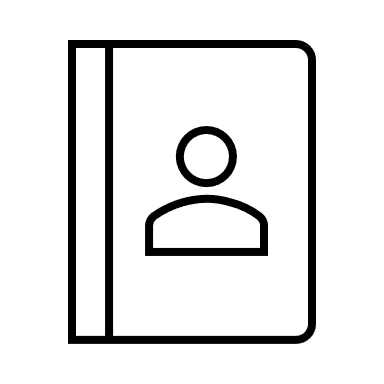
|  |
| --- |
| Contacts  Book  Project  Documentation Report   * By Marwan Tamer |
|  |
| March 13, 2023 |

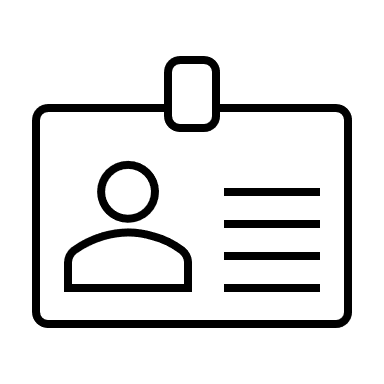
# *Address Book with solid fill*

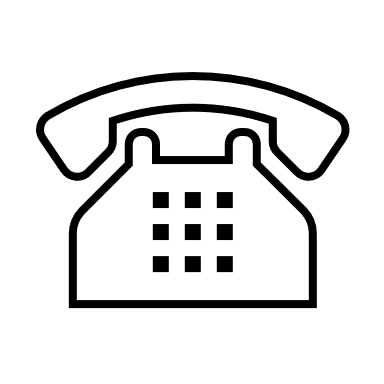
**What is the Contacts Book project?**

*The Contacts book project is a software product that is simply a software that can manage contacts and their information like phone numbers and their types, place of livelihood, age, notes, and so on…*

* *The project was built on gained knowledge in OOP (Object Oriented Programming) and Structured Programming methodologies and was coded in C++ language.*
* *The used skills in this project were:   
  1. Problem Solving Skills.  
  2. Knowledge of Software Engineering concepts.  
  3. Encapsulation.  
  4. Knowledge of dynamic memory allocation and Heap data structure.*

*The program is made of three Classes which are:  
1. ContactsBook{}*

*2. Contact{}*

*3. PhoneNumber{}*

*[1] PhoneNumber Class is the base of this class sequence as it’s the first class implemented because it will be composed/controlled by other functionalities in the other classes so it consists of:*

*(1). Two private data types of string type to store the the number and the type of that number.*

*(2). Three Functions which are:*

*1. void SetNumber(string, string) which takes two parameters and it’s used to assign the arguments passed to it to the private data types in the class.*

*2. void Show() which takes no parameters and return nothing it’s used to just show the information stored in the private data types which are number and the type.*

*3. bool found(string key) const which takes a parameter of string type and returns a Boolean either true or false and does not manipulate the private data members it’s used to match the argument passed to the stored string in the private data member to see if they are matched or not, the use of this function will be more obvious in the upcoming definitions.*

**Implementation**

1. void User() which has no parameters and no return type it’s used to just let the user enter the main information of the contact the user added.
2. void Show() which also has no parameters and no return type used to display the main information of the contact that was added previously

*(3). Two constructors one is default has no parameters and one is parameterized that has two parameters and the parameterized one is for creating the object and also setting the number information by SetNumber() function called in it.*

*[2] Contact class is the second class implemented or let’s say level 2 of our hierarchy now it’s the most important class to me because it contains the neural system of the software as we will see later in the functionalities below, so the class consists of:*

*(1). 4 private int data types*

*5 private string data types*

*1 private time\_t data type*

*1 private PhoneNumber data type*

*(2). 4 Functions which are:*

3. int GetUserId() which is simply used to return the user id only.

4. bool Search(string key) which takes a string as a parameter and it’s used to match the string with the contact first name and last name and match its phone number that is stored in an array of PhoneNumber.

(3). One constructor and one de-constructor a default constructor that initializes the variables and assign a dynamic memory location to a new PhoneNumber array of objects. De-constructor delete the memory allocation of that array to prevent memory leaks.

[3] ContactsBook class is the last class implemented in terms of order and it’s last level of our hierarchy and its functions will be the mainly interfaced functions in the main source file of the software, it consists of:

(1). 3 private data members ( two of type int and one of type Contact\*) and 5 public function members which are:

1. void AddContact() which is used to order the program to allocate a new location in the dynamic memory for a new Contact data type to create a new object from that data type and start to enter its information and store it in an array of objects.
2. void ShowAll() which is used to show all the contacts existing and stored in the memory.
3. void DelContact(int) which is used to iterate over the array of objects of contacts that has all the contacts and

their info stored in it and then delete/deallocate it by a simple specific algorithm used.

1. void EditContact() which is same as DelContact() function but instead of deleting it edits an existing contact.
2. void FindAll() which is used to iterate through the array in which the contacts are stored and match the search operation.

(2). A Constructor and a De-constructor, the constructor is used to create an array of objects (Contacts) and to initialize the size of that array and the count.

Testing

1. Tested the AddContact() method to make sure it actually did generate new objects in memory and allocate a new location as it was supposed to.
2. Tested EditContact() and DelContact() using the same procedure, determine whether the feature will operate as expected and whether the contact will be deleted or edited if it already exists by passing the same input to both procedures and if not how it will be handled.

1. Tested the ShowAll() function and observe the output if there are no contacts to show and if there are contacts to show.
2. Tested all the constructors through all the mainly interfaced functions and how they handle creating new objects and how they allocate new memory.

**In conclusion, the successful development of the contacts book software project demonstrates the importance of effective planning, collaboration, and execution. The software's user-friendly interface and powerful features have made it a valuable tool for organizing and managing contacts, improving productivity and efficiency for users.**

Conclusion