KONGU ENGINEERING COLLEGE

(Autonomous)

Perundurai, Erode – 638 060

DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION ENGINEERING

Estd : 1984

Transform Yourself

Supermarket Application System

AN MICRO PROJECT REPORT

for

JAVA PROGRAMMING (22ITC31)

Submitted by

Haries Ragavendra S (23EIR037)

Harini M (23EIR039)

Harshini Velmuruhan (23EIR040)

KONGU ENGINEERING COLLEGE

(Autonomous)

Perundurai, Erode – 638 060

DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION ENGINEERING

Estd : 1984

Transform Yourself

BONAFIDE CERTIFICATE

Name : **Haries Ragavendra S (23EIR037)**

Harini M (23EIR039)

Harshini Velmuruhan (23EIR040)

Course Code : **22ITC31**

Course Name : **JAVA PROGRAMMING**

Semester : **III**

Certified that this is a bonafide record of work for application project done by the above students for **22ITC31 – JAVA PROGRAMMING** during the academic year **2024 - 2025.**

Submitted for the Viva Voce Examination held on _____

Faculty In-Charge **Head of the Department**

INDEX

ABSTRACT

The growth of technology has transformed the shopping experience. Modern supermarket systems aim to provide efficient solutions to customer management, billing, and stock handling. This project focuses on designing a Supermarket Application System to manage customer records, offer personalized discounts, process payments, and streamline product inventory handling.

The methodology combines customer relationship management with an interactive system for shopping and billing. By incorporating product lists, validation checks, and secure payment options, this application ensures convenience and accuracy. It also integrates decision points for customer-specific discounts and personalized recommendations.

The proposed system demonstrated efficiency in managing transactions, including cash/card payments, and improved customer satisfaction by ensuring streamlined service. Future enhancements could include integrating AI-based recommendations and dynamic discount calculations based on shopping history.

PROBLEM STATEMENT

Efficient customer management is a significant challenge faced by large supermarkets in the modern retail landscape. With increasing competition and growing customer expectations, supermarkets must adopt efficient systems to maintain seamless operations while offering personalized services. Traditional approaches to customer and inventory management have proven to be inadequate in meeting these evolving demands, resulting in inefficiencies that affect both customer satisfaction and business profitability.

The retail industry, particularly supermarkets, relies on smooth and efficient operations to serve customers effectively. Over time, these establishments have expanded in scale, with more products, larger customer bases, and a growing reliance on technology. However,

the complexities of managing such operations have also increased. Customers today expect swift and accurate service, personalized offers, and seamless payment processes. Meeting these expectations requires advanced systems capable of handling diverse functions such as customer data management, inventory tracking, discount calculations, and payment processing.

In the past, customer management in supermarkets was a manual process involving physical ledgers or simple spreadsheets. This approach was labor-intensive and prone to human error, especially as the volume of customers increased. Inventory was tracked using manual logs, which often led to stock discrepancies and mismatches between customer demand and product availability. Similarly, discounts and offers were applied uniformly without considering individual customer preferences, reducing the overall appeal of loyalty programs. These inefficiencies not only affected customer satisfaction but also impacted the profitability and reputation of supermarkets.

The introduction of computerized systems brought some relief, allowing supermarkets to store and retrieve customer and inventory data more efficiently. Early systems helped streamline billing processes and reduce human error in basic operations. However, they lacked the ability to integrate complex functions such as dynamic discounting, real-time inventory updates, and secure payment validation. These limitations became increasingly evident as the scale and complexity of supermarket operations grew.

Despite advancements in technology, modern supermarket systems still face notable challenges. While most supermarkets have adopted digital systems, these are often fragmented, with customer data, inventory management, and payment systems operating independently. This lack of integration makes it difficult to offer a cohesive shopping experience. For instance, customer loyalty programs are frequently based on static rules rather than being tailored to individual preferences or shopping behavior.

Inventory management, a critical component of supermarket operations, also faces significant hurdles. Although digital systems

have replaced manual logs, discrepancies in stock levels remain a common issue. Customers often encounter situations where the desired product is unavailable or incorrectly matched with similar items from other brands. In such cases, existing systems rarely offer the flexibility to substitute products or provide customers with clear options for decision-making. This can lead to dissatisfaction and a loss of trust in the supermarket's ability to meet customer needs.

Payment processing is another area where gaps persist. With the widespread use of digital payment methods such as credit and debit cards, secure and efficient payment validation is essential. However, many systems do not integrate real-time bank balance checks, leading to declined transactions at the checkout counter. This not only inconveniences customers but also disrupts the flow of operations, creating delays for other shoppers. Moreover, the absence of a unified payment validation system increases the risk of fraudulent transactions, further complicating the payment process.

The need for advanced systems that address these challenges has become more urgent in recent years. Supermarkets are under pressure to differentiate themselves from competitors by offering superior customer experiences. Achieving this requires an integrated approach that combines efficient customer management, accurate inventory tracking, personalized discounting, and seamless payment processing. Existing systems, while functional to some extent, fall short of delivering these capabilities in a unified and scalable manner.

The limitations of current systems also hinder the ability of supermarkets to adapt to changing market conditions and customer preferences. For example, the increasing popularity of online shopping and home delivery services has introduced new complexities in inventory and order management. Traditional supermarket systems, designed primarily for in-store operations, struggle to accommodate these demands, further emphasizing the need for comprehensive solutions.

Addressing these challenges involves rethinking how supermarket systems are designed and implemented. Instead of relying on isolated modules for customer data, inventory, and payments, a unified system can provide a more holistic approach. Such a system would enable supermarkets to streamline operations, reduce errors, and enhance customer satisfaction. Additionally, it would create opportunities for innovation, such as dynamic pricing models and AI-driven recommendations, which could further differentiate supermarkets in a competitive market.

By analyzing the evolution of supermarket systems and their current limitations, it becomes clear that there is a significant opportunity to improve efficiency and customer satisfaction through better technology. This report explores these issues in detail, providing a comprehensive overview of the challenges faced by modern supermarkets and potential solutions to address them. Through this exploration, the report aims to contribute to the development of more effective and user-friendly systems that align with the needs of both supermarkets and their customers.

METHODOLOGY

Supermarket operations are multifaceted, encompassing customer management, inventory tracking, discount application, and payment processing. The proposed system addresses these challenges through structured methodologies designed to improve efficiency, accuracy, and customer satisfaction.

Customer Record Management

Managing customer records is fundamental to creating personalized experiences and improving service efficiency.

Existing Record Check involves maintaining a comprehensive customer database that includes details such as name, phone number, and discount percentage. Each time a customer initiates a transaction, their phone number is verified against the database.

This ensures that returning customers receive their applicable discounts while streamlining the service process.

Adding New Customers ensures inclusivity by allowing seamless onboarding. If the entered phone number does not exist in the database, a new record is created automatically. By assigning a default discount percentage of zero, the system accommodates new customers while laying the groundwork for future personalized benefits. This approach minimizes customer inconvenience and speeds up the checkout process.

Shopping List Management

Efficient handling of shopping lists is crucial for maintaining inventory accuracy and customer satisfaction.

Product Inventory Validation ensures that the products listed by the customer are matched against the supermarket's inventory. The system compares the product name and company name provided by the customer with the existing database. If a product match is found but the company name differs, the system prompts the user to confirm whether they wish to substitute the product or add the new combination to the inventory. This mechanism balances flexibility with inventory accuracy, enhancing the shopping experience.

Inclusive Pricing System is designed to calculate the total bill dynamically. The system incorporates the customer’s discount percentage and applicable taxes into the final price calculation. This ensures that pricing is fair, transparent, and tailored to each customer, fostering trust and loyalty.

Payment Integration

Payment processing is a critical component that directly impacts customer satisfaction and operational efficiency.

Cash or Card Options provide customers with the flexibility to choose their preferred payment method. For card payments, the

system integrates a mock bank balance checker to validate transactions. If the customer's account balance is sufficient, the transaction is approved, and a confirmation message is displayed.

Error Handling is incorporated to address issues such as insufficient funds or mismatched product entries. In such cases, the system displays clear warnings and provides customers with actionable options, such as using an alternate payment method or modifying their shopping list.

Technical Implementation

The proposed methodology relies on a robust technical framework to achieve its objectives.

Variables form the foundation of the system. These include arrays for product names (String[]), prices (double[]), and customer details. The variables are designed to store and retrieve data efficiently, supporting dynamic updates and scalability.

Methods like addProduct() and calculateTotal() streamline key operations. The addProduct() method facilitates the addition of new products to the inventory, while calculateTotal() handles the computation of the final bill, incorporating discounts and taxes.

Report Generation is an additional feature that summarizes customer activity and transaction details. These reports provide insights into shopping patterns and help supermarkets refine their operations over time.

IMPLEMENTATION

The proposed system was implemented as a Java application, structured to address the identified challenges comprehensively. The implementation process involved several critical components, which are detailed below.

Database Management

The system employs an in-memory database to store customer and product records. This approach offers fast data retrieval and eliminates the need for complex database configurations during the initial deployment phase. The database structure is designed to accommodate dynamic updates, allowing seamless addition, modification, or removal of records as needed.

For customer management, the database includes fields for name, phone number, and discount percentage. For product inventory, it maintains records of product names, company names, and prices. These fields are cross-referenced during shopping and billing operations, ensuring data consistency and accuracy.

Input Handling

User input is a critical component of the system. The application provides intuitive prompts for entering phone numbers, shopping lists, and payment preferences. Input validation mechanisms are in place to ensure that all entries conform to the expected format. For instance, phone numbers are checked for length and format, while product names are verified against the inventory database.

The shopping list interface allows customers to enter multiple products at once. For unmatched products or mismatched company names, the system provides clear prompts, enabling customers to make informed decisions. This user-friendly approach reduces errors and enhances the overall shopping experience.

Validation Mechanisms

String matching techniques are used to validate product entries. This involves comparing the customer’s shopping list with the inventory database to identify exact matches or near matches. For discrepancies, the system provides detailed feedback, such as suggesting alternative products or asking for confirmation to add new items.

Dynamic discount application is another key feature. Based on the customer's record, the system calculates the applicable discount percentage in real time and applies it to the final bill. This ensures that loyal customers receive the benefits they are entitled to while maintaining transparency.

Billing System

The billing system integrates all pricing components into a single, streamlined process. Taxes are applied at a predefined rate, and discounts are calculated dynamically based on customer records. The system generates a detailed bill that includes a breakdown of product prices, taxes, discounts, and the final total.

To further enhance transparency, the bill also includes a summary of any substitutions made during the shopping process. This ensures that customers are fully aware of the changes and associated costs.

Payment Processing

Payment handling is designed to be secure and efficient. For cash payments, the system simply records the transaction. For card payments, a mock bank balance system validates the customer's account before approving the payment.

The validation process involves checking whether the total bill amount is less than or equal to the available balance. If the validation is successful, the payment is processed, and a success message is displayed. For failed transactions, the system provides clear warnings and suggests alternate payment methods.

Advantages and Limitations

The implementation of the proposed system offers several advantages.

Advantages include:
1. Increased efficiency through automated processes.

2. Enhanced customer satisfaction due to personalized discounts and transparent billing.
3. Improved inventory accuracy through dynamic product validation.
4. Secure payment processing with real-time validation.

Limitations include:
1. Scalability challenges for larger supermarkets with extensive product catalogs.
2. Dependency on technical infrastructure, which may pose issues in low-resource settings.
3. Initial setup complexity, requiring thorough testing and configuration.

By addressing these limitations, the system can be further refined to meet the diverse needs of modern supermarkets, ensuring a seamless shopping experience for customers and efficient operations for retailers.

RESULTS AND DISCUSSION

The implementation of the Supermarket Application System yielded significant improvements in customer management, inventory validation, and payment processing. This section discusses the results obtained from the proposed system and compares them with existing systems to highlight the advancements achieved.

Improved Customer Management

One of the primary outcomes of the system was its ability to streamline customer record management. By maintaining a centralized database of customer details, including name, phone number, and discount percentage, the system ensured seamless interaction with returning customers. The automation of adding new customer records with a default discount percentage of zero reduced delays and made the system inclusive.

The personalized discount retrieval mechanism significantly enhanced the customer experience. Unlike traditional systems where discounts are static or generalized, this application dynamically adjusted discounts based on the customer's history. This capability not only built customer loyalty but also optimized the supermarket's promotional strategies.

Efficient Product Matching

The system demonstrated high accuracy in product-company pair validation. The inclusion of a matching mechanism for the shopping list minimized errors and confusion during the billing process.

In cases where the company name did not match the product in the inventory, the system's prompt for user confirmation added a layer of transparency. Customers were given the flexibility to substitute unmatched items or add new combinations to the database. This dynamic approach ensured that inventory remained up-to-date while catering to customer preferences.

Compared to existing systems that often rely on static inventory databases, this application introduced a more responsive method of managing product records. The capability to evolve the database in real time is a significant step forward in inventory accuracy.

Streamlined Payment System

The payment module effectively simplified the checkout process by offering both cash and card payment options. The integration of a mock bank balance validation system for card payments provided an added layer of security and reliability. Successful transactions triggered immediate confirmation messages, ensuring clarity for customers.

This approach addressed a common limitation in many existing supermarket systems, where card payments often fail due to incomplete validation checks. By simulating real-time bank balance checks, the system ensured that transactions were approved only when sufficient funds were available.

Comparison with Existing Systems

A comparative analysis with traditional and modern supermarket systems revealed several advantages of the proposed application.

1. Enhanced Inventory Tracking: Unlike traditional systems, which often rely on manual updates or periodic audits, the proposed system dynamically updates the inventory based on user inputs. This ensures that the database remains accurate and reflective of actual stock levels.
2. Personalized Discounting: Existing systems typically employ static discount policies that lack customer-specific adjustments. The proposed system introduced dynamic discounting, providing a tailored experience that fosters customer loyalty.
3. Error-Free Transactions: Manual billing processes in traditional systems are prone to errors, especially during peak hours. The automated validation and billing processes in the proposed system eliminated such errors, ensuring accurate and reliable transactions.

Limitations and Challenges

While the proposed system performed well in addressing many challenges, certain limitations were observed.

1. Scalability: The use of a mock in-memory database limits the system's scalability for large-scale supermarket operations. As the number of customers and products increases, the database may require significant optimization or migration to a more robust platform.
2. Dependency on Predefined Data: The initial setup of the product and customer databases requires considerable effort. Any inaccuracies in the predefined data can impact the system's performance.
3. Limited Payment Validation: Although the mock bank balance system simulated real-time validation, it does not account for more complex scenarios, such as network delays or external system failures, which could occur in a real-world environment.

Discussion

The results indicate that the proposed system effectively addresses the key challenges in supermarket operations. Improved customer management, accurate product matching, and a robust payment system contribute to a more seamless shopping experience.

However, the system's reliance on static databases and limited scalability highlight areas for future improvement. For larger supermarkets, the integration of cloud-based databases and advanced payment gateways would enhance performance and reliability.

In conclusion, the Supermarket Application System represents a significant advancement over existing systems by combining personalized customer management, dynamic inventory tracking, and secure payment processing. The results validate the system's ability to optimize supermarket operations while offering a superior shopping experience.

CONCLUSION

The Supermarket Application System addresses key challenges in managing customer records, product inventory, and payment processing. By integrating dynamic discount application, real-time inventory validation, and secure payment checks, the system significantly enhances the efficiency and reliability of supermarket operations.

One of the most impactful features of the system is its ability to streamline customer interactions through personalized discount retrieval and seamless onboarding of new customers. This not only builds customer loyalty but also reduces the manual effort required for managing records. Additionally, the system’s ability to validate product entries and prompt users for unmatched items ensures accurate billing and up-to-date inventory.

The payment processing module, which includes both cash and card options, adds flexibility and security to transactions. The mock bank balance checker simulates real-time validation, ensuring that card payments are processed efficiently without errors. These features collectively create a smooth and user-friendly checkout experience.

The implementation of this project demonstrates the potential for a simple yet powerful application to transform supermarket operations. While the system provides a robust framework, future enhancements could further elevate its capabilities. For instance, integrating AI-driven discount suggestions could enable even more personalized customer experiences. Advanced fraud detection mechanisms for payments and cloud-based scalability for larger databases could also address some of the current limitations.

In summary, this project serves as a foundational framework for developing next-generation supermarket systems. By addressing core operational challenges and offering scope for future innovation, the Supermarket Application System lays the groundwork for more advanced and comprehensive solutions in the retail industry.

SAMPLE CODING:

```
package Supermarket;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class SuperMarketApplicationSystem {

int cid;
    CustomerDetails[] C = new CustomerDetails[10];
    int i = 0;
    ArrayList<Product> productList;
    Cart cart;

    public SuperMarketApplicationSystem() {
```

```
        productList = new ArrayList<>();
        cart = new Cart();
        initializeProducts();
    }



    private void initializeProducts() {
        productList.add(new Product("Apple", 0.5, 100));
        productList.add(new Product("Banana", 0.3, 150));
        productList.add(new Product("Milk", 1.0, 50));
        productList.add(new Product("Bread", 1.5, 30));
        productList.add(new Product("Eggs", 2.0, 60));
        productList.add(new Product("Cheese", 3.0, 25));
        productList.add(new Product("Chicken Breast", 5.0, 40));
        productList.add(new Product("Rice", 0.8, 80));
        productList.add(new Product("Pasta", 1.2, 75));
        productList.add(new Product("Tomato Sauce", 2.5, 35));
        productList.add(new Product("Coffee", 7.0, 20));
        productList.add(new Product("Tea", 3.5, 50));
        productList.add(new Product("Frozen Vegetables", 2.0, 45));
        productList.add(new Product("Orange Juice", 3.0, 40));
        productList.add(new Product("Cereal", 4.0, 55));
        productList.add(new Product("Snack Bars", 1.5, 60));
        productList.add(new Product("Chips", 2.5, 100));
        productList.add(new Product("Soda", 1.0, 80));
        productList.add(new Product("Water Bottle", 0.5, 200));
        productList.add(new Product("Yogurt", 0.9, 30));
        productList.add(new Product("Chocolate", 1.2, 70));
    }

    public static void main(String[] args) {

          SuperMarketApplicationSystem system = new
SuperMarketApplicationSystem();



        String name;
        long phone;
        int cid,c;
        Scanner sc = new Scanner(System.in);
        operation s = new operation();

        System.out.println("Hello! Welcome to our Shop\nKindly Enter your
Name :");
```

```
        name = sc.nextLine();

        System.out.println("Enter your Phone Number :");
        phone = sc.nextLong();

        cid = s.checkCus(name, phone);

        System.out.println("\tSHOPPING MENU");
            int choice;
            do {
                System.out.println("1. Add Product to Cart\n2. View Cart\n3.
Discount Percentage\n4. Exit");
                choice = sc.nextInt();
                switch (choice) {
                    case 1:
                        system.addProductToCart(sc);
                        break;
                    case 2:
                        int a;
                        s.C[cid].discountPer += system.cart.viewCart()/10;
                        System.out.printf("Your Discount Percentage: %.2f%%\n",
s.C[cid].discountPer);
                        break;
                    case 4:
                        System.out.println("Exiting...");
                        break;
                    case 3:
                        if (cid != -1) {
                                    System.out.printf("Your Discount Percentage:
%.2f%%\n", s.C[cid].discountPer);
                                } else {
                                    System.out.println("No discount available for
new customers.");
                                }
                                break;
                    default:
                        System.out.println("Choose appropriate option");
                        break;
                }
        } while (choice != 4);


        sc.close();
    }
```

```
    private void addProductToCart(Scanner sc) {
        System.out.println("Available Products:");
        for (int j = 0; j < productList.size(); j++) {
            Product product = productList.get(j);
            System.out.printf("%d. %s - $%.2f (Stock: %d)\n", j + 1,
product.getName(), product.getPrice(), product.getStock());
        }

        System.out.println("Select a product by number:");
        int productChoice = sc.nextInt() - 1;

        if (productChoice >= 0 && productChoice < productList.size()) {
            System.out.println("Enter quantity:");
            int quantity = sc.nextInt();
            cart.addProduct(productList.get(productChoice), quantity);
        } else {
            System.out.println("Invalid product choice.");
        }
    }
}



class operation extends SuperMarketApplicationSystem {
    public int checkCus(String n, long ph) {
        for (int j = 0; j < i; j++) {
            if (ph == C[j].phone) {
                System.out.println("Welcome back " + C[j].name + "\nYour Discount
Percentage: " + C[j].discountPer + "%.");
                return j;
            }
        }
        System.out.println("Hello! New to our Shop Don't Worry " + n +
"\nAdding your details to our DB");
        addCus(n, ph);
        return i - 1; // Return the index of the newly added customer
    }

    public void addCus(String n, long ph) {
        if (i < C.length) {
            C[i] = new CustomerDetails(n, ph);
            System.out.println("Your Details have been Successfully Added to our
DB");
            i++;
        } else {
```

```
            System.out.println("Customer database is full. Cannot add more
customers.");
        }
    }
}

class CustomerDetails {
    public long phone;
    public String name;
    public double discountPer;

    CustomerDetails(String n, long p) {
        phone = p;
        name = n;
        discountPer = 0;
    }
}

class Product {
    String name;
    double price;
    int stock;

    public Product(String name, double price, int stock) {
        this.name = name;
        this.price = price;
        this.stock = stock;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public int getStock() {
        return stock;
    }

    public void reduceStock(int quantity) {
        stock -= quantity;
    }
}
```

```
class Cart {
    ArrayList<Product> products;
    double totalAmount;

    public Cart() {
        products = new ArrayList<>();
        totalAmount = 0.0;
    }

    public void addProduct(Product product, int quantity) {
        if (product.getStock() >= quantity) {
            products.add(product);
            product.reduceStock(quantity);
            totalAmount += product.getPrice() * quantity;
            System.out.println(quantity + " " + product.getName() + "(s) added to
the cart.");
        } else {
            System.out.println("Not enough stock for " + product.getName() + ".
Available stock: " + product.getStock());
        }
    }

    public  double viewCart() {
        System.out.println("Items in your cart:");
        for (Product product : products) {
            System.out.println("- " + product.getName() + ": $" +
product.getPrice());
        }
        System.out.printf("Total Amount: $%.2f\n", totalAmount);
        return totalAmount;
    }
}
```

OUTPUT:

```
Hello! Welcome to our Shop
Kindly Enter your Name :
Hari
Enter your Phone Number :
7904179709
Hello! New to our Shop Don't Worry Hari
Adding your details to our DB
Your Details have been Successfully
Added to our DB
SHOPPING MENU
1. Add Product to Cart
2. View Cart
3. Discount Percentage
4. Exit
1
Available Products:
1. Apple - $0.50 (Stock: 100)
2. Banana - $0.30 (Stock: 150)
3. Milk - $1.00 (Stock: 50)
4. Bread - $1.50 (Stock: 30)
5. Eggs - $2.00 (Stock: 60)
6. Cheese - $3.00 (Stock: 25)
7. Chicken Breast - $5.00 (Stock: 40)
8. Rice - $0.80 (Stock: 80)
9. Pasta - $1.20 (Stock: 75)
10. Tomato Sauce - $2.50 (Stock: 35)
11. Coffee - $7.00 (Stock: 20)
12. Tea - $3.50 (Stock: 50)
13. Frozen Vegetables - $2.00 (Stock: 45)
14. Orange Juice - $3.00 (Stock: 40)
15. Cereal - $4.00 (Stock: 55)
16. Snack Bars - $1.50 (Stock: 60)
17. Chips - $2.50 (Stock: 100)
18. Soda - $1.00 (Stock: 80)
19. Water Bottle - $0.50 (Stock: 200)
20. Yogurt - $0.90 (Stock: 30)
21. Chocolate - $1.20 (Stock: 70)
Select a product by number:
1
Enter quantity:
10
10 Apple(s) added to the cart.
Place order 1.Yes 2.No
1
Order Placed
```