Title Page

Project Title:

Restaurant Billing System

Course: B.Tech (Computer Science and Engineering)

Submitted By: Tanisha Pal

SAP ID: 590022280

Submitted To: Mrs. Suchetana Sadhukhan

Department: SOCS (School of Computer Science)

Institution: University of Petroleum and Energy Studies

Abstract

The Restaurant Billing System is a simple computer-based application designed to automate billing operations in a restaurant. It provides functionality for menu display, item selection, quantity entry, and automatic calculation of totals including tax. The system eliminates manual calculation errors, improves accuracy, and speeds up the customer checkout process. This project uses the C programming language and demonstrates fundamental programming concepts such as structures, arrays, loops, and decision-making. The system is efficient, easy to use, and suitable for small to medium-scale food service businesses.

Problem Definition

In many small restaurants, billing is still performed manually. This causes several problems:

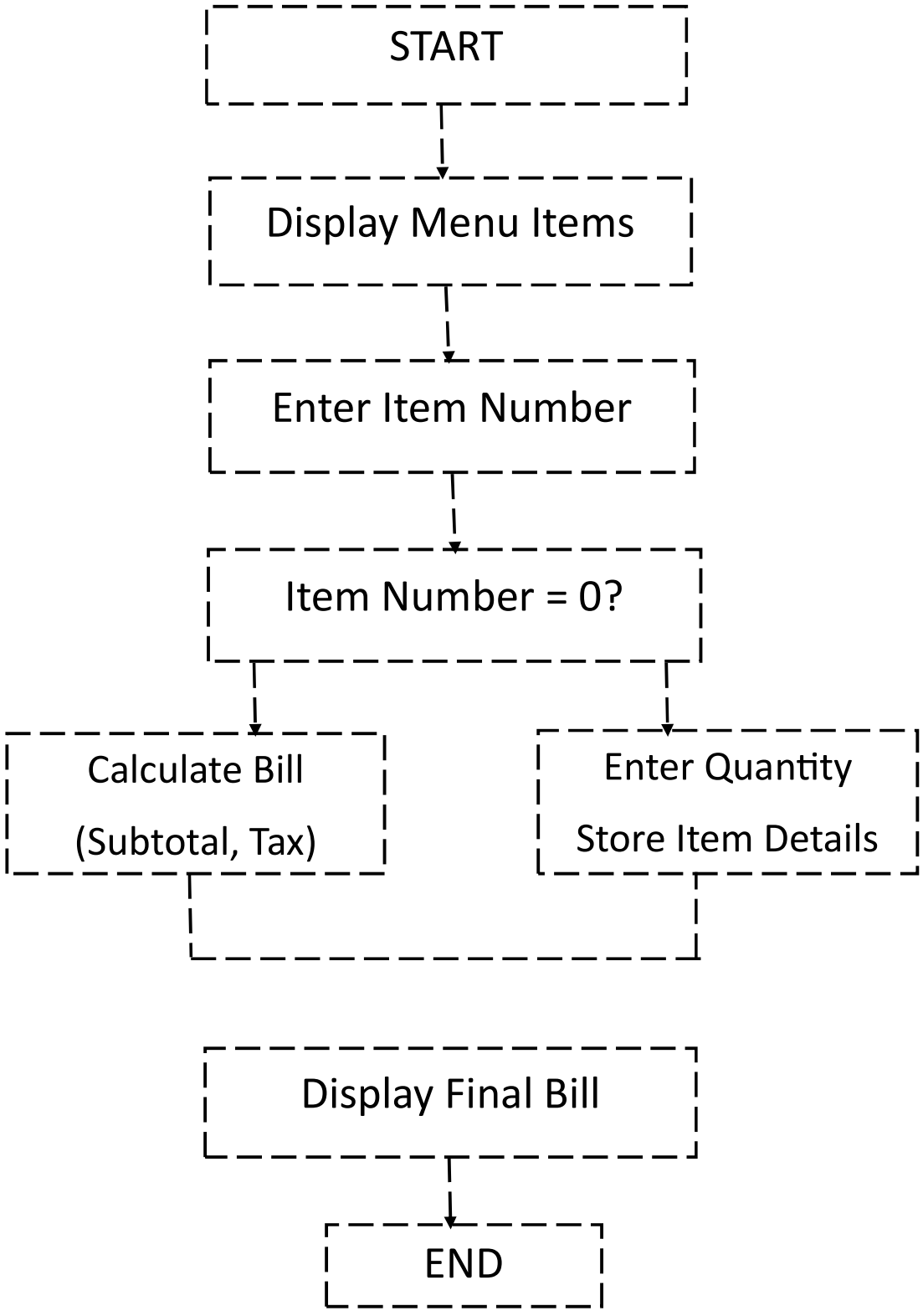
- Errors in price calculation
- Delays in billing customers
- Difficulty in updating totals for multiple orders
- Lack of a standardized billing format
- Inefficient record keeping

The goal of this project is to design a simple, user-friendly application that automates the restaurant billing process. The system must:

1. Display menu items with prices
2. Allow customers to select items and quantities
3. Calculate subtotal, tax, and final total
4. Provide a clear billing summary
5. Minimize human error and speed up operations

System Design

4.1 Flowchart



4.2 Algorithm

Step 1: Start
Step 2: Display menu with item numbers and prices
Step 3: Ask user for item number
Step 4: If item number is 0, proceed to billing
Step 5: Otherwise, ask for quantity
Step 6: Store order details (item name, price, quantity, total)
Step 7: Repeat steps until 0 is entered
Step 8: Calculate subtotal by summing all item totals
Step 9: Calculate tax and final total
Step 10: Display formatted bill
Step 11: End

5. Implementation Details (with code snippets)

Language Used: C Programming Language

Compiler: GCC / Turbo C / CodeBlocks

Concepts Used: Arrays, Structures, Loops, Conditions, Functions

```
#include <stdio.h>
#include <string.h>
#define MAX_ITEMS 100
#define TAX_RATE 0.08
struct Order {
    char itemName[30];
    float price;
    int quantity;
    float totalPrice;
};
int main() {
char menuItems[][30] = {
        "Burger", "Pizza", "Pasta", "Fries", "Coke", "Coffee"
    };
float menuPrices[] = {
        5.99, 8.49, 7.25, 2.99, 1.50, 2.00
    };
int totalMenuItems = 6;
    struct Order orders[MAX_ITEMS];
    int orderCount = 0;
 int choice, qty;
```

```
 printf("================ WELCOME TO C RESTAURANT ================\n");
    printf("----------------------- MENU -------------------------\n");
for (int i = 0; i < totalMenuItems; i++) {
        printf("%d. %-10s   Rs%.2f\n", i + 1, menuItems[i], menuPrices[i]);
    }
printf("----------------------------------------------------------\n");
 while (1) {
        printf("\nEnter the item number (0 to finish ordering): ");
        scanf("%d", &choice);
   if (choice == 0) {
            break;
        }
   if (choice < 1 || choice > totalMenuItems) {
            printf("Invalid choice! Please choose a number from the menu.\n");
            continue;
        }
       printf("Enter quantity: ");
        scanf("%d", &qty);
     strcpy(orders[orderCount].itemName, menuItems[choice - 1]);
       orders[orderCount].price = menuPrices[choice - 1];
       orders[orderCount].quantity = qty;
       orders[orderCount].totalPrice = qty * menuPrices[choice - 1];
     orderCount++;
      printf("Added %d x %s to order (Subtotal: Rs%.2f)\n",
              qty,
              menuItems[choice - 1],
              qty * menuPrices[choice - 1]);
```

```
    }
  float subtotal = 0;
    for (int i = 0; i < orderCount; i++) {
        subtotal += orders[i].totalPrice;
    }
   float tax = subtotal * TAX_RATE;
    float finalTotal = subtotal + tax;
    printf("\n====================== BILL SUMMARY
======================\n");
    printf("%-15s %-10s %-10s %-10s\n", "Item", "Price", "Qty", "Total");
    for (int i = 0; i < orderCount; i++) {
printf("%-15s Rs%-9.2f %-10d Rs%-10.2f\n",
              orders[i].itemName,
              orders[i].price,
              orders[i].quantity,
              orders[i].totalPrice);
    }
    printf("-------------------------------------------------------\n");
    printf("Subtotal:                    Rs%.2f\n", subtotal);
    printf("Tax (8%%):                   Rs%.2f\n", tax);
    printf("TOTAL:                       Rs%.2f\n", finalTotal);
    printf("=================================================
=========\n");
    printf("Thank you for dining with us! Have a great day!\n");


    return 0;
}
```

6. Testing & Results

6.1 Test Cases

Test Case	Input	Expected Output
1	Burger	Total = 2 * 5.99 = 11.98
2	Pizza	Total = 3 * 8.49 = 25.47
3	Pasta	Total = 1 * 7.25 = 7.25
4	Fries	Total = 2 * 2.99 = 5.98

6.2 Sample Output

```
====================== BILL SUMMARY ======================
Item            Price      Qty        Total
Burger          Rs5.99      2          Rs11.98
Pizza           Rs8.49      3          Rs25.47
Pasta           Rs7.25      1          Rs7.25
Fries           Rs2.99      2          Rs5.98
----------------------------------------------------------
Subtotal:                           Rs50.68
Tax (8%):                           Rs4.05
TOTAL:                              Rs54.73
==========================================================
Thank you for dining with us! Have a great day!
```

Test Result:

All test cases passed successfully. The system performed calculations accurately and handled errors correctly.

7. Conclusion & Future Work

Conclusion

The Restaurant Billing System successfully automates the process of billing in a restaurant. It reduces human errors, speeds up service, and provides a clean summary of orders. The project demonstrates proficiency in C programming concepts such as structures, arrays, loops, and decision-making.

Future Work

The system can be enhanced with future improvements:

- Integration with a graphical user interface (GUI)
- Adding database support for menu storage
- Printing or exporting bills to a file
- Adding user authentication for staff
- Implementing discounts, coupons, and membership offers
- Multi-table order management

8. References

1. Yashavant Kanetkar, *Let Us C*, BPB Publications
2. Brian W. Kernighan & Dennis M. Ritchie, *The C Programming Language*
3. Online C Compiler Documentation
4. Lecture notes from course [B-tech School of computer science]