

Project

Submitted To Sir Usman Ghani

Submitted By Safwan Rao

Roll No S23BDOCS1M01225

Subject Data structure & Algorithm

Department Computer Science

*Report on the Phone Directory Application Code*

1. Introduction

This report details the implementation and functionality of a simple phone directory application written in C++. The application allows users to add, search, and display contacts. The code utilizes basic C++ constructs, including vectors, strings, structures, and control flow statements to achieve its functionality.

2. Code Overview

The application is composed of several functions that handle different tasks:

1. \*Data Structure Definition\*: A struct named Contact is used to store contact information.

2. \*Function Definitions\*: Three main functions are defined:

- addContact(): Adds a new contact to the contacts vector.

- searchByName(): Searches for a contact by name.

- displayContacts(): Displays all contacts in the vector.

3. \*Main Function\*: The main() function provides a menu-driven interface for the user to interact with the application.

3. Detailed Code Explanation

3.1. Data Structure

**cpp**

struct Contact {

string name;

string phoneNumber;

};

- The Contact structure holds two members: name and phoneNumber, both of which are strings.

3.2. Adding a Contact

cpp

void addContact(vector<Contact>& contacts) {

Contact newContact;

cout << "Enter name: ";

cin >> newContact.name;

cout << "Enter phone number: ";

cin >> newContact.phoneNumber;

contacts.push\_back(newContact);

}

- Prompts the user to enter a name and phone number.

- Creates a Contact object and adds it to the contacts vector.

3.3. Searching for a Contact

**cpp**

void searchByName(const vector<Contact>& contacts, const string& name) {

for (const auto& contact : contacts) {

if (contact.name == name) {

cout << "Found contact: " << contact.name << " (" << contact.phoneNumber << ")\n";

return;

}

}

cout << "Contact not found.\n";

}

- Iterates through the contacts vector to find a contact with the given name.

- If found, prints the contact’s information; otherwise, prints a "Contact not found" message.

##### 3.4. Displaying All Contacts

cpp

void displayContacts(const vector<Contact>& contacts) {

cout << "Contacts:\n";

for (const auto& contact : contacts) {

cout << contact.name << " (" << contact.phoneNumber << ")\n";

}

}

- Iterates through the contacts vector and prints each contact’s name and phone number.

##### 3.5. Main Function and User Interface

cpp

int main() {

vector<Contact> contacts; // Store contacts

int choice;

do {

cout << "\nPhone Directory Application\n";

cout << "1. Add Contact\n";

cout << "2. Search by Name\n";

cout << "3. Display All Contacts\n";

cout << "4. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

addContact(contacts);

break;

case 2:

cout << "Enter name to search: ";

string searchName;

cin >> searchName;

searchByName(contacts, searchName);

break;

case 3:

displayContacts(contacts);

break;

case 4:

cout << "Exiting. Goodbye!\n";

break;

default:

cout << "Invalid choice. Try again.\n";

}

} while (choice != 4);

return 0;

}

- Implements a loop that displays a menu and processes user input.

- The user can choose to add a contact, search for a contact, display all contacts, or exit the application.

- Uses a switch statement to call the appropriate function based on the user’s choice.

4. Functionality and User Interaction

- \*Adding Contacts\*: The user is prompted to enter a name and phone number, which are then stored in the contacts vector.

- \*Searching Contacts\*: The user can search for a contact by name. If the contact exists, its details are displayed.

- \*Displaying Contacts\*: All contacts stored in the contacts vector are displayed with their names and phone numbers.

- \*Exiting\*: The user can exit the application by choosing the appropriate option from the menu.

5. Conclusion

This phone directory application provides a simple but effective way to manage contacts. It demonstrates basic programming concepts such as structures, vectors, loops, and conditionals in C++. The application can be further enhanced by adding features like editing or deleting contacts, saving contacts to a file, or handling duplicate entries.

https://github.com/Safwanrao/Phone-Directory/upload