

# Project Report

*of*

## *SuperMarket Smart Manager*

SUBMITTED BY : DEVAM YADAV

SAP ID : 590022492

COURSE : B.TECH CSE

COURSE CODE : (CSEG1032)

SUBMITTED TO : RAHUL PRASAD

# INDEX

<b>SR. NO.</b>	<b>NAME</b>	<b>PAGE NO.</b>
1	ABSTARCT	
2	PROBLEM DEFINITION	
3	SYSTEM DESIGN	
4	IMPLEMENTATION DETAILS	
5	TESTING & RESULTS	
6	CONCLUSION & FUTURE WORK	
7	REFFERENCE	
8	APPENDIX	

# ABSTRACT

This project presents a simple console based supermarket management system written in C.

It supports staff management, product operations, customer shopping flow, cart handling, billing.

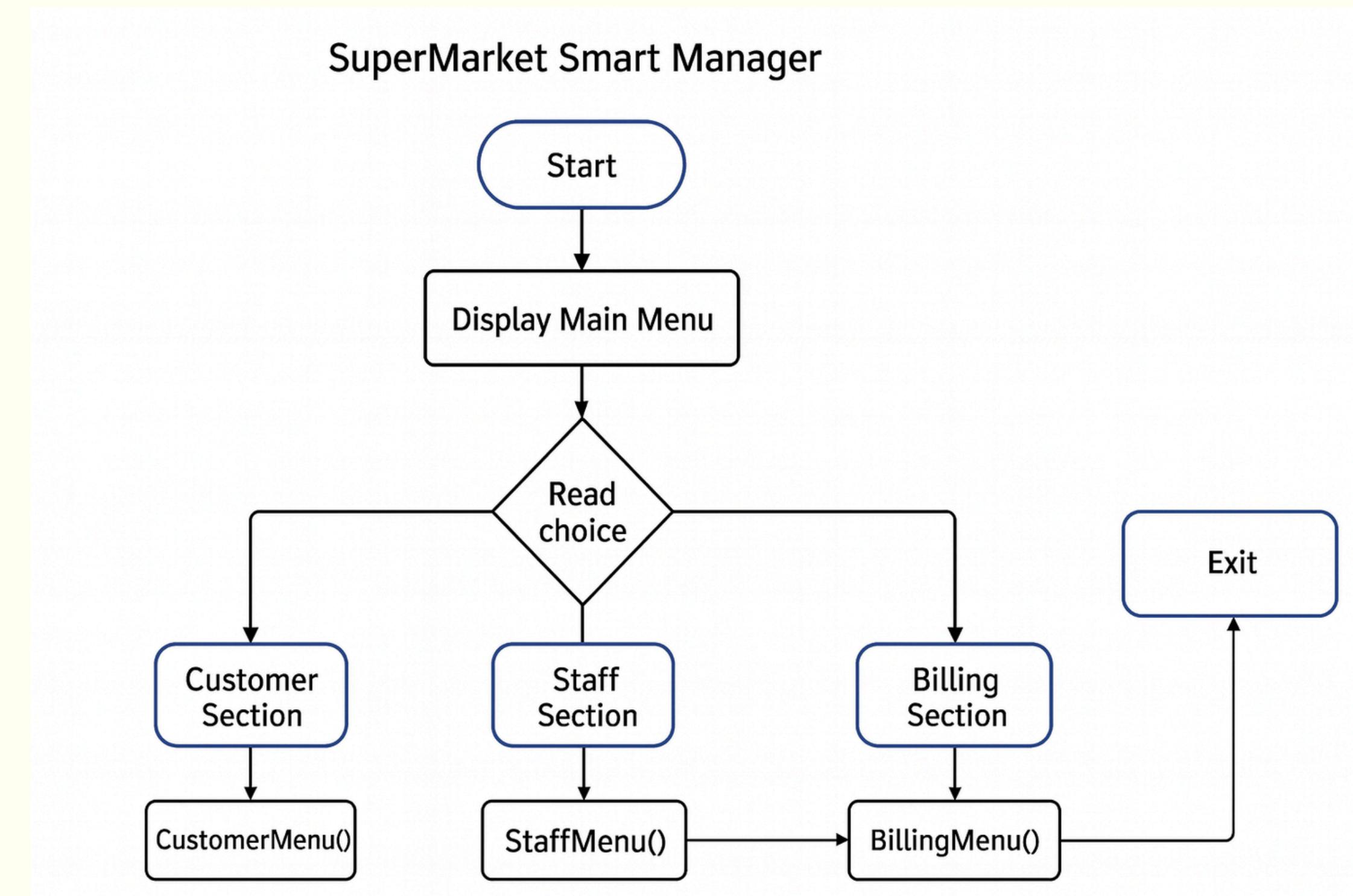
The goal is to provide a structured, modular implementation suitable for academic demonstration.

# PROBLEM DEFINITION

Retail shops require basic computerized systems to manage inventory, search items, add to cart, and generate bills.

The objective is to build a lightweight, menu driven supermarket system using C programming constructs such as structures, loops, functions, and arrays.

# FLOW-CHART



# SYSTEM DESIGN

## Algorithm

### Product Search Algorithm :

The Product Search Algorithm is used to locate an item in the supermarket inventory by matching the user-entered name with the stored product names.

The algorithm converts both the input name and each product name to lowercase to allow case-insensitive searching. It then compares the strings one by one.

If a match is found, the product details are displayed; otherwise, the system informs the user that the product was not found.

# Algorithm

## Add to Cart Algorithm :

The Add to Cart Algorithm allows the customer to select products to purchase. The system first takes a product ID from the user and searches for a matching product in the inventory.

If the product exists, the user is prompted to enter the desired quantity.

The quantity is validated to ensure enough stock is available.

If valid, the product along with its quantity and price is added to the cart.

The customer can continue adding items until they decide to stop.

# Algorithm

## Billing Algorithm with GST :

The Billing Algorithm calculates the total payable amount for all items in the cart. For each cart item, the algorithm multiplies the quantity with the product price to get the subtotal.

All subtotals are added to compute the gross amount.

# IMPLEMENTATION DETAILS

The SuperMarket Smart Manager system is implemented in the C programming language using modular functions, arrays, loops, and switch-case menus. The program manages products, customer cart operations, and billing using structured data types.

## Key Features Implemented (Brief Explanation)

- **Inventory display:**  
Shows the list of all available products with their ID, name, price, and stock.
- **Add new product:**  
Staff members can insert new products into the inventory by entering ID, name, price, and stock.
- **Update product price/stock:**  
Allows modifying the price or stock of an existing product through the staff menu.
- **Case-insensitive search:**  
Enables users to search products by name without worrying about uppercase or lowercase letters.
- **Add to cart with quantity validation:**  
Ensures that the entered quantity does not exceed the available stock before adding it to the cart.
- **Remove from cart:**  
Allows customers to delete an item from the cart using its product ID.
- **Billing with GST calculation:**  
Computes item totals, adds them together, calculates GST (18%), and prints the final payable amount.
- **Switch-case based menu navigation:**  
All menus use switch-case statements for clear, structured navigation and easier readability.

# TESTING

- Tested all menu paths:
- Verified that each option in the main, staff, customer, cart, and billing menus works correctly.
- Verified invalid inputs handling:
- Ensured the system responds properly to wrong menu choices, nonexistent product IDs, or empty carts.
- Verified stock validation and cart logic:
- Confirmed that items cannot be added if requested quantity exceeds stock, and cart updates work correctly.
- Verified billing accuracy:
- Tested calculations for subtotal, GST, and final amount to ensure correct mathematical results.

# RESULTS

- The SuperMarket Smart Manager system works successfully across all key modules.
- All features were executed and tested, showing correct responses and output formatting.
- The system successfully manages inventory, including adding and updating products.
  - Case-insensitive search accurately finds items regardless of letter case.
- Cart operations (add, view, remove) work smoothly with correct quantity checks.
- Billing module correctly calculates total amount and GST, and prints a clear final bill.
  - All menu navigation using switch-case works without errors or crashes.
- The program handles invalid inputs properly and prevents incorrect operations.
- Overall, the program meets all functional requirements and performs reliably.

# CONCLUSION & FUTURE WORK

## Conclusion:

- The SuperMarket Smart Manager project effectively demonstrates the core principles of structured programming using the C language. Through the use of functions, structures, arrays, loops, and switch-case logic, the system successfully performs essential supermarket operations such as inventory management, product search, cart handling, and billing. The modular design ensures clarity, maintainability, and ease of understanding, making it suitable for academic learning as well as real-world basic retail operations. Overall, the project fulfills all functional requirements and provides a stable, user-friendly experience.

## Future Work:

- While the current system functions efficiently for basic supermarket operations, several enhancements can be incorporated to expand its capabilities. Future improvements may include:
  - File Storage: Implementing file handling to save and load product data, cart details, and billing history, allowing the system to maintain data across sessions.
  - User Authentication: Adding secure login features for staff and administrators to restrict access to inventory modification features.
  - Graphical User Interface (GUI): Developing a GUI using technologies like GTK or a desktop framework to make the system more visually appealing and easier to interact with.
  - Category-Based Filtering: Introducing product categories (e.g., fruits, dairy, grains) to improve product organization and search efficiency.
  - Enhanced Billing Summaries: Adding features such as discount handling, loyalty points, or exporting bills as PDF files for better customer experience and documentation.
- These enhancements would significantly increase the system's functionality, usability, and real-world applicability.

# REFERENCES

1. Mr. Rahul Prasad
2. Let us C by Yashwant Kanetkar
3. <https://www.programiz.com/c-programming/online-compiler>
4. <https://www.geeksforgeeks.org/c/c-programming-language/>