



**NEW HORIZON
COLLEGE OF ENGINEERING**

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

A MINI PROJECT REPORT ON

PASSWORD HASHING

*Submitted in fulfilment for the award of the
degree of Bachelor of Engineering*

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

PREKSHA SHRIDHAR

1NH17CS102

'B' SECTION

Reviewed by

Ms. Priti B Badar

Sr. Asst professor

Department of Computer Science Engineering



**NEW HORIZON
COLLEGE OF ENGINEERING**

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

Certificate

This is to certify that the mini project work titled

Password Hashing

*Submitted in fulfilment for the award of the degree of Bachelor of
Engineering*

**Preksha Shridhar
1NH17CS102**

*During the academic year
2018-2019*

**Signature of
Reviewer**

**Signature of
HOD**

Semester End Examination

Name of the Examiner

Signature with date

1.

.....

2.

.....

ABSTRACT

Main object of the project is to implement features of object oriented programming. Abstraction, encapsulation, class, object, function overloading are the features used to implement the program.

The user sign up using his details giving his email-id and the password. After signing up it throws a unique number by which the user can try to find his account. The admin logs in giving correct password and admin can view all the user who are signed up with their password being hashed.

1. User sign up giving his name email-id details and password.
2. Throws a unique number of the account.
3. User can try to find his account using the unique number which was given.
4. Admin logs in using the correct password.
5. All the mail-id who are signed up will be displayed with their password being hashed.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned my efforts with success.

I thank the management, **Dr. Mohan Manghnani**, Chairman of NEW HORIZON EDUCATIONAL INSTITUTIONS for providing the necessary infrastructure and creating a good environment.

I also record here the constant encouragement and facilities extended to me by **Dr. Manjunatha**, Principal, NHCE, **Dr. Prashanth.C.S.R**, Dean Academics, and **Dr. B. Rajalakshmi**, Head of the Department of Computer Science and Engineering. I extend my sincere gratitude to them.

I also express my gratitude to **Ms. Priti B Badar** Sr. Asst professor, my project reviewer for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Finally a note of thanks to all the teaching and non-teaching staff of Computer Science and Engineering Department for their cooperation extended to me and my friends, who helped me directly or indirectly in the course of the project work.

PREKSHA SHRIDHAR

1NH17CS102

TABLE OF CONTENTS

Chapter	Topic	Page Number
1	Introduction	1
1.2	Problem Definition	2
1.3	Outcome	2
2	Object oriented features	3
2.1	Objects	3
2.2	Class	3
2.3	Data abstraction	3
2.4	Polymorphism	4
2.5	Encapsulation	5
2.6	Inheritance	5
3	Requirements & Design	7
3.1	Requirements	7
3.2	Algorithm	7
3.3	Flow chart	10
3.4	Class diagram	12
3.5	Data flow diagram	13
4	Implementation	14
5	Output Snapshots	21
6	Conclusion	25
7	Reference	26

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

C was developed by Dennis Ritchie and C++ was developed by Bjarne Stroustrup. When compared C with C++, C is the subset of C++. C contains 32 keyword and C++ contains 52 keywords. C doesn't support object-oriented programming, but C++ being an object-oriented programming language, it supports polymorphism, encapsulation, and inheritance. For the development of code, C supports procedural programming and C++ is known as hybrid language because C++ supports both procedural and object-oriented programming paradigms.

Project is more effectively implemented in C++ rather than in C. because C++ is widely used when higher level languages are not efficient. Code using C++ is much faster compared to C code, which makes it a better solution for applications where performance is important.

With C++, you can code for any platform like Mac, Windows and Linux. C++ programmers generally focuses on applications that work directly with hardware or that need better performance than other languages can offer. C++ will let us do almost anything except for the syntax is right. It's a flexible language.

The main object is to demonstrate object-oriented concepts which is classes, objects, inheritance, polymorphism.

- User sign in with their email-id and their password.
- Password is hashed and stored.
- User can search their account by typing the reference number provided during sign in.
- Admin logs in and searches for the details of users, admin views the hashed password.

1.2 PROBLEM DEFINITION

For security reasons, we may want to store password in hashed form. This guards against the possibility that someone who gains unauthorized access to the database can retrieve the password of every user in the system. The project is done using object-oriented program in c++ to implement the technique of password hashing which is used to hash the password given by the user while logging in. In this program the admin is not authorized to view the password of the employees. This program is implemented using abstraction, encapsulation and inheritance.

The choice 1,2,3 that is signing up, finding account, Admin login are provided accordingly.

When the choice signing up is chosen, the user has to log in with their e-mail id and password. Entering this will provide a unique code to the account.

For the second option, that is to find the account. User can find their own account by entering the unique number which was provided during the sign-in. the 3rd option that is Admin login, Admin logs in by entering the correct password have the access to look at the details of the users. Except with their password being hashed. In this way the password of every object(user) is secured.

1.3 OUTCOME

During the execution, console displays the choices of creating account. It gives option to sign up finding account, admin log-in. Sign up requires users to enter email-id and password and it gives the unique id number to the entered email. Option to find account asks for given id number and returns the id and password without the password being hashed. The admin login asks for admin password and returns employee email with their password being hashed.

CHAPTER 2

OBJECT ORIENTED FEATURES

2.1 OBJECTS

Objects are basic run time entities in an object oriented system. In my project I am creating objects for various classes such as student, amount, fine etc. Using these objects I am invoking that particular function which is a member function of that class. It displays the operations performed in it.

2.2 CLASS

The entire set of data and the code of an object can be made a user-defined data type with the help of class. Class is thus a collection of objects of similar type and usually classes are user defined data types and act as built in data types.

In this project I am using class to define various data types. I have implemented multiple classes it provide template for data and it contains member functions, which perform specified operation.

2.3 DATA ABSTRACTION

Abstraction is one of the features of object-oriented programming, where only the relevant details are shown to the user and hide the irrelevant details. That is represent the needed information in program without presenting the details.

It is a technique that relies on separation of interface and implementation.

In C++ we use classes to define abstract data types of our own choice.

The two advantages of data abstraction are:

1. Makes the application secure by making data private and avoiding the user level error that may corrupt the data.

2. This avoids code duplication and increase the code reusability.

In the program the data abstraction is done by hiding the function of encryption and decryption. These functions are called through the header file. encryption.h is the header file. This file contains the hashing of password and returning the hashed password and decrypting the password.

2.4 POLYMORPHISM

The word polymorphism means having many forms. In other simple words, we can say that polymorphism is the ability of a message to be displayed in more than one form. Polymorphism occurs when there is a hierarchy of classes and these are related by the inheritance.

Polymorphism is one of the feature of oops that allows the object to behave differently in different conditions. In C++ we have two types of polymorphism:

1) Compile time Polymorphism

Example of compile time polymorphism is the function-overloading and operator overloading.

2) Runtime Polymorphism

Example of runtime polymorphism is function overriding. Function Overriding is when child class declares a method, which is already present in the parent class. Here child class overrides the parent class.

2.5 ENCAPSULATION

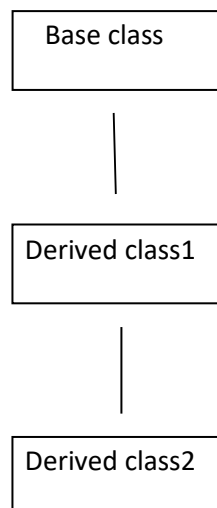
Encapsulation is a process of combining the data members and functions in a single unit called class. This is done to prevent direct access to the data, the access to the data is done through the functions of the class. While working with different procedural languages, it is not always clear which functions work on which variable but this is solved by oops that is object-oriented programming provides a way to place the data and relevant functions together in the same object.

To do this: Make all the data members private and create public setter and getter functions for each data members in such a way that the set function set the value of data member and get function get the value of data members.

In the program encapsulation is done by declaring the data members that is name, email-id and password of the object in the class. These data members are used in the class function that is taking the user or the object details and display the details.

2.6 INHERITANCE

Inheritance is one of the feature of oops, it allows the child class to aquire the properties and functionality of parent class. In other words, one of the most useful aspects of object-oriented programming is reusability. Inheritance is the process of creating a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class. This is very important concept since the feature helps to reduce the code size.



To make it more understanding let's take an example: let's assume that human is a class that has properties such as height, weight, color etc and functionality such as eating, sleeping, dreaming, working etc.

PASSWORD HASHING

In the program inheritance is used by creating two classes. One is the parent class named user. And the derived class named admin in which the properties and functionality of the parent class that is user class are inherited.

The admin class contains the display function. This function displays the details that is name, mail-id and the hashed password of the user.

CHAPTER 3

REQUIREMENTS AND PROJECT DESIGN

3.1 REQUIREMENTS

HARDWARE REQUIREMENTS:

- PIV 2.8 GHz Processor and Above
- RAM 512MB and Above
- HDD 20 GB Hard Disk Space and Above

SOFTWARE REQUIREMENTS:

- Ubuntu-Terminal
- Visual Studio .Net 2005 Enterprise Edition
- Internet Information Server 5.0 (IIS)

3.2 ALGORITHM

Step 1: Start.

Step 2: Print choices.

1. Sign up.
2. Find account.
3. Admin login.
4. Exit.

Step 3: Switch(ch)

PASSWORD HASHING

1. Case 1:

- a. Input name, mail-id, password.
- b. Hashing of password.

While(i !=len)

Convert to ascii value: pt-96.

For(j=0 ; j<key)

```
{  
    K=K*ascii;  
    K=K%n;  
}
```

Convert back: k+96.

- c. Gives unique number to id signed in.
- d. Store the data.

2. Case 2:

- a. Enter the unique number to find the account.
- b. Decryption

while(m[i] != '\0')

for(j = 0; j < key; j++)

```
{  
    k = k * ct;  
    k = k % n;  
}
```

- c. Show data with original password.

3. Case 3:

PASSWORD HASHING

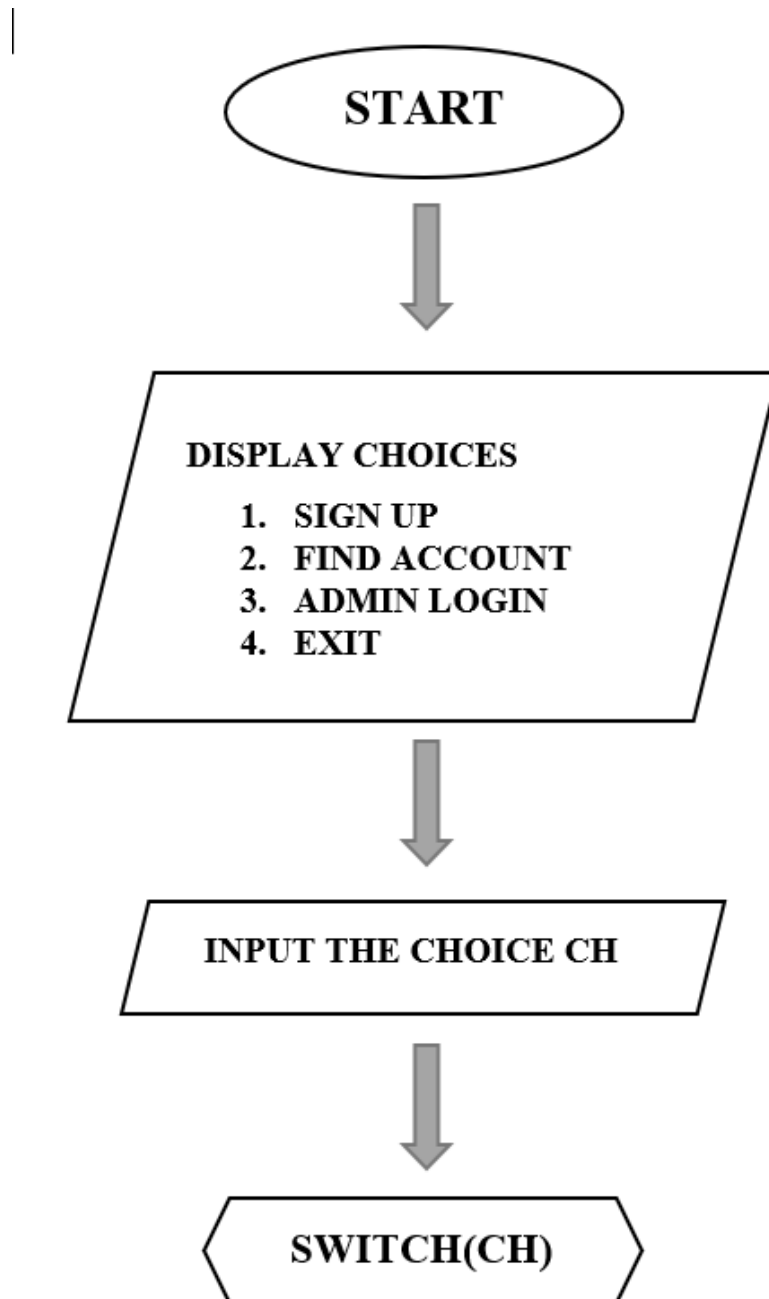
- a. Enter admin password.
- b. Compare with original password.
- c. If true
 - Print user(object) mail-id, password being hashed.

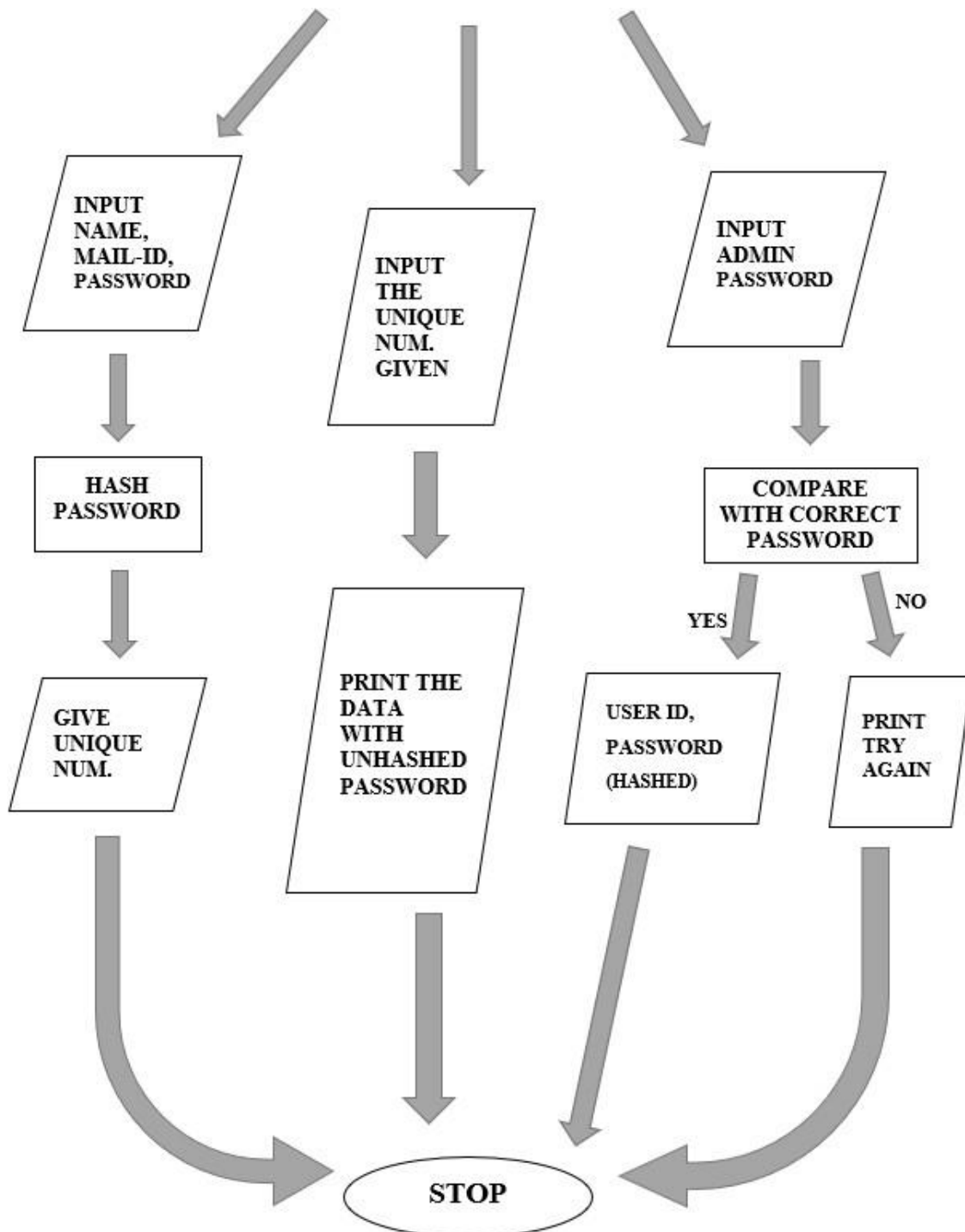
Else: Print try again.

4. Case 4: Exit.

Step 4: Stop.

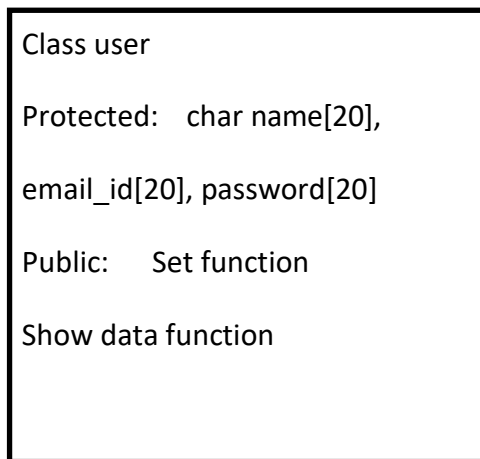
3.3 FLOWCHART



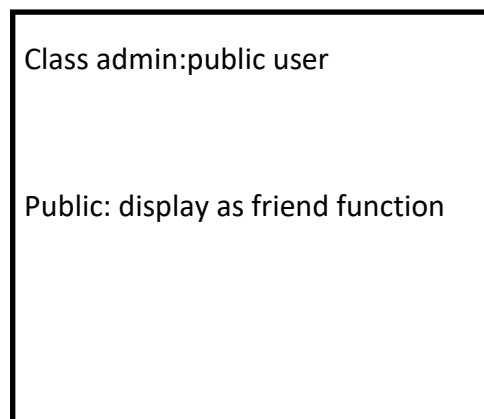


3.4 CLASS DIAGRAM

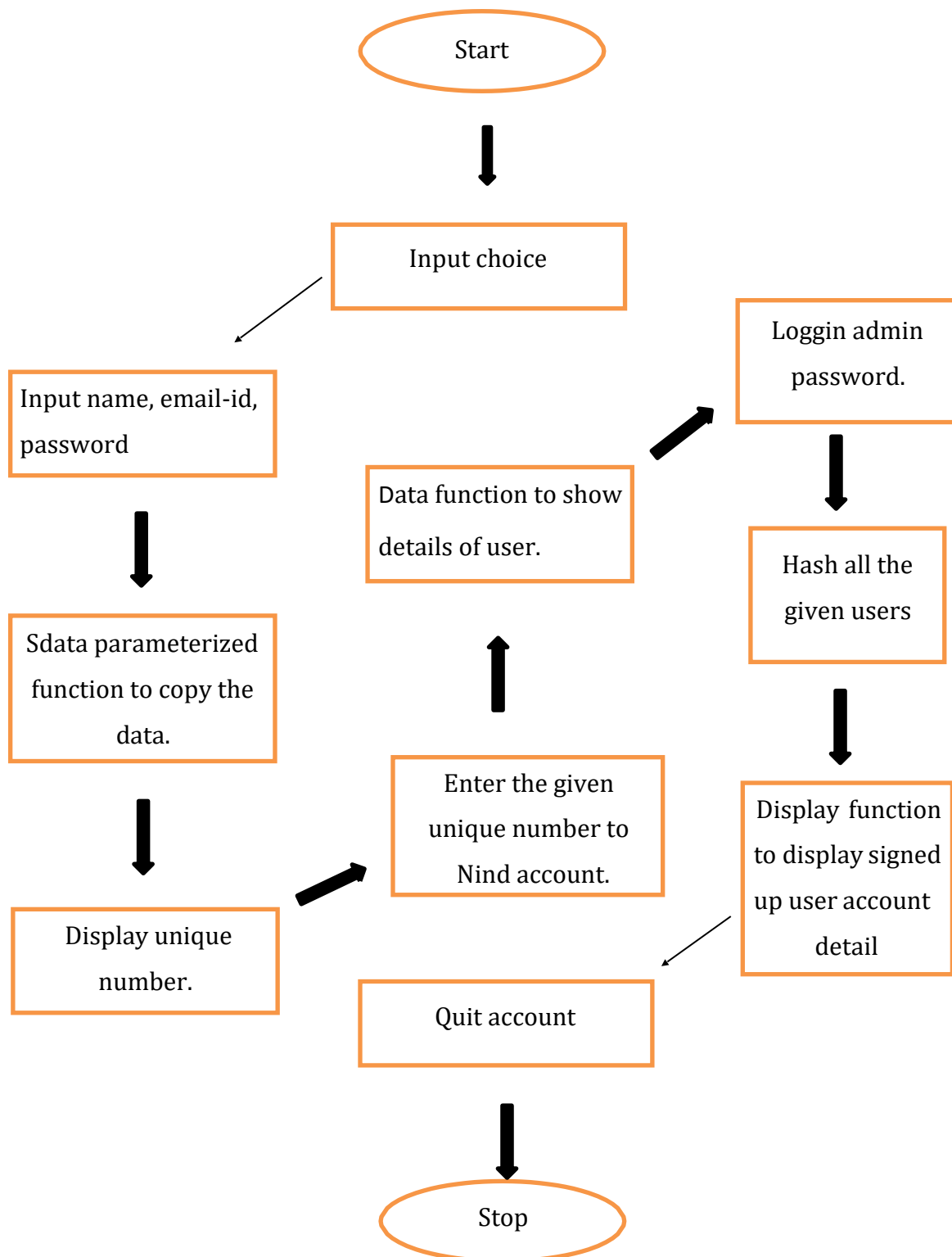
BASE CLASS



DERIVED CLASS



3.5 DATA FLOW DIAGRAM



CHAPTER 4

IMPLEMENTATION

In this project I'm implementing the method of securing the password of the login user by hashing the password. In this there are three functions which are :

1. Sign up: the user logs in by giving user's name, email-id and password. Which gives a unique number when the account is created.
2. Finding account: the user can find their account by giving their unique number.
3. Admin: admin logs in to find the user who have logged in under him.

```
#include<iostream>

#include<string.h>

#include<iomanip>

#include"hashing.h"

using namespace std;

class user

{

protected:

    char email_id[30],password[20],name[20];

public:

    void setdata(char id[30],char pass[20],char n[20])

    {

        strcpy(email_id,id);

        strcpy(password,pass);
```

PASSWORD HASHING

```
        strcpy(name,n);
    }

void showdata(char *email,char *password,char *name)
{
    cout<<"NAME: "<<name<<endl;
    cout<<"EMAIL ID: "<<email<<endl;
    original(password);
    cout<<"PASSWORD:"<<password<<endl;
    hashing(password);
}

};

class admin: public user
{
public:
    friend void display(admin obj);
};

void display(admin obj)
{
    cout<<le | <<setw(28)<<obj.email_id<<le | <<setw(28)<<obj.password<<endl;
}
}
```

PASSWORD HASHING

```
int main()
{
    char na[20],eid[30],ps[20];

    int count=0,key,i,j=0,ch;

    char password_admin[]={"admin123"},password_check[20];

    admin ob1[30];

    for(i=0;i<=30;i++)
    {
        cout<<"\n\n ----- \n\n";
        cout<<"1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit\n\n";
        cout<<"-----SELECT OPTION ----- \n\n";

        cin>>ch;

        switch(ch)
        {
            case 1:count++;

                cout<<"Enter your name: ";

                cin>>na;

                cout<<"\nEnter valid email id: ";

                cin>>eid;

                cout<<"\nEnter password of MAX 20 characters: ";

                cin>>ps;

                hashing(ps);

                cout<<"\nYour unique number to find your account is:

"<<count<<endl;

                ob1[j].setdata(eid,ps,na);
```

```
        j++;

        break;

    case 2:cout<<"enter the unique number to find your: ";

        cin>>key;

        ob1[key-1].showdata(eid,ps,na);

        break;

    case 3:cout<<"Enter admin password: ";

        cin>>password_check;

        if((strcmp(password_admin,password_check)==0))

        {

            cout<<"\n\nCORRECT PASSWORD\n\n";

        }

        else

        {

            cout<<"\n\nTRY AGAIN\n\n";

            break;

        }

        cout<<" -----\n";

        cout<<le|<<setw(28)<<"EMAIL ID"<<le|<<setw(28)<<"PASSWORD";

        cout<<"\n ----- \n\n";

        for(i=0;i<count;i++)

            display(ob1[i]);

        break;

    case 4:cout<<"*****QUITTING*****\n\n";

        exit(0);
```

```
        default:cout<<"enter valid choice\n";

    }

}

return 0;

}
```

```
#include<iostream>

#include<stdlib.h>

#include<math.h>

#include<string.h>

using namespace std;

int x=11, y=23, n=253, t=220, i, flag;

long int temp[50], j;

void hashing(char m[])

{

    long int pt, ct, key = 13, k, len;

    i = 0;

    len = strlen(m);

    while(i != len)

    {

        pt = m[i];
```

```
    pt = pt - 96;

    k = 1;

    for(j = 0; j < key; j++)
    {
        k = k * pt;

        k = k % n;
    }

    temp[i] = k;

    ct= k + 96;

    m[i] = ct;

    i++;
}

m[i] = '\0';
}
```

```
void original(char m[])
{
    long int pt, ct, key = 17, k;

    i = 0;

    while(m[i] != '\0')
    {
        ct = temp[i];

        k = 1;

        for(j = 0; j < key; j++)
        {
```


PASSWORD HASHING

```
    k = k * ct;

    k = k % n;

}

pt = k + 96;

m[i] = pt;

i++;

}

m[i] = '\0';

}
```

CHAPTER 5

OUTPUT SNAPSHOTS

OPTION DISPLAY

```
-----  
1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit  
-----SELECT OPTION-----
```

THE FIRST OPTION ASKS FOR INPUT

```
-----  
1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit  
-----SELECT OPTION-----  
1  
Enter your name: jacob  
Enter valid email id: jac@gmail.com  
Enter password of MAX 20 characters: jacob1234@
```

A UNIQUE ID NUMBER IS PROVIDED FOR EACH SIGN UP

```
1  
Enter your name: jacob  
Enter valid email id: jac@gmail.com  
Enter password of MAX 20 characters: jacob1234@  
_###_Your unique number to find your account is: 1
```

PASSWORD HASHING

SIGNED UP ACCOUNTS

```
1
  Enter your name: jacob
  Enter valid email id: jac@gmail.com
  Enter password of MAX 20 characters: jacob1234@
_###_ Your unique number to find your account is: 1

-----

-----

1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit
-----SELECT OPTION-----

1
  Enter your name: lanis
  Enter valid email id: lan@gmail.com
  Enter password of MAX 20 characters: lany!@#132
_###_ Your unique number to find your account is: 2

-----

-----

1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit
-----SELECT OPTION-----

1
  Enter your name: kiter
  Enter valid email id: kiter
  Enter password of MAX 20 characters: ki@123asd
_###_ Your unique number to find your account is: 3
```

SELECTING SECOND OPTION

```
-----

1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit
-----SELECT OPTION-----

2
enter the unique number to find your: █
```

PASSWORD HASHING

GIVE THE PROVIDED UNIQUE NUMBER TO FIND THE ACCOUNT. DISPLAY DETAILS OF THE ACCOUNT WITH ORIGINAL PASSWORD.

```
1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit
-----SELECT OPTION-----
2
enter the unique number to find your: 2
DETAILS OF YOUR ACCOUNT
NAME: lanis
EMAIL ID: lan@gmail.com
PASSWORD: lany!@#132
```

THIRD OPTION. ENTER ADMIN PASSWORD.

```
1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit
-----SELECT OPTION-----
3
ENTER ADMIN PASSWORD: █
```

IF THE PASSWORD IS CORRECT ALL THE SIGNED UP ACCOUNTS ARE DISPLAYED WITH THEIR PASSWORD BEING HASHED.

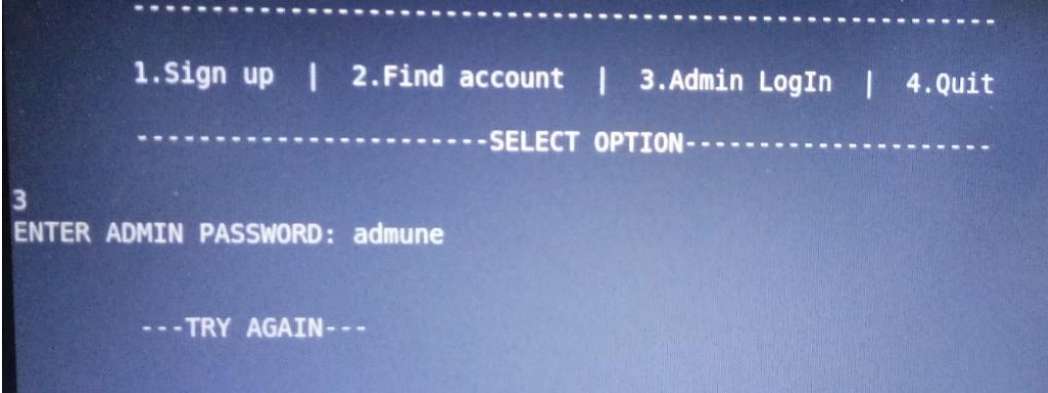
```
3
ENTER ADMIN PASSWORD: admin123

^^^CORRECT PASSWORD^^^

-----
EMAIL ID                                PASSWORD
-----
jac@gmail.com                          0a
lan@gmail.com                          0a0{00-030
kiter                                  規03aM0
```

PASSWORD HASHING

IF THE ADMIN LOGGIN PASSWORD DOES NOT MATCH THE ORIGINAL PASSWORD THEN THE MESSAGE TRY AGAIN IS DISPLAYED.



```
-----  
1.Sign up | 2.Find account | 3.Admin LogIn | 4.Quit  
-----SELECT OPTION-----  
3  
ENTER ADMIN PASSWORD: admune  
  
---TRY AGAIN---
```

CHAPTER 6

CONCLUSION

This project is done using the concept of oops. The oops concept makes it easier to create a program. This project is created for hashing passwords which is very useful for securing the passwords.

This program helps to guard against the possibility that someone who gains unauthorized access to the database can retrieve the passwords of every user in the system. While the user is creating an account and enters the password, the password is stored in hashed form. When someone other than user tries to access they'll view the password which will be hashed except the user have the authority to view the original password.

This concept is mainly for creating mail-id or securing facebook or any other social media sources used in a company or any working place.

CHAPTER 7

REFERENCES

www.Geekforgeek.com

www.Stackoverflow.com

www.tutorialsonpoint.com