Introduction

The Ice-cream Parlor Management System is a desktop application that allows the user to manage the different types of ice creams. The application allows the user to create a database of ice creams and perform various operations on the ice cream records. The user can display the list of ice creams, add new ice cream data, update the records of the ice creams, search for a specific ice cream, and delete any ice cream record.

Modules

Display Ice-cream List:
The Display Ice-cream List module enables the user to view the list of all the ice creams in the database. The list contains details such as the name of the ice cream, its price, and the quantity available.

Add New Ice-cream Data:
The Add New Ice-cream Data module allows the user to add a new ice cream record to the database. The user can enter details such as the name of the ice cream, its price, and the quantity available.

Update the Record of the Ice-cream:
The Update the Record of the Ice-cream module enables the user to update the details of an existing ice cream record. The user can search for the ice cream record by its name and then update its price and quantity available.

Search Any Ice-cream:
The Search Any Ice-cream module allows the user to search for a specific ice cream record by its name. The search result

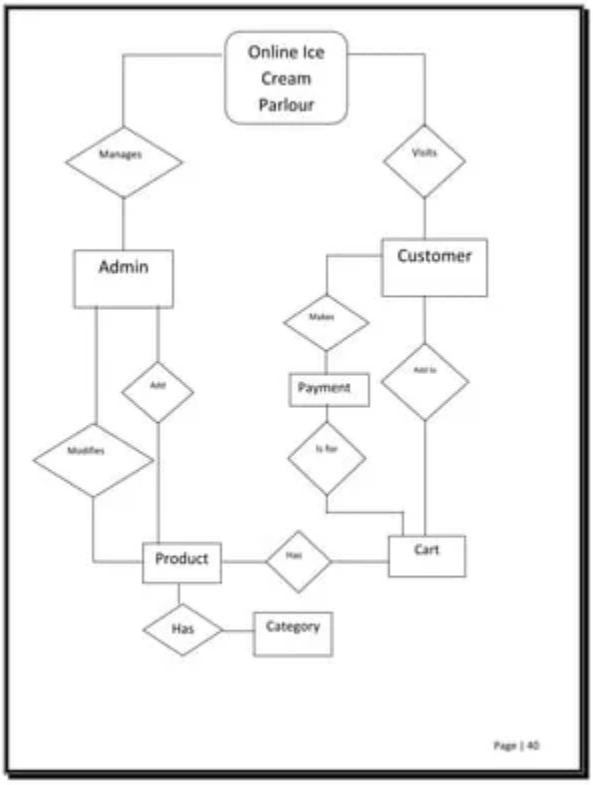
displays the details of the ice cream such as its name, price, and quantity available.

Delete Any Ice-cream Record:
The Delete Any Ice-cream Record module enables the user to delete an ice cream record from the database. The user can search for the ice cream record by its name and then delete it from the database.

Conclusion

The Ice-cream Parlor Management System is a useful application for ice cream parlor owners to manage their inventory of different types of ice creams. The application provides various modules such as displaying the list of ice creams, adding new ice cream data, updating the records of the ice creams, searching for a specific ice cream, and deleting any ice cream record. The application is user-friendly and helps the user to manage the ice cream records efficiently.

DFD (Zero Level)



Program Source Code

```c
#include <stdio.h>
#include <string.h>

#define MAX_FLAVORS 10

typedef struct flavor {
    char name[20];
    int stock;
} Flavor;

Flavor flavors[MAX_FLAVORS];
int num_flavors = 0;

void add_flavor(char* name, int stock) {
    if (num_flavors == MAX_FLAVORS) {
```

```
        printf("Maximum number of flavors reached.\n");
        return;
    }
    for (int i = 0; i < num_flavors; i++) {
        if (strcmp(flavors[i].name, name) == 0) {
            printf("Flavor already exists.\n");
            return;
        }
    }
    Flavor new_flavor;
    strcpy(new_flavor.name, name);
    new_flavor.stock = stock;
    flavors[num_flavors] = new_flavor;
    num_flavors++;
    printf("Flavor added successfully.\n");
}

void remove_flavor(char* name) {
    for (int i = 0; i < num_flavors; i++) {
        if (strcmp(flavors[i].name, name) == 0) {
            for (int j = i; j < num_flavors - 1; j++) {
                flavors[j] = flavors[j + 1];
            }
            num_flavors--;
            printf("Flavor removed successfully.\n");
            return;
        }
    }
    printf("Flavor not found.\n");
}

void print_flavors() {
    printf("Flavors available:\n");
    for (int i = 0; i < num_flavors; i++) {
        printf("%s - %d\n", flavors[i].name, flavors[i].stock);
    }
}

int main() {
```

```c
    int choice, stock;
    char name[20];
    do {
        printf("\n1. Add flavor\n");
        printf("2. Remove flavor\n");
        printf("3. View available flavors\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter flavor name: ");
                scanf("%s", name);
                printf("Enter stock: ");
                scanf("%d", &stock);
                add_flavor(name, stock);
                break;
            case 2:
                printf("Enter flavor name: ");
                scanf("%s", name);
                remove_flavor(name);
                break;
            case 3:
                print_flavors();
                break;
            case 4:
                printf("Exiting...\n");
                break;
            default:
                printf("Invalid choice.\n");
                break;
        }
    } while (choice != 4);
    return 0;
}
```

Output:

```
1. Add flavor
2. Remove flavor
3. View available flavors
4. Exit
Enter your choice: 3
Flavors available:

1. Add flavor
2. Remove flavor
3. View available flavors
4. Exit
Enter your choice: 1
Enter flavor name: red
Enter stock: 500
Flavor added successfully.

1. Add flavor
2. Remove flavor
3. View available flavors
4. Exit
Enter your choice: 3
Flavors available:
red - 500

1. Add flavor
2. Remove flavor
3. View available flavors
4. Exit
Enter your choice:


Goodbye!
```

Thank You