DSA Final Project:

Title: Contact Book Management

Group Members:

Syed Agha Shah Hassan(44483)

Muhammad Talha Aslam(44220)

ContactBook C++ Code Documentation

1. Introduction

This C++ code implements a simple ContactBook using a doubly linked list to store contacts. The program allows users to perform various operations such as adding, editing, deleting contacts, searching for contacts, displaying all contacts, and deleting all contacts.

2. Code Structure

The code is structured into a class named `ContactBook`, which contains member functions to handle different operations. The main function initializes an instance of the ContactBook class and provides a user interface for interacting with the ContactBook.

2.1. Node Structure

A `Node` structure is defined to represent individual contacts in the ContactBook. Each node contains the name, phone number, and pointers to the next and previous nodes.

2.2. ContactBook Class

The `ContactBook` class encapsulates the functionalities of the ContactBook. It includes member functions for adding contacts, displaying contacts, searching contacts, deleting contacts, editing contacts, and saving/loading contacts to/from a file.

3. Code Implementation

3.1. Constructor

The `ContactBook` class has a constructor that initializes the head of the linked list and sets default values for the contact name and phone number.

3.2. CreateNode()

The `CreateNode` function allows users to add a new contact to the ContactBook. It prompts the user for the name and phone number and creates a new node in the linked list.

3.3. Display()

The `Display` function prints all contacts in the ContactBook. It sorts the contacts using the BubbleSort algorithm before displaying them.

3.4. Search()

The `Search` function allows users to search for contacts by name or phone number. It prompts the user for a search criterion and displays the contact details if found.

3.5. DeleteAllContacts()

The `DeleteAllContacts` function deletes all contacts in the ContactBook, freeing up memory.

3.6. DeleteContactBySearch()

The `DeleteContactBySearch` function allows users to delete a specific contact by searching for it based on the name or phone number.

3.7. BubbleSort()

The `BubbleSort` function sorts the contacts alphabetically by name using the BubbleSort algorithm.

3.8. EditContacts()

The `EditContacts` function allows users to edit the details of a specific contact by searching for it based on the name or phone number.

3.9. OflineSave() and reopenCB()

These functions handle the offline saving and reopening of the ContactBook to/from a file named "contactbook.txt."

4. Usage

The `main` function initializes an instance of the `ContactBook` class, prompts the user for their name, and displays a welcome message. It then calls the `Structure` function to provide a menu for users to interact with the ContactBook.

5. Conclusion

This C++ ContactBook implementation provides a basic framework for managing contacts efficiently. Users can easily add, edit, delete, and search for contacts in a user-friendly interface.