./

Report – Phonebook

Course Code: <CODE>



Version Number: 1.0.0

Team Members: Sandesh Prabhu

Candidate ID: 105209

Module: C++ Programming

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

# 

Problem statement

|  |  |  |
| --- | --- | --- |
| Table of contents | | |
| Sl no | Name | Page No |
| 1 | Problem statement | 3 |
| 2 | Description | 3 |
| 3 | Data requirements | 4 |
| 4 | Software requirements | 4 |
| 5 | System design (Flowcharts) | 5-11 |
| 6 | Test plan | 12 |
| 7 | Expected output | 13 |

Problem statement

To develop a phonebook management system using C++ programming (OOPs) which can add, delete, modify, and search the record in the system.

Description

Phone book system is a program to store all-important contact details. This will help the user to easily search and manage contacts using this system. The OOP concepts of C++ like Functions, Classes and objects are used in this program. This program will add, delete, modify or edit, search and list records in the database.

The homepage of the program contains the main menu which has basic functions of the phonebook management system. When adding a record to the phone book, the system will ask for personal information such as name, gender, father's name, phone number, email and address. Then you can modify, list, search and delete these records.

For main menu switch case is used and displayed all the options of menu

Menu list includes:

-> List all contacts: lists all the contacts stored in the database with their respective contact details.

-> Add new contact: with information such as name, phone number, address, gender and email.

-> Delete contact: deletes a contact from database.

-> Edit contacts: edit information given while adding the contacts – name, phone number, address, and email.

-> Search contacts: based on name.

-> Exit: To exit the program.

The phonebook database stores the information about the user. A class is defined with private members as first name, last name, gender, phone number and email id. A constructer is defined to initialize the data members. The public members contains member function like adding, listing, deleting, modifying and searching. Classes and objects helps organize complex data in a more meaningful way with its associated member functions. The major advantage of C++ over C programming is support for Object Oriented Programming.

Data Requirements

Phone Book requires the data such as name, gender, mobile number, email id, address, birthday date etc. program uses an array of structures to store the information. Each element of the structure array has the information of the single contact.

Software requirements

1) OS: - Windows XP/7/10

2) Compilers: - G++ Compiler.

3) Others:-

1. Doxygen: - to document the code.
2. Make: - to build and manage the project automatically.

System design

The program is divided into two parts. First part contains the main function to choose the operation to be performed on the phonebook. Second part contains all the definitions of operations like listing, adding, deleting, modifying and searching.

In the program, the user enters the choice and it’s further processed using the switch case. Then control takes to the particular case and executes the corresponding function to it.

If the choice is entered beyond 6, then it prompts the user to enter the choice again. If choice 6 is chosen then it exits the program.

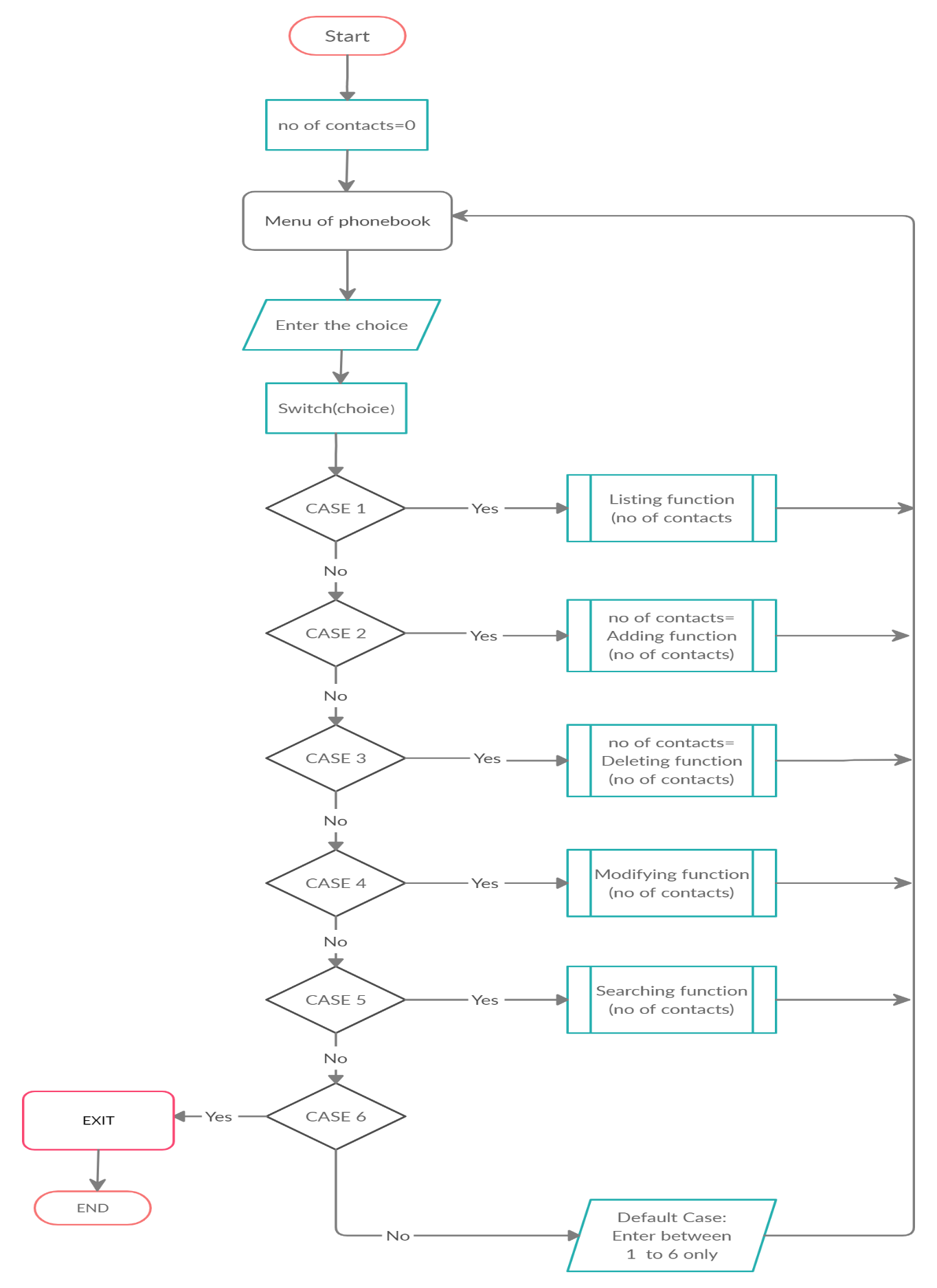
With the existing 5 functions, some extra functions are defined to optimize the code.

Searches function: it takes the input parameter as name and total no of contacts. The whole database is searched for the contact with similar name. If it is found then it returns the index of the contact else return -1.

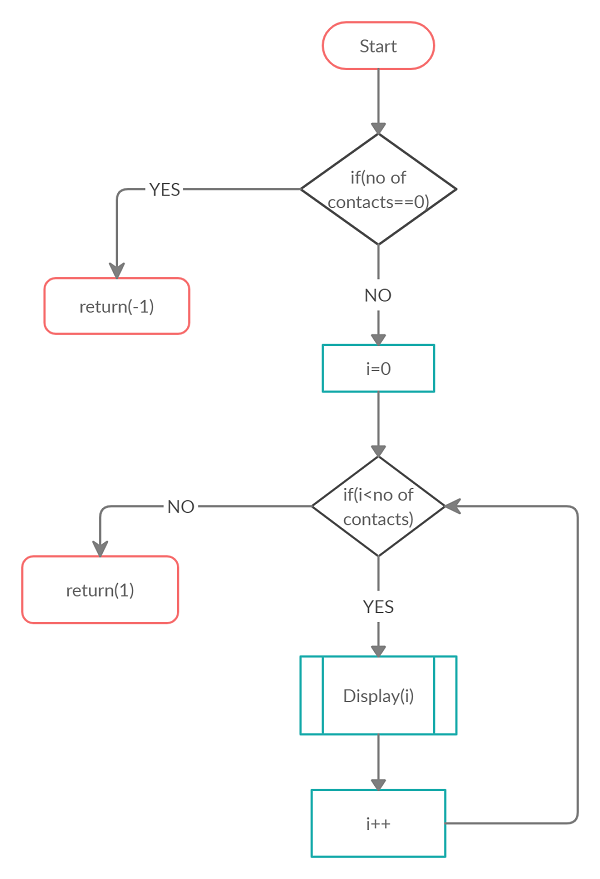
Enter function: it takes the input as index of the contact details to be entered.it takes the contact details from the user and stores it in the particular index. Contact details like name, phone number, gender, email ID are entered by the user.

Display function: it takes the index as the input parameter. The contact details name, phone number, gender, email id located at index are displayed.

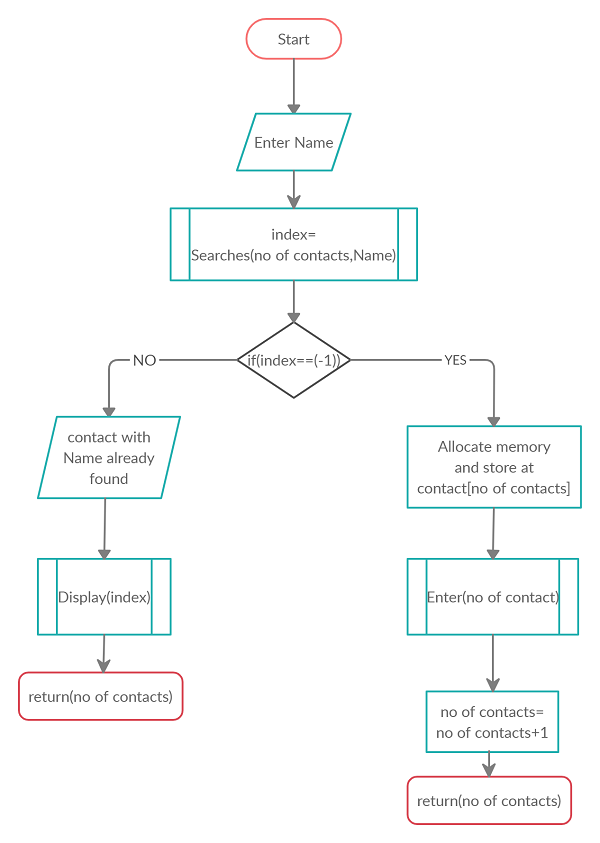
A simple flowchart of the phonebook menu is shown below.



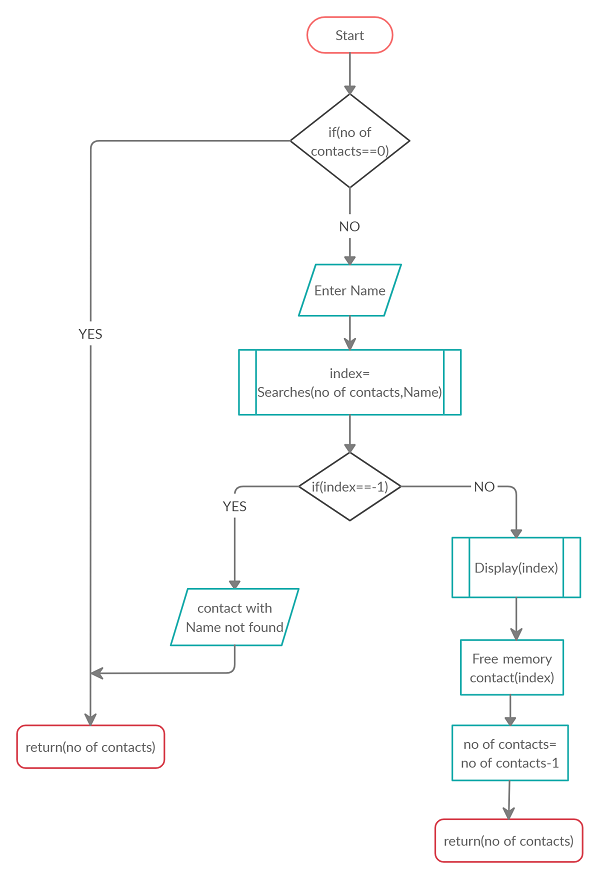
Flow chart of listing function



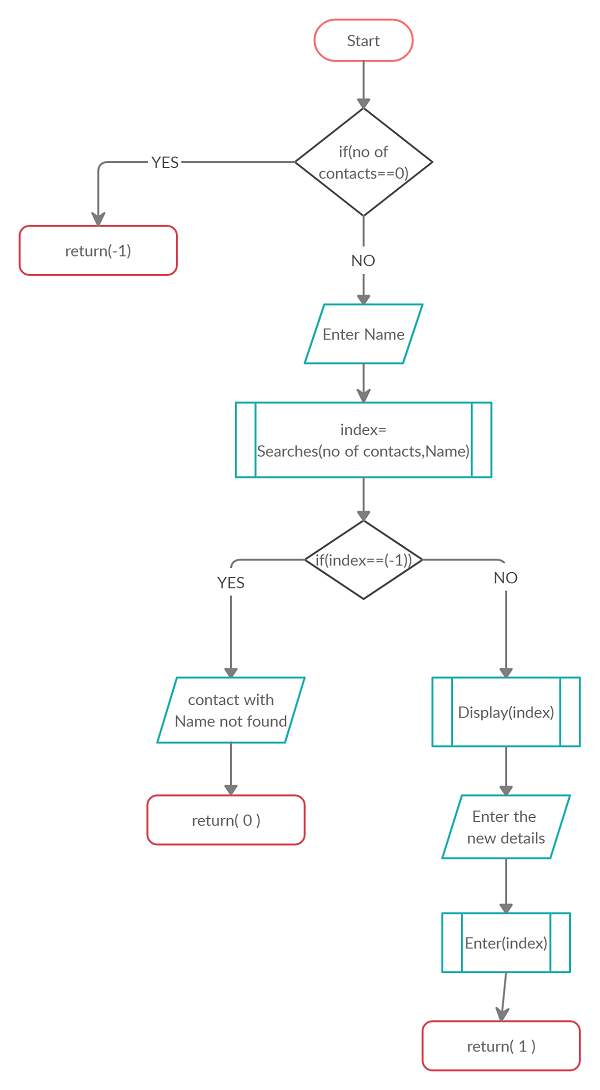
Flow chart of adding function



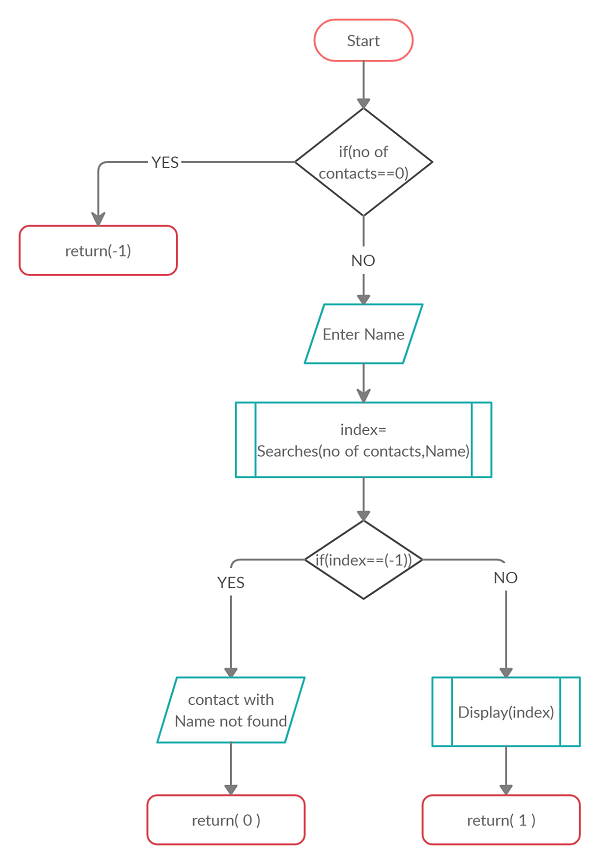
Flow chart of deleting function



Flow chart of modifying function



Flow chart of searching function



Test plan

Testing involves the design of test cases that validate that the internal program logic of the phonebook application is functioning properly. Boundary conditions of the functions are tested.

Testing is done on functions add, delete list, modify and search.

Test cases

Testing of adding function->

1) Test case 1-: To add the record in the database with the same name.

Expected results: Record should not be added and display message record already exists.

Testing of deleting function->

2) Test case 2:- To remove the record which does not exist in the database.

Expected results: The program should display records that cannot be deleted as it does not exist.

Testing of searching function->

3) Test case 3:- To search the record which does not exist in the database.

Expected results: The program should display no records found in the database.

Testing of listing function->

4) Test case 4:- To list the records without adding any of the records.

Expected results: Display no records found.

Testing of modifying function->

5) Test case 5:- To modify the record which does not exist in database.

Expected results: Display records cannot be modified as they do not exist.

Expected output

The complete phonebook system which can store the contact details and manage it according to the user. The program makes it easier for the user to connect to his contacts.

The personal information and family information of the contact are stored in a single record, which will allow users to easily search and find the desired contact. The program handles the four operations of adding, deleting, modifying, and searching according to user selection. Each operation is performed as a separate function, so the control enters a different function, and all data added or modified or deleted will be stored in the array of objects of type class contact.

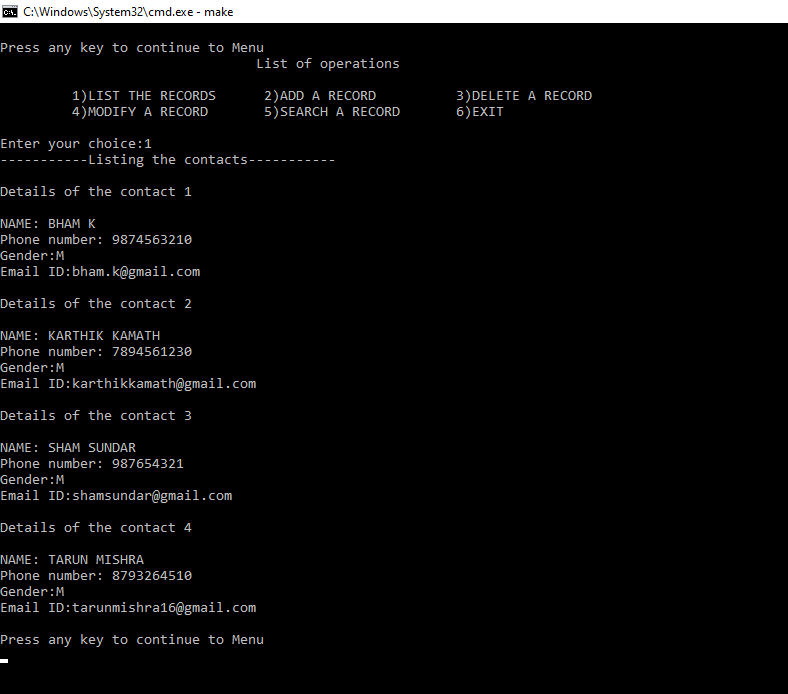


Fig: Phonebook program