PURBANCHAL UNIVERSITY

GOMENDRA MULTIPLE COLLEGE

CONTACT MANAGEMENT SYSTEM

IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF
[BCA - I]

Submitted by

BIRAJ MAINALI : [**313665**]
HEM BAHADUR BASET : [**313641**]
SUBARNA THAPA : [**313672**]

Under the supervision of
[**MR. BHAWESH KAFLE**]

[10] [2078]

Acknowledgement

According to the course of study of BCA I semester determined by P.U., a computer project is to be carried out for the partial fulfillment of the requirements for the Bachelor in Computer Application. Therefore, as students of the course, we have developed a small desktop application “Contact Management System".

We feel very glad for getting such an opportunity to accomplish the BCA First semester project and feeling the experience of the team environment and team spirit. This gave us insight knowledge about the practical aspect of the various stages and procedures of software development projects.

This report consists of an explanation of the requirements and design of the new systems. The requirements are stated in forms such as natural languages to provide a clear picture of each requirement. We are very thankful to Gomendra Multiple College for facilitating the competition of the entire project work.

First of all, we would like to thank our superior, Mr. Bhawesh Kafle for his valuable guidance and encouragement throughout the project. We also thank him for helping us out in difficulties by showing the right way and for the valuable suggestions which were very crucial for the success of our project our sincere gratitude goes to all the faculty members of the department of Computer Application for their encouraging support and guidance.

Finally, we would like to thank all our friends for their constant support, continuous encouragement, and valuable suggestions throughout the duration of the project.

Introduction

Contact management system has been made to record basic contact information, Contact details include Name, number, address, email as we can mark contacts as favorite and it includes the created date and created by. That gives a better understanding of contact.

Also, it has an authentication system so we can track who is creating which contact and when also If the user is not registered to the system then s/he will not be able to create or add a new contact and modify accordingly.

The main feature that is included in this application is as follows

- Add new contact information
- View all recorded contact - A unique contact number.
- Search added contact by contact name
- Update contact information
- Remove contact by contact number
- Register User - Unique username required
- Mark as favorite later if required.
- Current user contact report

2. Technical Information

1. Langauge C
2. File Handling
3. Pointers
4. Structures
5. Conditional Infromation
6. Interaction statement
7. Functions

Project Details

1. Technical expect of function

01. **FileProvider**: This method is for ensuring the file is ready to use. If single *fopen*(file pointer method access local files) could not open the file we need to validate created file pointer is not null which makes the code more messy and dirty too. So we made a file provider which accepts two-argument filename and mode. The arguments are determining which file to open and what to do. For more accuracy, we have made some constants such as file mode and file name.

02. **FindByContact**: This method is the most used and key method of this program. It returns structure a contact structure that also accepts type long contact number, And it finds the first matching contact number and returns it as a Contact structure with a valid flag which is a Boolean type, by default contact has a true flag, if contact is not found then we return false that help to ensure either matching contact found or not

03. **Dashboard**: Basically dashboard contains main activities such as view, create, search, update and remove

04. **CreateContactDetail**: Create method ask for required information that is required to save a contact, basically we are using contact structure to store and manipulate the contact. It returns void.

05. **Display**: The display method plays a role to make a better UI it adds a Page banner that gives a clear vision to the user of what s/he actually doing for, it accepts a set of characters and displays it

06. **UpdateContactDetails**: Basically update method is used for updating existing contact details, first thing first it asks for a contact number that is unique and we can validate if it exists or not, if exist it allows us to update further information.

07. **ViewContactList**: The view method is to show a list of added contact, it accepts one argument that is "queryFirstName" which allow us to filter the contact list by the first name. The noted thing is if the existing contact first name equals to query string then the all matching contact will appear. If "queryFirstName has no value it means no value and it list all the contact. And if it has value filters according to value and lists it. After all, it returns void

08. **ViewContactList**: The view method is to show a list of added contact, it accepts one argument that is "queryFirstName" which allow us to filter the contact list by the first name. The noted thing is if the existing contact first name equals to query string then the all matching contact will appear. If "queryFirstName has no value it means no value and it list all the contact. And if it has value filters according to value and lists it. After all, it returns void

09. **CurrentUserContacts**: This method lists all recorded contact whose contact is recorded by the current login user

10. **RemoveContact**: Basically removed method used to remove existing contact. It accepts one argument which is contact number. If provided contact number exists in the local file it returns 1 else -1 so the Remove function returns the integer, the main reason for making this function is for reusability. However, we provide a contact number it ensures the existence and removes it also return a specific integer so we can ensure removing.

11. **SaveChanges**: Basically this method accepts the one argument which is a structure named "contact" that ensures passing argument contact named variable contact Number is existing, it means duplicate and it returns -1. If not, it returns 1 which means success also assigns Id to a unique random number that helps to ensure to find and validate unique contact. Finally, it returns type integer

12. **MaskAsFavourite**: Removed method accepts one argument that is the existing contact number. It updates the Favorite flag to true. That determines the existing contact is now web favorite symbolled it while listing added contact.

13. **EnsureUniqueContact**: As we named it ensure we are not saving duplicate contact.it accepts contact numbers and checks from a file if found existing contact number which is equal to the contact number it returns 1 else 0 so we can check heck through the contact number. For e.g.

```
if(EnsureUniqueContact(contact) ==1) {
 // Perform related task.
}
```

14. **RegisterUser**: This method asks the user for required information that is required to authenticate users such as username and password

15. **EnsureUniqueUserName**: This method is made to prevent unique usernames. That helps to easy for login and more efficient authentication. It accepts parameter username and if matching contact number found it navigate to Register User method.

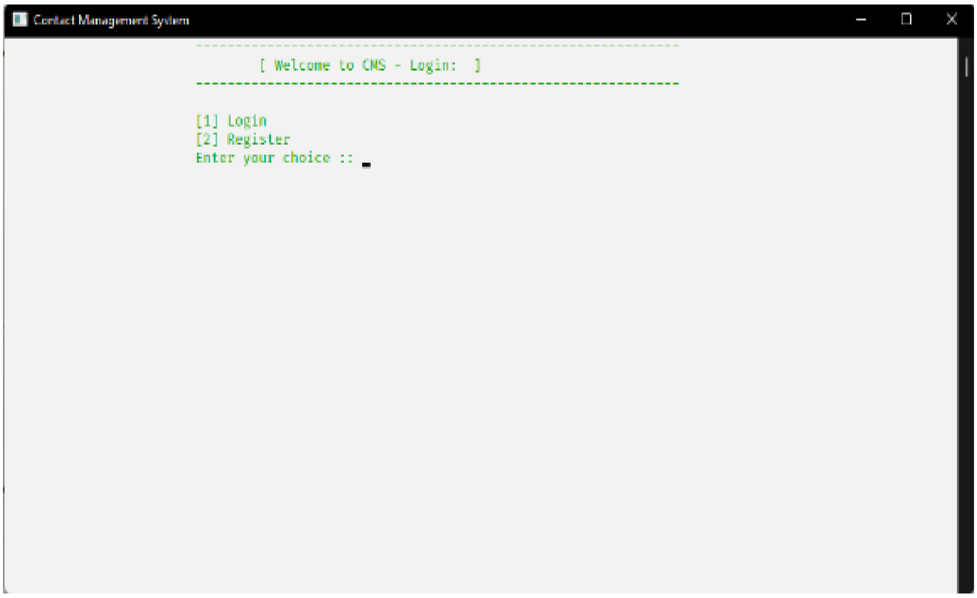
16. **Login**: Helps to authenticate user, it basically checks if the user exists on a local file named "auth_store". It is just found by username and password and it navigates to the dashboard where we can perform actual contact management tasks.

17. **CreateInitialStore**: Basically, this method return void, it is just created to create initial files therefore it is not created it creates the file, and we won't face error like "*segment core dumped error*"

18. **Next**: For this method, we can be called it a navigation method. It accepts one keypress if pressed key is [Enter] then is redirect it to the dashboard otherwise close the program.

2. Application Level overview

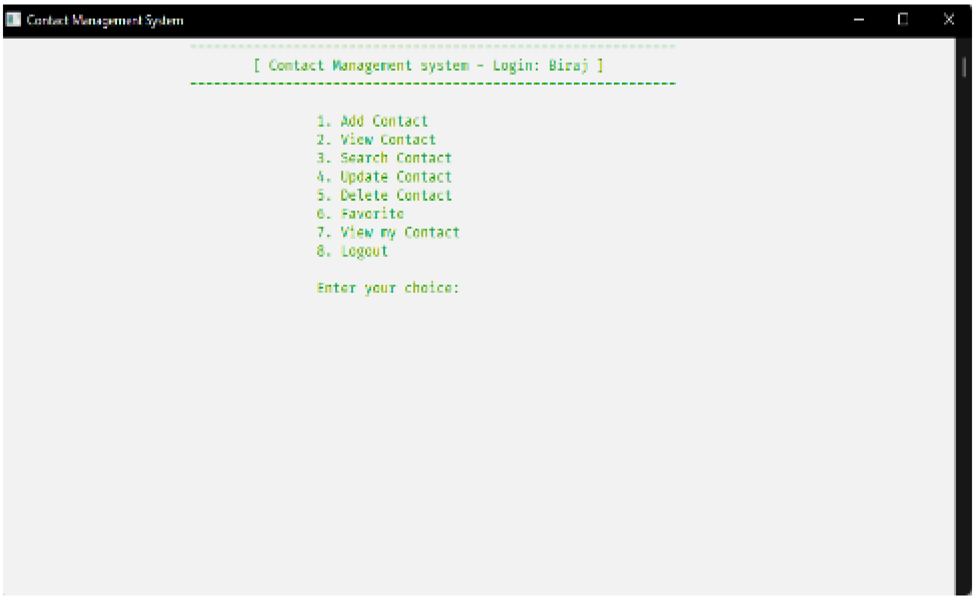
Authentication window



```
--------------------------------------------------------------
        [ Welcome to CMS - Login:  ]
--------------------------------------------------------------

[1] Login
[2] Register
Enter your choice :: _
```

Dashboard window

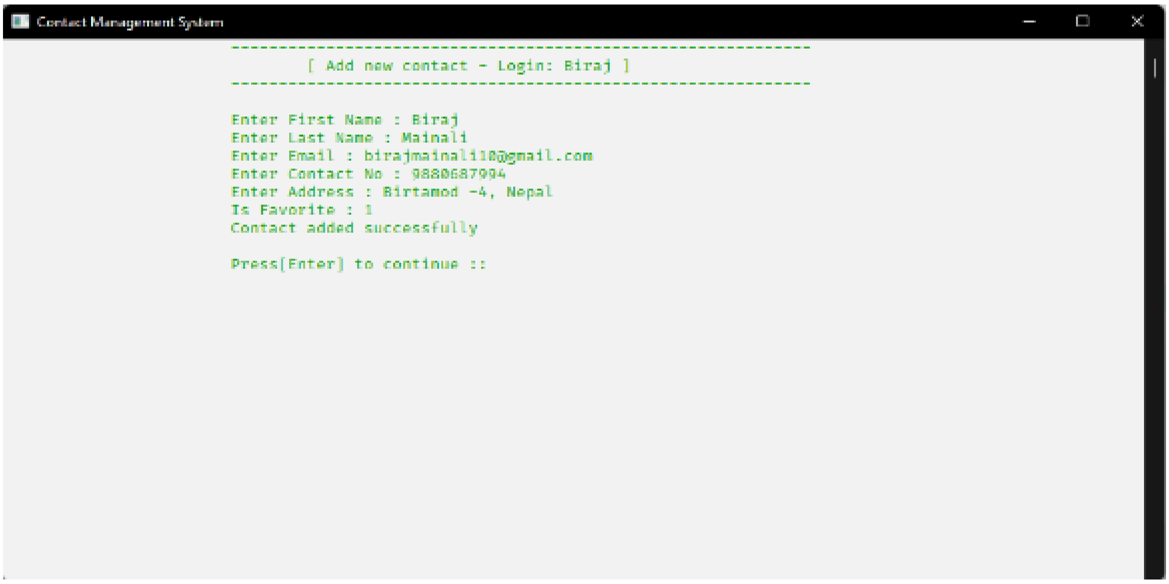


```
--------------------------------------------------------------
        [ Contact Management system - Login: Biraj ]
--------------------------------------------------------------

                1. Add Contact
                2. View Contact
                3. Search Contact
                4. Update Contact
                5. Delete Contact
                6. Favorite
                7. View my Contact
                8. Logout

                Enter your choice:
```

New contact window



```
-------------------------------------------------------------
        [ Add new contact - Login: Biraj ]
-------------------------------------------------------------

Enter First Name : Biraj
Enter Last Name : Mainali
Enter Email : birajmainali10@gmail.com
Enter Contact No : 9880687994
Enter Address : Birtamod -4, Nepal
Is Favorite : 1
Contact added successfully

Press[Enter] to continue ::
```

3. Hardware-software Requirement

- **Windows:** 7 or newer
- **MAC:** OS X v10.7 or higher
- **Linux:** Ubuntu

Hardware Requirements

We strongly recommend a computer with fewer than 5 years old.

- Processor: Minimum 1 GHz; Recommended 2GHz or more
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above

4. Group Information

1. Time Scheduling

Student Name: Biraj Mainali

S.N	Job description	Date from	Date to	Reference
1	Algorithm	10th January	17th January	
2	Coding	16th January	18th January	
3	Limitation and scope of improvement	17th January	20th January	
4	Review and correction	21st January	26th January	Discussion with group members

Student Name: Subarna Thapa

S.N	Job description	Date from	Date to	Reference
1	Introduction to our work	10th January	17th January	
2	Requirement analysis for Contact Management	16th January	18th January	
3	Algorithm / Coding	17th January	20th January	Discussion with group members
1	Sample coding	21st January	26th January	

Student Name: Hem Bahadur Basnet

S.N	Job description	Date from	Date to	Reference
1	Introduction to our work	10th January	17th January	
2	Requirement analysis and coding	16th January	18th January	
3	Algorithm / Coding	17th January	20th January	Discussion with group members
1	Sample coding	21st January	26th January	

Conclusion

We hope that the experience we gained while making the “Contact Management System” will be beneficial for us in the coming days. As we are only beginners, there are a lot of areas where we need to improve and this project has helped us in a big way.

We are thankful to the management of our college for providing the necessary resources. Without their cooperation, it would not have been possible to complete the project.

We would like to express our gratitude to all friends and teachers who helped us by providing valuable suggestions to make the project better.

The mistakes we made during the making of this project have taught us various new things about C and implement Logic. We hope we can create better programs in the Future or Upgrade this program to do Other Additional Works.

Finally thank you to all individuals who have assisted and encouraged us in the development of this project.