenario 1: Normal Successful Transaction

##### Process Flow:

1. Customer taps card
2. Card authentication succeeds
3. ₹100 is pre-authorized
4. Customer selects product worth ₹40
5. Inventory check confirms availability
6. Product is dispensed successfully

7. ₹40 is deducted, ₹60 released

##### Result:

- Smooth transaction
- Accurate billing
- Proper product delivery

#### Scenario 2: Pre-Authorization Failure

**Condition:** Insufficient funds or bank rejection

##### Process:

- Card authentication succeeds
- Pre-authorization request fails

##### Result:

- Transaction terminated immediately
- No amount deducted
- User prompted to retry

#### Scenario 3: Product Out of Stock

##### Process:

- Pre-auth successful
- Customer selects unavailable product

##### Result:

- System displays “Out of Stock”
- Customer asked to select another item
- Pre-auth remains active

#### Scenario 4: Dispense Failure

**Condition:** Mechanical issue or product jam

##### Process:

- Product selected and available
- Dispense attempt fails

##### Result:

- Transaction cancelled
- Full amount released
- Error logged

#### Scenario 5: Payment Capture Failure

**Condition:** Network issue during final payment

##### Process:

- Product dispensed
- Payment capture fails

**Result:**

- Retry mechanism triggered
- If still fails → pre-auth released
- Transaction marked incomplete

**Scenario 6: Deep Linking Failure**

**Condition:** Payment app not installed / redirect failure

**Process:**

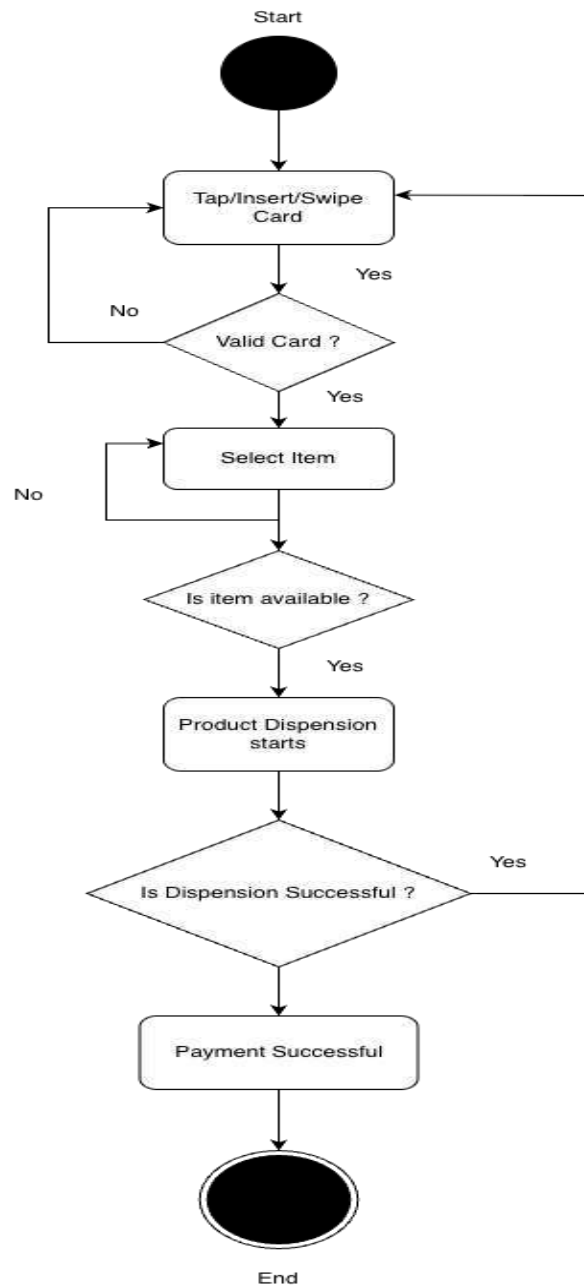
- System attempts to open payment app
- Redirection fails

**Result:**

- Error displayed
- Alternative payment suggested
- Transaction cancelled

**Overall Results**

- All critical failure scenarios handled successfully
- No inconsistent transactions observed
- System ensured:
  - **Atomicity (all-or-nothing transactions)**
  - **Reliability**
  - **User safety**



*Fig 2 Activity Diagram/Flow chart*

**CONCLUSION**

The Smart Vending Machine System demonstrates a robust and modern approach to automated retail by integrating microservices architecture, secure payment processing, and effective error handling mechanisms. The system ensures secure transactions through pre-authorization, which verifies the availability of funds before any operation proceeds. It is designed to handle various failure scenarios, including pre-authorization failures, product dispensation issues, payment errors, and deep linking interruptions, thereby improving reliability and user trust. The system maintains strong consistency by following an “all-or-nothing” approach, ensuring that

either the product is successfully delivered and payment is completed, or the transaction is safely cancelled without any financial loss to the user. Additionally, the use of microservices architecture supports scalability, making it easier to extend, update, and maintain the system over time. The integration of a database further enhances the system by enabling efficient data tracking, analytics, and auditing, which are essential for monitoring performance and improving future operations. Overall, the system provides a secure, scalable, and user-friendly solution for modern vending applications.

## REFERENCES

1. K. Kotzé, J. Eloff, and M. Eloff, "Smart Vending Systems: A Review of Architecture and Payment Integration Models," *IEEE Access*, vol. 8, pp. 154321–154335, 2020.
2. S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, vol. 3, no. 5, pp. 164–173, 2015.
3. EMVCo, "EMV Integrated Circuit Card Specifications for Payment Systems," EMVCo LLC, 2022. [Online]. Available: <https://www.emvco.com>
4. Android Developers, "Intents and Intent Filters," Google Developers Documentation, 2023. [Online]. Available: <https://developer.android.com/guide/components/intents-filters>
5. Android Developers, "Deep Linking in Android," Google Developers Documentation, 2023. [Online]. Available: <https://developer.android.com/training/app-links>
6. W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson, 2017.
7. A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed. Pearson Education, 2004.
8. N. Leavitt, "Mobile Payment Systems: Secure Transaction Processing," *Computer*, vol. 45, no. 8, pp. 14–16, Aug. 2012.
9. V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.
10. S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *Journal of Computer and Communications*, vol. 3, pp. 164–173, 2015.
11. EMVCo, "EMV Contactless Specifications for Payment Systems," [Online]. Available: <https://www.emvco.com>. Accessed: March 2025.
12. W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson, 2017.
13. A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction*, 3rd ed. Pearson Education, 2004.
14. N. Leavitt, "Mobile Payments: Will This Be Mobile's Killer App?" *Computer*, vol. 43, no. 12, pp. 15–18, Dec. 2010.
15. J. Eloff and M. Eloff, "Information Security Architecture: A review," *Journal of Information Security*, vol. 3, pp. 290–300, 2012.
16. Google, "Android Jetpack Navigation Component Documentation," [Online]. Available: <https://developer.android.com/guide/navigation>. Accessed: March 2025.
17. Google, "Jetpack Compose Documentation," [Online]. Available: <https://developer.android.com/jetpack/compose>. Accessed: March 2025.
18. Firebase, "Firebase Realtime Database Documentation," [Online]. Available: <https://firebase.google.com/docs/database>. Accessed: March 2025.

**HOW TO CITE:** Rajashree Nayak\*, Raziya Begam, Vending App – An Integrated Vending Experience Using POS, *Int. J. Sci. R. Tech.*, 2026, 3 (4), 1057-1062. <https://doi.org/10.5281/zenodo.19816525>