Ice Cream Parlor Application Documentation

This documentation provides a detailed explanation of the Ice Cream Parlor Application, a simple CLI-based application for managing an ice cream parlor's seasonal flavor offerings, ingredient inventory, customer flavor suggestions, and allergy concerns. The application is built using Python and SQLite.

Table of Contents

Database Setup
File: database_setup.py

This script sets up the SQLite database and creates the necessary tables.

```
import sqlite3

# Connect to the SQLite database (or create it if it doesn't
exist)
conn = sqlite3.connect('ice_cream_parlor.db')
cursor = conn.cursor()

# Create the tables
cursor.execute('''CREATE TABLE IF NOT EXISTS SeasonalFlavors (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      name TEXT NOT NULL,
      description TEXT,
      ingredients TEXT
)''')

cursor.execute('''CREATE TABLE IF NOT EXISTS Ingredients (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      name TEXT NOT NULL,
      quantity INTEGER
)''')

cursor.execute('''CREATE TABLE IF NOT EXISTS CustomerSuggestions (
      id INTEGER PRIMARY KEY AUTOINCREMENT,
      name TEXT NOT NULL,
      suggested_flavor TEXT,
```

```
    allergy_concern TEXT
)''')

cursor.execute('''CREATE TABLE IF NOT EXISTS Allergens (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL UNIQUE
)''')

cursor.execute('''CREATE TABLE IF NOT EXISTS Cart (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    flavor_id INTEGER,
    FOREIGN KEY (flavor_id) REFERENCES SeasonalFlavors (id)
)''')

# Commit changes and close the connection
conn.commit()
conn.close()
```

Explanation

- Database Connection: Establishes a connection to the SQLite database ice_cream_parlor.db. If the database does not exist, it will be created.
- Table Creation: Creates the tables SeasonalFlavors, Ingredients, CustomerSuggestions, Allergens, and Cart if they do not already exist.
- Commit and Close: Commits the changes and closes the database connection.

Database Operations
File: db_operations.py

This script contains functions to perform various database operations.

```
import sqlite3

def add_seasonal_flavor(name, description, ingredients):
    conn = sqlite3.connect('ice_cream_parlor.db')
    cursor = conn.cursor()
    cursor.execute('''INSERT INTO SeasonalFlavors (name,
description, ingredients) VALUES (?, ?, ?)''', (name, description,
ingredients))
    conn.commit()
    conn.close()

def add_ingredient(name, quantity):
```

```
    conn = sqlite3.connect('ice_cream_parlor.db')
    cursor = conn.cursor()
    cursor.execute('''INSERT INTO Ingredients (name, quantity)
VALUES (?, ?)''', (name, quantity))
    conn.commit()
    conn.close()

def add_customer_suggestion(name, suggested_flavor,
allergy_concern):
    conn = sqlite3.connect('ice_cream_parlor.db')
    cursor = conn.cursor()
    cursor.execute('''INSERT INTO CustomerSuggestions (name,
suggested_flavor, allergy_concern) VALUES (?, ?, ?)''', (name,
suggested_flavor, allergy_concern))
    conn.commit()
    conn.close()

def add_allergen(name):
    conn = sqlite3.connect('ice_cream_parlor.db')
    cursor = conn.cursor()
    cursor.execute('''INSERT INTO Allergens (name) VALUES (?)''',
(name,))
    conn.commit()
    conn.close()

def search_flavors(keyword):
    conn = sqlite3.connect('ice_cream_parlor.db')
    cursor = conn.cursor()
    cursor.execute('''SELECT * FROM SeasonalFlavors WHERE name
LIKE ? OR description LIKE ?''', ('%' + keyword + '%', '%' +
keyword + '%'))
    results = cursor.fetchall()
    conn.close()
    return results

def add_to_cart(flavor_id):
    conn = sqlite3.connect('ice_cream_parlor.db')
    cursor = conn.cursor()
    cursor.execute('''INSERT INTO Cart (flavor_id) VALUES (?)''',
(flavor_id,))
    conn.commit()
    conn.close()

def view_cart():
    conn = sqlite3.connect('ice_cream_parlor.db')
    cursor = conn.cursor()
    cursor.execute('''SELECT SeasonalFlavors.name FROM Cart JOIN
SeasonalFlavors ON Cart.flavor_id = SeasonalFlavors.id''')
```

```
    results = cursor.fetchall()
    conn.close()
    return results
```

Explanation

- add_seasonal_flavor(name, description, ingredients): Adds a new seasonal flavor to the SeasonalFlavors table.
- add_ingredient(name, quantity): Adds a new ingredient to the Ingredients table.
- add_customer_suggestion(name, suggested_flavor, allergy_concern): Records a customer suggestion in the CustomerSuggestions table.
- add_allergen(name): Adds a new allergen to the Allergens table.
- search_flavors(keyword): Searches for flavors in the SeasonalFlavors table based on a keyword.
- add_to_cart(flavor_id): Adds a flavor to the cart.
- view_cart(): Displays the contents of the cart.

Main Application
File: app.py

This script provides a CLI for interacting with the application.

```
from db_operations import (
    add_seasonal_flavor, add_ingredient, add_customer_suggestion,
    add_allergen, search_flavors, add_to_cart, view_cart
)

def main():
    while True:
    print("\nWelcome to the Ice Cream Parlor!")
    print("1. Add Seasonal Flavor")
    print("2. Add Ingredient")
    print("3. Add Customer Suggestion")
    print("4. Add Allergen")
    print("5. Search Flavors")
    print("6. Add to Cart")
    print("7. View Cart")
    print("8. Exit")
    choice = input("Choose an option: ")

    if choice == '1':
        name = input("Enter flavor name: ")
        description = input("Enter flavor description: ")
```

```
            ingredients = input("Enter ingredients (comma
separated): ")
            add_seasonal_flavor(name, description, ingredients)
            print("Seasonal flavor added!")
      elif choice == '2':
            name = input("Enter ingredient name: ")
            quantity = int(input("Enter quantity: "))
            add_ingredient(name, quantity)
            print("Ingredient added!")
      elif choice == '3':
            name = input("Enter your name: ")
            suggested_flavor = input("Enter suggested flavor: ")
            allergy_concern = input("Enter any allergy concerns: ")
            add_customer_suggestion(name, suggested_flavor,
allergy_concern)
            print("Customer suggestion added!")
      elif choice == '4':
            name = input("Enter allergen name: ")
            add_allergen(name)
            print("Allergen added!")
      elif choice == '5':
            keyword = input("Enter search keyword: ")
            results = search_flavors(keyword)
            if results:
                  print("Search Results:")
                  for result in results:
                        print(f"ID: {result[0]}, Name: {result[1]},
Description: {result[2]}, Ingredients: {result[3]}")
            else:
                  print("No results found.")
      elif choice == '6':
            flavor_id = int(input("Enter flavor ID to add to cart:
"))
            add_to_cart(flavor_id)
            print("Flavor added to cart!")
      elif choice == '7':
            cart_items = view_cart()
            if cart_items:
                  print("Cart Items:")
                  for item in cart_items:
                        print(f"- {item[0]}")
            else:
                  print("Cart is empty.")
      elif choice == '8':
            print("Goodbye!")
            break
      else:
            print("Invalid choice, please try again.")
```

```
if __name__ == "__main__":
      main()
```

Explanation

- main(): The main function that provides a CLI for the user to interact with the application.
- Add Seasonal Flavor: Prompts the user to enter details of a new seasonal flavor and adds it to the database.
- Add Ingredient: Prompts the user to enter details of a new ingredient and adds it to the database.
- Add Customer Suggestion: Prompts the user to enter a customer suggestion and adds it to the database.
- Add Allergen: Prompts the user to enter a new allergen and adds it to the database.
- Search Flavors: Prompts the user to enter a keyword and displays search results from the SeasonalFlavors table.
- Add to Cart: Prompts the user to enter a flavor ID to add to the cart.
- View Cart: Displays the contents of the cart.
- Exit: Exits the application.