Contact Management System

Intro to C Programming

CSCI-1110-02

OWLS



University of New Haven

TAGLIATELA COLLEGE OF ENGINEERING, West Haven, CT

Submitted to:

Dr. Reza Sadeghi

Spring 2021

**Final Project Report of Contact Management System**

**Team Name** **OWLS**

**Team Members**

1. Franco Lituma [flitu1@unh.newhaven.edu](about:blank) (Team Head)

2. Lisanne Price [lpric1@unh.newhaven.edu](about:blank) (Team Member)

3. Thomas Arcati [tarca2@unh.newhaven.edu](about:blank) (Team Member)

**Roles of Team Members**

**1. Franco Lituma:** Team head, accomplished the task of encryption and decryption, created the capabilities of each administrator and the general user including the accessibility and the restrictions . Aided in creating the username and password prompt for each user. Aided in giving admin access to certain task and debug the code.

**2.** **Lisanne Price:** Report and presentation documenter and editor, accomplished the task of login information and welcome page, the menu for the enlisted functions necessary for the system, warning message if a user made an error, aid in testing and debugging the code, and aid in making the Contact Management System a more user friendly.

**3. Thomas Arcati:** Presenter, accomplished the task of editing task events and details, aided in how the system would store and save data for each existing user.

**Table of Contents**

Table of Figures……………………………………………………………….4

Table of Tables………………………………………………………………..5

Introduction……………………………………………………………………6

Project descriptions…………………………………………………………....7

List of key variables…………………………………………………………..9

List of functions………………………………………………………………10

Admin functions…………………………………………………………........11

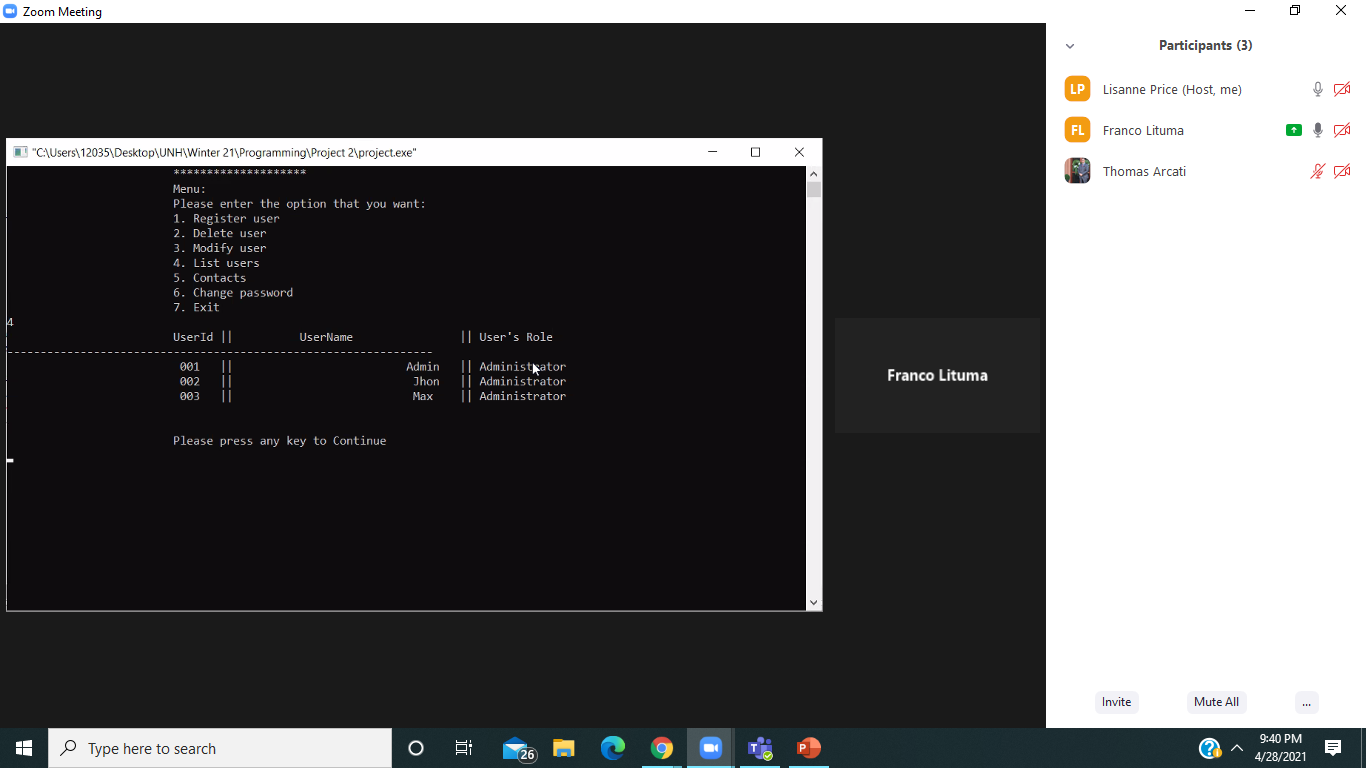
Guess functions………………………………………………………….........12

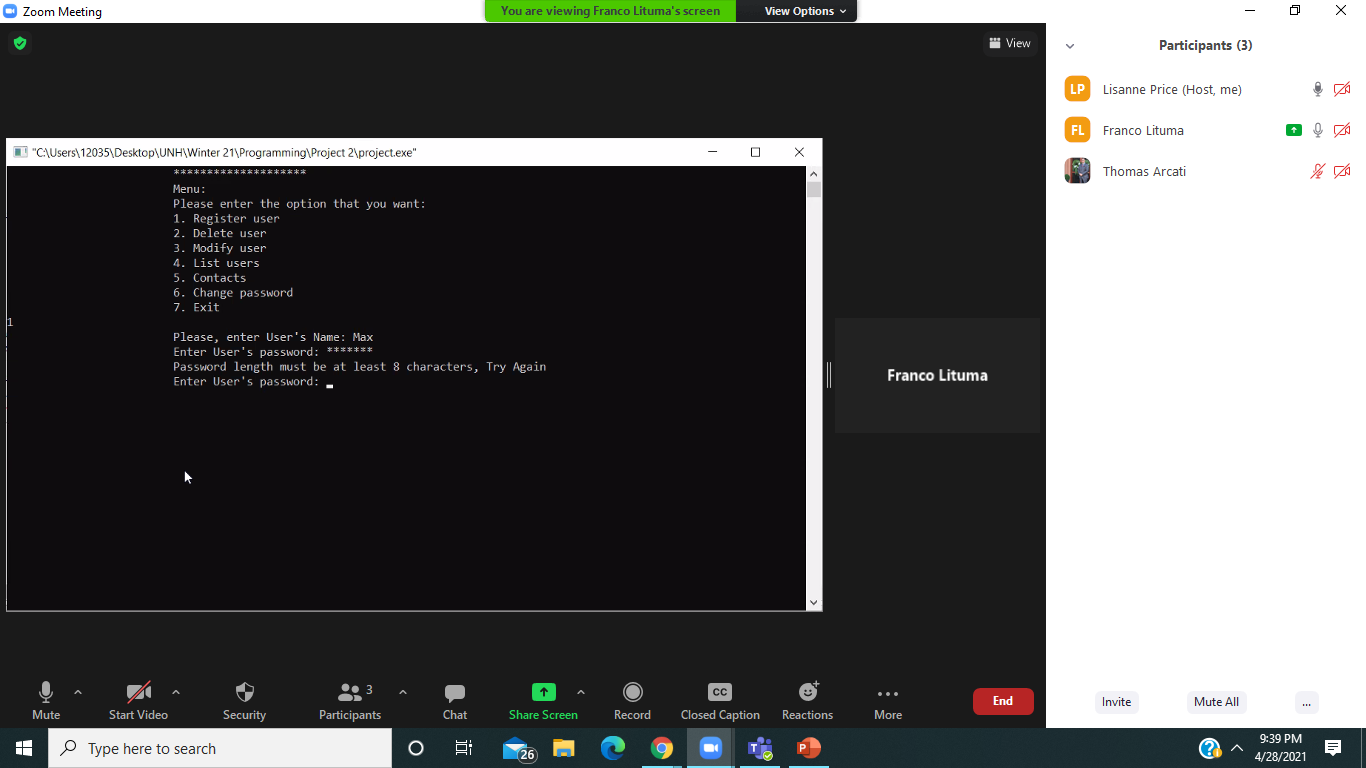
References…………………………………………………………………….13

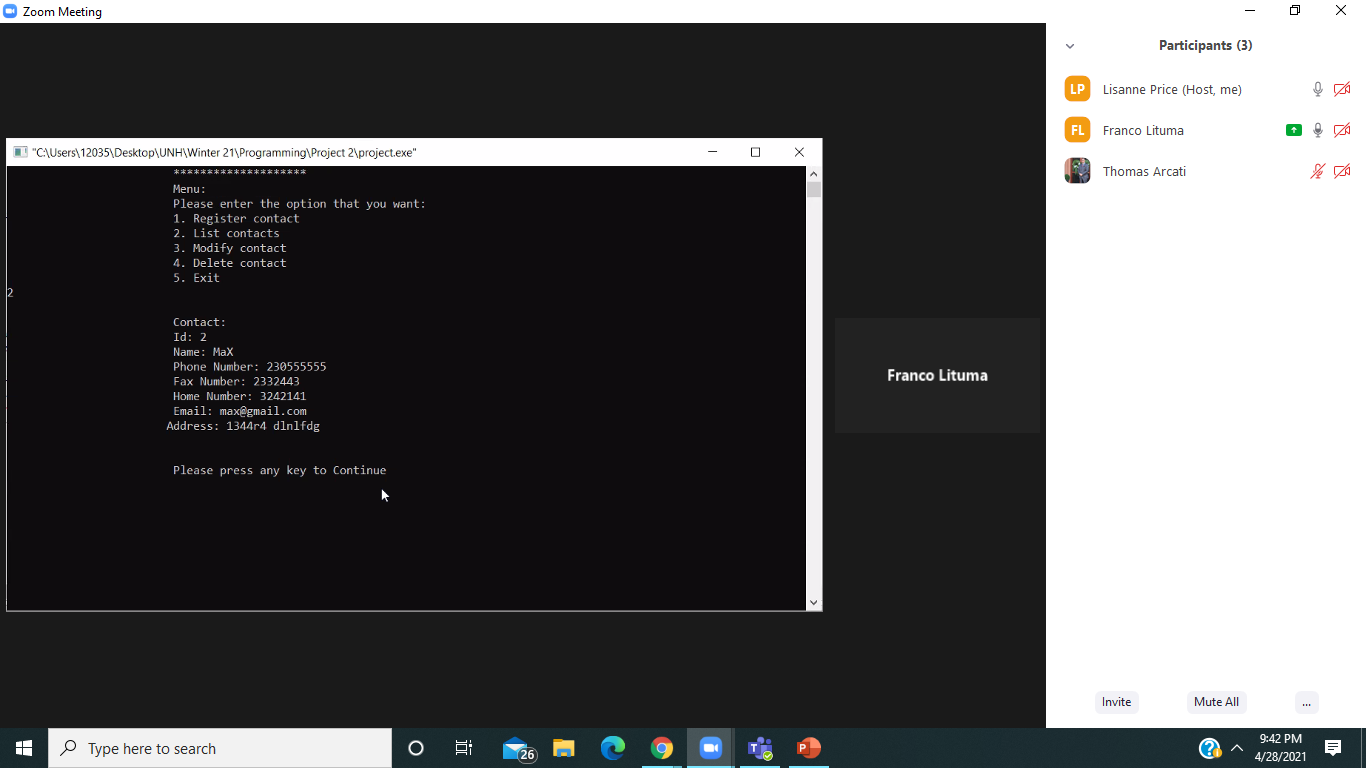
**Table of Figures**

Text

Description automatically generated **Fig. 1 Welcome to the CMs.**

  
**Fig. 2 Main menu with the list of user.**

  
**Fig. 3 Main menu and message to enter a new user.**

  
**Fig. 4 Secondary menu and the list of contacts .**

**Table of tables**

|  |  |
| --- | --- |
| readPassword | Reads the inputted password from a list of previously inputted passwords |
| iterU | Iterates or repeats the elements within a structure and function throughout the program. |
| adminMenu | Brings the program to the Menu set for only admin access |
| ciphuserpassword | Encrypts the users password in order to easily store said password |
| caesarcipher | Allows letters of passwords to be changed into readable intergers |
| loadusers | Provides a list of current inputted users for admin control |
| strNonciph | A string of characters used to decipher encrypted information. |
| usertype | Differentiates between admin and regular users in order to allow correct menu be used. |
| Registeruser | Records Users information inputted by the admin |
| userid | Is the location the user enters their id in order to login to the user interface |

|  |  |
| --- | --- |
| char email | Allows for the input of a contacts email address |
| char address | Allows for the input of a contacts address |
| intcontact Id | Allows for quick selection of contact in list of contact |
| intuserId | Allows for quick selection of user in list of users |
| CharuserName | Allows for input of the user’s username |
| char user Password | Allows for input of the user’s password |
| int userType | Allows for quick designation of user type, be it admin or user |
| charcontactName | Allows for input of the contact’s name wished to be stored |
| charphoneNumber | Allows for input of the contact’s phone number wished to be stored |
| charfaxNumber | Allows for input of the contact’s fax number wished to be stored |
| charhomeNumber | Allows for input of the contact’s home number wished to be stored |
| char readString[200] | Allows for the string to be read in order to decipher encrypted information. |
| FILE \*usersFile | Stores information in a separate file for users |
| char fileName[50] | Allows input of a name for the file |

**Introduction**

The Contact Management System, CMS is used for recording contact details inputted by different users including cell phone numbers, fax numbers, home numbers, and email addresses. The system was created in such a way that administrators and users have access to the system by a login system with their individual username and password. The system possesses the capability to make changes to any existing user by the administrator such as adding and deleting the user, and changing passwords. Both the administrator and user have the ability to make changes to contacts including add, delete, search or modify any contact information in the system. Administrators are presented with more privileges in terms of accessing a wider array of information and control of the system leaving the basic user with only contact information control.

Tempser.userId shows if there were different users, the new one will be added to a new position to make it unique. addUser saves the user within the list. If a user like this predefined doesn't exists, the Administrator creates a new user. Malloc locates in memory and reads the user from the temporary list to write to the user list. It is printed that a new user was created correctly, and we go to the saveusersFile function. The Contact Management System is very beneficial, as it provides a platform for users to store, and organize contact data that is readily accessible, which in return increases productivity and improves data management.

**Project Description**

The Contact Management System greets its user with a welcome page and a menu of all the functions that are applicable to a specific user. At that point in time, the user will be prompted to login with their respective username and password as an avid user or administrator. The structure stores the users in the system, and this explains the user structure. It has its own information about the user and several contacts. With the code, the userType variable allows us to identify as an administrator user who has the option of adding, modifying, or deleting contacts within this structure. On the other side, if you are not an administrator, it will prompt the user to just add the contacts. If they do not have an account within the system, they will be prompted to create one. The administrator at this point, is able to add a user to the Contact System, by creating a new username and password for that new user. The administrator also possesses the ability to remove any user’s account or information as they deem it fit.

When the respective users have logged in, a generated list of commands will appear to prompt the user which varies based on the type of user is logging in. The program has a default username and password, and the readPassword function is accessed to verify if the entered password is correct. The respective comparison is made within this function, and access to encrypt the username and password of the administrator is present for privacy protection. If the comparison of these encrypted functions is correct, the initial menu is presented. This menu allows the user to enter an Administrator or User type user based off the users’ credentials within the system. It allows administrators properly using the system to delete, modify and review the list of users. This also allows both administrators to load the contacts of each user that has been present in the system. This menu allows administrators and users to add contacts within the system. The system gives the administrator and users access to the contact list. After logging in, the various commands present will prompt the admin or user to add contact detail information such as: First name, Last name, cell phone, workplace phone, fax number, email address, gender, and age. The admin or user will also have the option of removing, editing or searching contact record details. On the other hand, the guest user may only be prompted to edit their contact details and information.

Since the system is readily accessible to users other than administrators, the Contact Management System was designed in a manner to be predominantly user friendly. It greets the user with a welcome page and gives an alert/warning message if a user tries to put a contact information with a name that does not exist. To ensure that each user has the best experience possible, a menu of all functions that are readily available to that particular user will be easily navigable and reports are generated in a tabular form for the simplicity of its users. If the user does not enter a correct option in the menu, the screen will show a message saying “Please select a valid option”. The menu presents an option to change the password of the contact, and finally the user can return to the main menu. If the person does not enter a correct option at this point, the program shows the menu again. Once users navigate their way through the system, all the information needed will be accessible to them, and once they are through, they will be prompted to exit and logout the system.

**List of Key Variables**

**Variables Name Description**

|  |  |
| --- | --- |
| char email | Allows for the input of a contacts email address |
| char address | Allows for the input of a contacts address |
| intcontact Id | Allows for quick selection of contact in list of contact |
| intuserId | Allows for quick selection of user in list of users |
| CharuserName | Allows for input of the user’s username |
| char user Password | Allows for input of the user’s password |
| int userType | Allows for quick designation of user type, be it admin or user |
| charcontactName | Allows for input of the contact’s name wished to be stored |
| charphoneNumber | Allows for input of the contact’s phone number wished to be stored |
| charfaxNumber | Allows for input of the contact’s fax number wished to be stored |
| charhomeNumber | Allows for input of the contact’s home number wished to be stored |
| char readString[200] | Allows for the string to be read in order to decipher encrypted information. |
| FILE \*usersFile | Stores information in a separate file for users |
| char fileName[50] | Allows input of a name for the file |

The list of variables provides, are crucial and important for the generation and operation of the Contact Management System. It is used for the login and main menu of the system.

**List of Functions**

**Administrator and Guest Functions**

**Function Description**

|  |  |
| --- | --- |
| readPassword | Reads the inputted password from a list of previously inputted passwords |
| iterU | Iterates or repeats the elements within a structure and function throughout the program. |
| adminMenu | Brings the program to the Menu set for only admin access. |
| ciphuserpassword | Encrypts the user’s password in order to easily store said password. |
| caesarcipher | Allows letters of passwords to be changed into readable integers. |
| loadusers | Provides a list of current inputted users for admin control. |
| strNonciph | A string of characters used to decipher encrypted information. |
| usertype | Differentiates between admin and regular users in order to allow correct menu be used. |
| Registeruser | Records Users information inputted by the admin |
| userid | Is the location the user enters their id in order to login to the user interface |

**Findings**

Based on the extensive research done for this project, the purpose was to record varied information that was securely inputted by users in the Contact Management System. The system prompted users to store information in secured text files and made it readily accessible to the user when needed. The system was predominately created to have contact information for respective personnel and for the authorized individuals to process, use, and navigate through this information. To accomplish such a task, we needed external resources of information that aided in mastering the concept of encryption and decryption in specified parts of C programming. Based on the lecture material that was covered, and these external resources, the group developed the Contact Management System, and this gave each member a better understanding of how to generally make a better contribution to the contact system.

**References**

[https://www.youtube.com/watch?v=kmq3qS72kcg](about:blank)

[https://blog.hubspot.com/sales/contact-management-software](about:blank)

Professor Reza Sadeghi

[https://www.youtube.com/watch?v=DsYKagOoap0](about:blank)