



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS**

**A Project Report
On
Password Manager**

Submitted By:

Aashish Shroff (THA076BCT001)

Arun Subedi (THA076BCT010)

Grishma Raj Khanal (THA076BCT016)

Nishant Uprety (THA076BCT023)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

Under the Supervision of

Er. Saroj Shakya

October, 2021

COPYRIGHT

The author has agreed that the Library, Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering in any use of the material of this project report. Copying or publication or the other use of this report for financial gain without approval of to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering and author's written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head

Department of Electronics and Computer Engineering

Thapathali Campus, Institute of Engineering

Lalitpur, Kathmandu

Nepal

ACKNOWLEDGMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and We are extremely privileged to have got this all along the completion of our project. We couldn't have done without such supervision and assistance.

We would like to thank our **Department of Electronics and Computer Engineering**, for providing us with the opportunity to do the project work, the project was helpful for us to explore and implement the knowledge of programming and to collaborate as a team and serve as a way to showcase our creativity. Our team did a lot of learning and work towards the completion of the project.

We respect and thank Er. Saroj Shakya, for providing us an opportunity to do the project work and giving us all support and guidance, which made complete the project possible. We are extremely thankful to him for providing such a nice support and guidance, although he had busy schedule.

We would like to thank various writers, developers and youtubers whose works we have referenced for the project. The articles and videos were a great source of knowledge and help for all of us. The many confusions and many ways to tackle our arising problems were made ease with the help of those articles.

Finally, we thank all the friends who have been involved with us in the project in one way or another. We are thankful and fortunate enough to get constant encouragement, support and guidance in successful completion of our project.

Aashish Shroff (THA076BCT001)

Arun Subedi (THA076BCT010)

Grishma Raj Khanal (THA076BCT016)

Nishant Uprety (THA076BCT023)

ABSTRACT

Passwords are fundamental for information security. They are used as a first line defense in securing almost all our electronic information, networks, servers, devices, accounts, databases, files and many more. Most of us now have a multitude of passwords we need to somehow keep track and remember. This project will also provide an overview of how password management software works and also recommendation for secure password practices. In this project we intent to present a password manager program which keeps tracks and store all these different passwords created in a single place. Various functions and libraries of C++ has been used in creating the project. This project saves the users login credentials provides the user the option to edit, delete and add new credentials as they need.

Keywords: Modified Ceaser cipher, Passphrase, Security, SFML

Table of Contents

COPYRIGHT	i
ACKNOWLEDGMENT	ii
ABSTRACT	iii
List of Figures	vi
List of Tables	vii
List of Abbreviations	viii
1. INTRODUCTION	1
1.1 Background.....	1
1.2 Motivation	1
1.3 Objectives	2
2. LITERATURE REVIEW	3
2.1 Previous Projects	3
2.2 Existing system limitations.....	4
2.3 Techniques and Algorithm	4
2.4 Proposed System Benefits	5
3. METHODOLOGY	6
3.1 Program Execution Flow	6
3.1.1 Register	6
3.1.2 Login.....	7
3.1.3 Accessing saved info.....	8
3.1.4 Update User data.....	8

3.1.5	Signing out	9
3.2	Identifying subsystems	9
3.2.1	Application/Control layer	9
3.2.2	Data layer	10
3.2.3	Presentation layer	10
3.3	Encryption and Decryption techniques.....	10
4.	SYSTEM DESCRIPTION.....	11
4.1	Block diagram and system architecture	12
4.2	Tools and Environment	16
5.	RESULT AND ANALYSIS.....	19
5.1	Program Execution Flow	19
5.1.1	Loading Screen	19
5.1.2	New User Registration	19
5.1.3	New User Details	20
5.1.4	Existing Users	21
5.2	Analysis	23
6.	CONCLUSION AND FUTURE ENHANCEMENT.....	24
6.1	Conclusion and Limitations	24
6.2	Future Enhancement	25
	References	26

List of Figures

Figure 2-1: Ceaser Cipher technique [2].....	4
Figure 3-1: Block Diagram for first time registration.....	7
Figure 3-2: Block diagram for Login.....	7
Figure 3-3: Conversion of encrypted password to decrypted and vice versa	8
Figure 3-4: Block diagram for Updating User data	9
Figure 3-5: Block Diagram of control flow between different layers of system	10
Figure 3-6: Algorithm used for Encryption and Decryption [4].....	11
Figure 4-1: Basic layout of application.....	13
Figure 4-2: Login Class model	14
Figure 4-3: Registration Class model	15
Figure 4-4: UserInfo Class model.....	16
Figure 5-1: Loading screen	19
Figure 5-2: New User Registration	20
Figure 5-3: New User Details	21
Figure 5-4: Existing Users	22

List of Tables

Table 4-1: Classes used in the program	12
--	----

List of Abbreviations

IOE	Institute of Engineering
IDE	Integrated Development Environment
API	Application Programming Interface
SFML	Simple and Fast Multimedia Library
PM	Password Manager
UI	User Interface
GCC	GNU Compiler Collection
I/O	Input/output
GUI	Graphical User Interface

1. INTRODUCTION

Password Manger is a program related to present issue with large amount of login credentials and personal security details you need to remember. It stores and saves those credentials so users don't have to go through the trouble of remembering all of them.

1.1 Background

In current age of internet, there are hundreds, if not thousands of sites, applications, devices we need to visit and use in day-to-day basis, most of them are personalized with a security feature with a username or an email accompanied with a password because of this many people across the globe use same or easy to remember password in all of their site, which is an unhealthy digital behavior.

A password manager is essentially an encrypted digital vault that stores secure password login information people use to access apps and accounts on your mobile device, websites and other services. In addition to keeping your identity, credentials and sensitive data safe. It will free user the stress of remembering passwords or even worse writing in physical paper.

There are many password breaches and identity theft that happens more often than people think. With all the recent news of security breaches and identity theft, having a unique password for each location can go a long way to ensuring that if one site gets hacked, your stolen password can't be used on other sites. You're basically using multiple passwords to create your own security features. This helps you stay safe online and be sure your account won't be compromised easily.

1.2 Motivation

Communicating and working remotely has become the regular routine for us, so it is major necessity to have a strong password for online accounts. Using multiple accounts makes us difficult to memorize all login info. It sometimes can become time consuming to login through different websites and application in-case user mistypes or forgets his password. After being troubled for a long time our team members had always a wish to find some easy and quick method to access the login data for every account. Hereby,

realizing the priorities with user's data this project is brought in creation so as to facilitate anyone with simpler method to login different websites without wasting any time.

If you use weak passwords or the same one everywhere, you are only making it easier for someone to compromise all your accounts. Many people even still in 2021 are found to follow terrible password practices like using 'password' or 12345678' as their password, either by deliberately accepting greater risk for the sake of convenience. To solve this problem, we aim to add a feature to a basic password manager software which generates a random but strong passwords and passphrases for the users.

1.3 Objectives

The main objectives of our project were:

- To design and develop a clean and easy to use password managing software.
- To promote the use of Passphrases instead of Passwords.

2. LITERATURE REVIEW

2.1 Previous Projects

Passwords have a much longer history than just the digital era. Sentries for centuries would use watchwords to identify friend or foe. The Roman military reportedly used passwords as a way to distinguish friend from foe

Fernando Corbató, widely regarded as the godfather of the modern computer password, introduced the idea to computer science while working at the Massachusetts Institute of Technology (MIT) in 1960 to help keep individual files private, the concept of a password was developed so that users could only access their own specific files for their allotted four hours a week. However, as the world wide web exploded in the 90s, more and more people began using the internet on a regular basis, creating reams of sensitive data and information in the process. Even before that early computer scientists were working on a way to make passwords more secure. And, to do that, computer science took a leaf from cryptology. [1]

Hashing, then Salting were introduced to ensure the security of the password by adding another level to it with the purpose to stop password leakage mainly. But the with the increase in cybercrimes users are recommended to adopt a “passphrase” strategy for increased security or adopt two step verification, where a password is only one step in gaining access to sensitive data.

In the last decade, startups and researchers have proposed appropriately futuristic methods to strengthen passwords, or replace them completely. These range from password managers like Dashlane, LastPass, KeePass, which stores centralize and encrypt passwords and other personal data.

2.2 Existing system limitations

The User Interface design is not as clean and easy to use. With the additions of complex features the system of the available program crashes frequent which sometimes results in password leakage.

A password manager stores all the passwords of users and fills the details automatically. If we rely completely on password managing software we tend to forget even particularly important passwords after a long time.

2.3 Techniques and Algorithm

One of the simplest examples of substitution cipher is the Ceaser cipher, which is said to have been used by Julius Caesar to communicate.

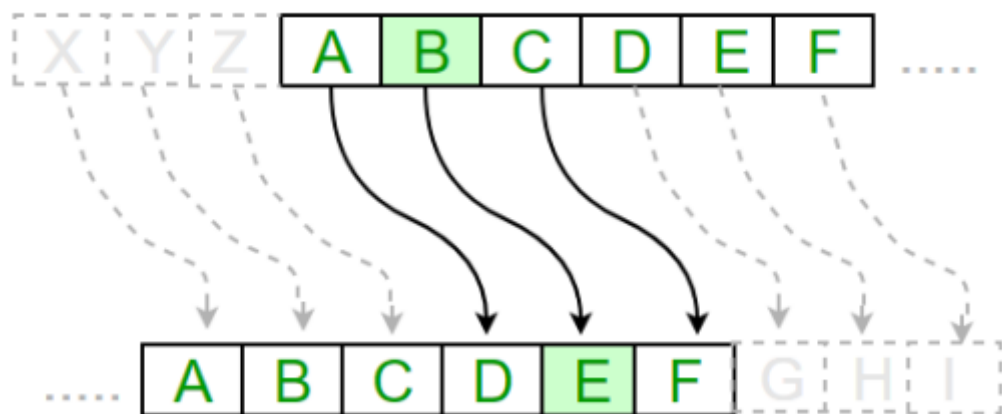


Figure 2-1: Ceaser Cipher technique [2]

In this program we are implementing Modified Ceasar cipher which also is an example of substitution cipher. Program consists of two methods encrypt and decrypt. The encrypt method has two parameter one the plain text and second is key. In Modified Caesar cipher each alphabet of plain text is may not necessarily replaced by key bits down the order instead the value of key is incremented and then it is replaced with new key value. The decryption method also has two parameters one encrypted message and key. It does opposite process of encryption. [3]

2.4 Proposed System Benefits

1. A clean and bugs free program is to be developed to provide smooth operation of task.
2. Provision to use passphrases instead of only password for master password

3. METHODOLOGY

A password manager assists in generating and retrieving complex passwords and storing such passwords in an Encrypted format or calculating them on demand. Password managers typically require a user to remember one "master" password to unlock and access any information stored in their database. The major entities in our project are encryption/decryption, File handling, password generator, and SMFL graphics.

3.1 Program Execution Flow

The program will start from main function in main.cpp where Login class and Registration class are executed as per the user input.

3.1.1 Register

When the program is executed, "SFML" window is rendered which ask the user to sign up and enter the username and master password in register section. If new username matches with previous users, then the user has to re-enter a new username.

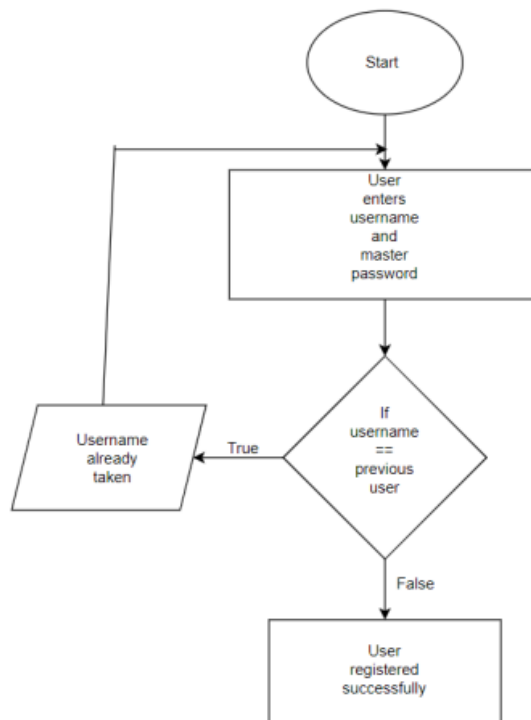


Figure 3-1: Block Diagram for first time registration

3.1.2 Login

The user is then taken to login. If the user enters a wrong password or username in login section, it will display invalid input and again asked for the login.

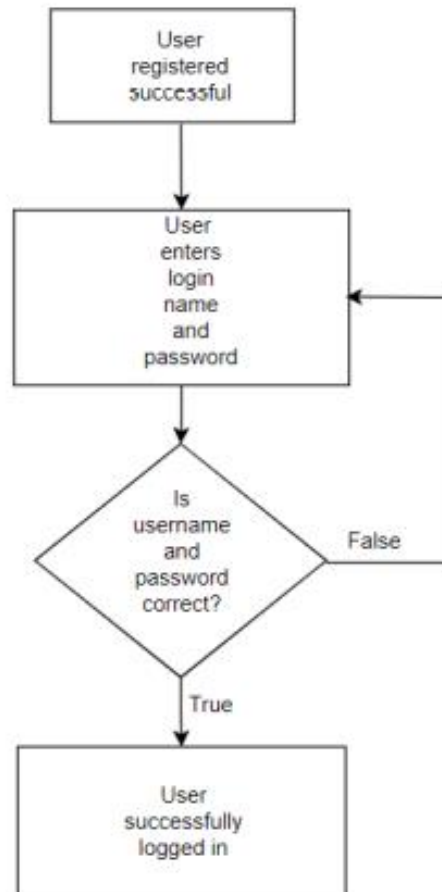


Figure 3-2: Block diagram for Login

3.1.3 Accessing saved info

After successful login, Program retrieves password from file in decrypted format using the algorithm used for decryption in secure class and displays the info on the screen.

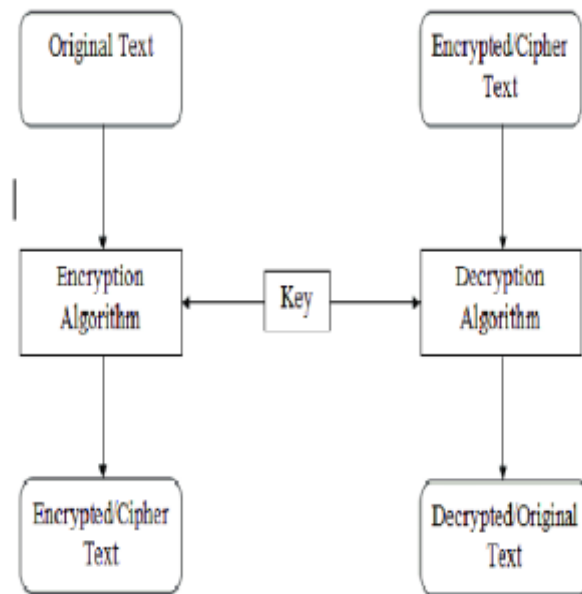


Figure 3-3:Conversion of encrypted password to decrypted and vice versa

3.1.4 Update User data

Users can access their saved information, and will be able to add or delete a particular data.

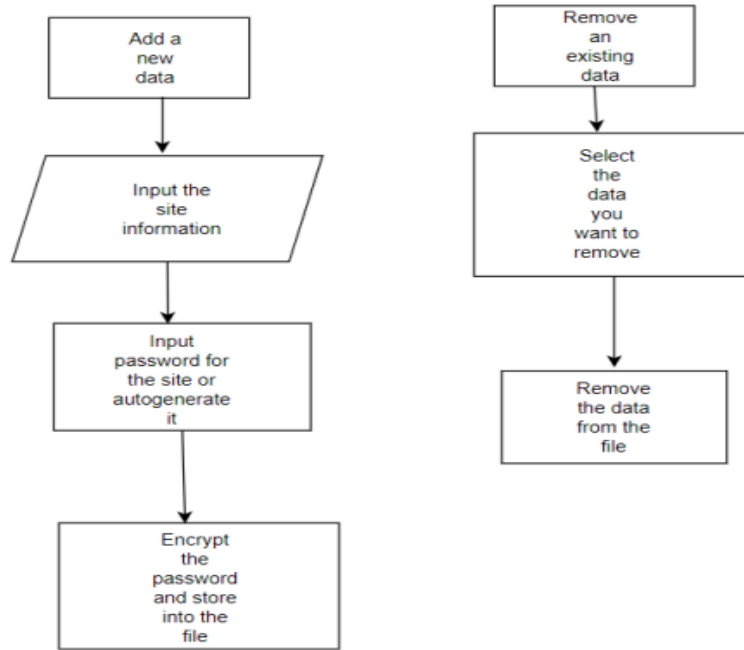


Figure 3-4: Block diagram for Updating User data

3.1.5 Signing out

Users can exit from their password manager account via sign out option which took the user to the login window.

3.2 Identifying subsystems

This project is based on three different layer system viz. application layer, data layer and presentation layer.

3.2.1 Application/Control layer

The application layer contains what will call and manage the whole operation. The controller will facilitate the tasks between the different classes & functions. It allows the user to handle the communication between the user and his/her personal data in each user interface.

3.2.2 Data layer

The data layer consists of the database or file which stores information of the user with username or email, password (in encrypted format), along with name of websites.

3.2.3 Presentation layer

The presentation layer holds the different screens that the user will be able to see after completing the register and login. This layer pulls information from the database while program run and displays it on the screen. The presentation layer consists of the main window/screen for the new register/sign-up option and login option for old user, and ask for the master password. After that the user's detailed information will get displayed with all the saved email and password in decrypted format where they can add or remove a particular detail.

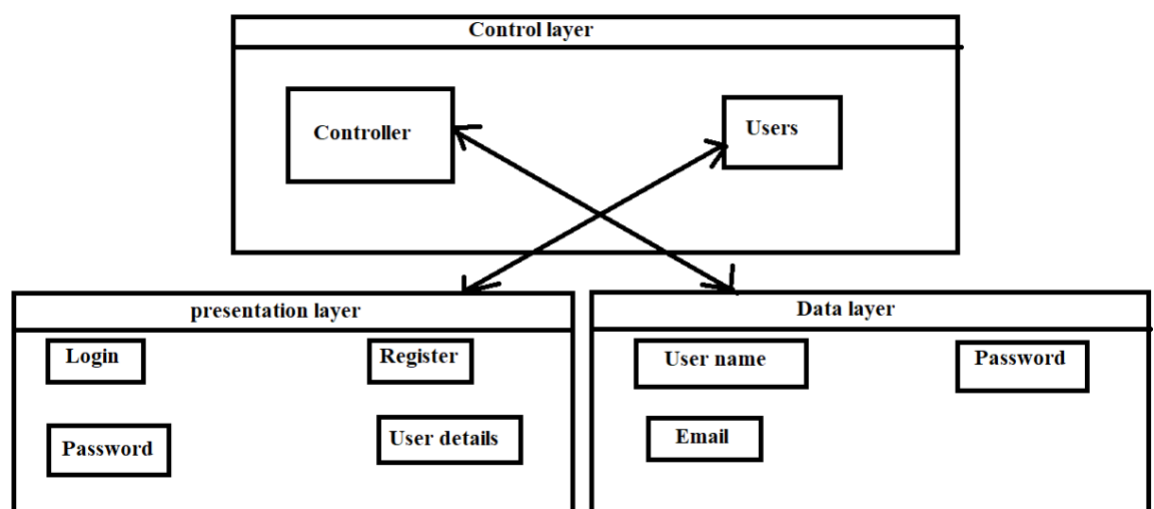


Figure 3-5: Block Diagram of control flow between different layers of system

3.3 Encryption and Decryption techniques

Password manager uses encryption technique to translate plain text data (plaintext) into something that appears to be random and meaningless (cipher text). Decryption is the process of converting cipher text back to plaintext.

In this project, we use Modified Caesar cipher which is an example of substitution cipher. Program consists of two methods encrypt and decrypt. The encrypt method has two parameter one the plain text and second is key. In Modified Caesar cipher each alphabet of plain text may not necessarily replaced by key bits down the order instead the value of key is incremented and then it is replaced with new key value. The decryption method also has two parameters one encrypted message and key. It does opposite process of encryption.

The goal of every encryption algorithm is to make it as difficult as possible to decrypt the generated cipher text without using the key. If a really good encryption algorithm is used, there is no technique significantly better than methodically trying every possible key. For such an algorithm, the longer the key, the more difficult it is to decrypt a piece of cipher text without possessing the key.

$$c_i = E(p_i) = p_i + 3$$

A full translation chart of the Caesar cipher is shown here.

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

Using this encryption, the message

TREATY IMPOSSIBLE

would be encoded as

T	R	E	A	T	Y	I	M	P	O	S	S	I	B	L	E
w	u	h	d	w	b	l	p	s	r	v	v	l	e	o	h

Figure 3-6: Algorithm used for Encryption and Decryption [4]

4. SYSTEM DESCRIPTION

This project mostly revolved around the safety and easy accessibility of user data. So, the major entities were file handling, encryption/decryption and SMFL graphics. The program was firstly divided into different classes with each class having specific jobs. The application will start with a Login window or main window where the user is asked to login. After successful login, the user can navigate through saved user-info. The

login-user data will be saved in user file and the saved passwords for each user will be stored in separate files after encryption. The different classes made for specific purposes in this application can be summarized as:

Table 4-1: Classes used in the program

Classes	Purpose
Main	Main file of the application
Login	Login Page of the application
Secure	Handles the encryption & decryption
Registration	Registration of users
UserInfo	User Information

4.1 Block diagram and system architecture

The program will start from main function in main.cpp where Login class and Registration class are executed as per user input. The Login page will show a screen where user can input their username and password or select Register option to register as a new user. If the user logs in, s/he will be able to access saved information along with option to add new password or generate a random secure password for use. The required class will be called as required by the program.

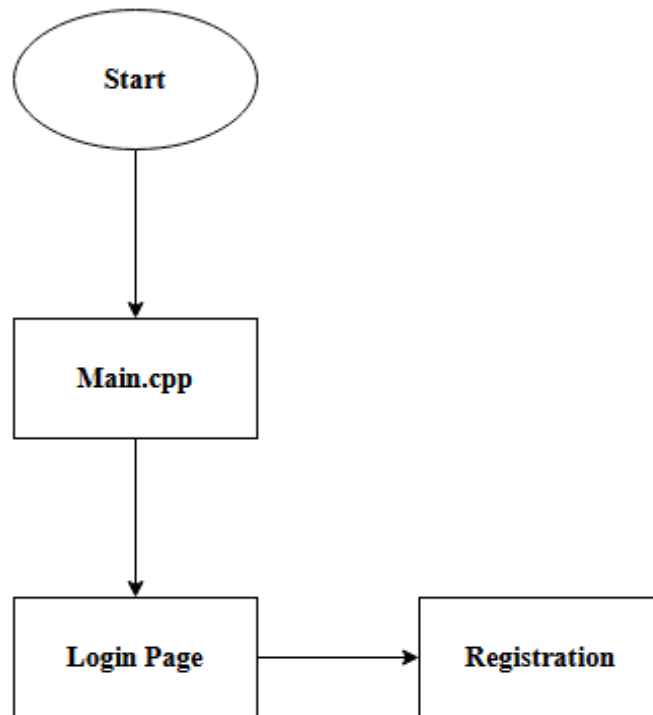


Figure 4-1: Basic layout of application

4.1.1 Login Class

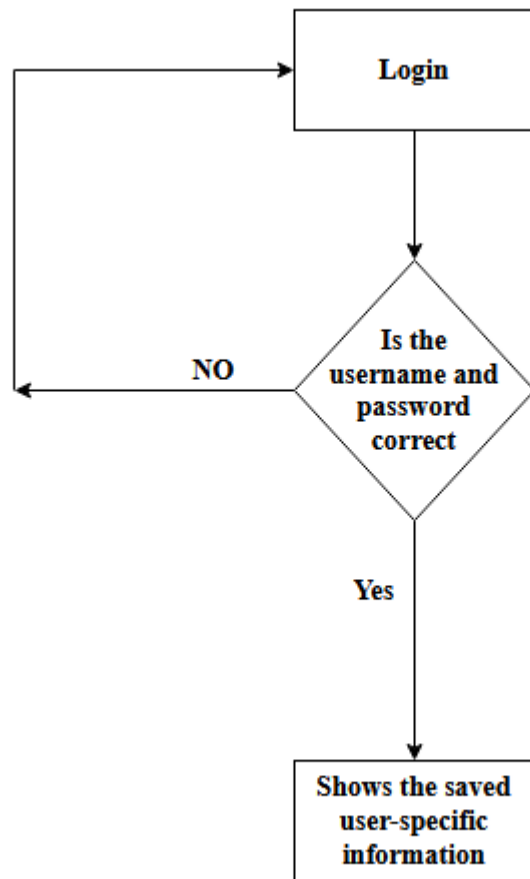


Figure 4-2: Login Class model

4.1.2 Registration Class

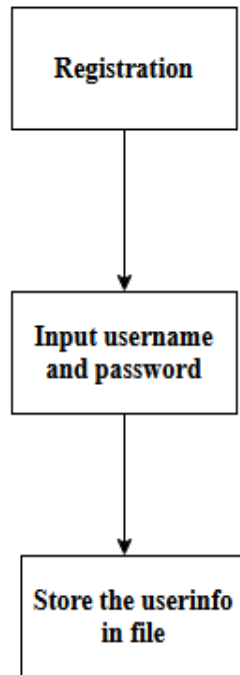


Figure 4-3: Registration Class model

4.1.3 UserInfo Class

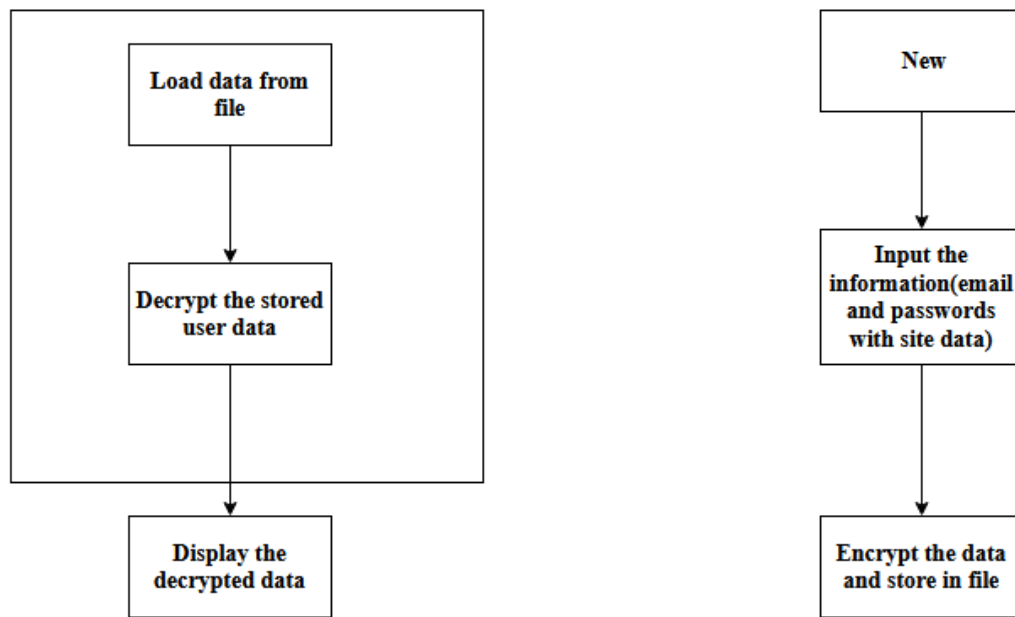


Figure 4-4: UserInfo Class model

4.2 Tools and Environment

The proper use of tools either for communication or for programming can have a massive impact on the outcome of any project. The tools that were used in the project can be classified mainly in two categories: Programming Language and Environment tools.

4.2.1 Programming language and libraries

4.2.1.1 C++ Programming Language

C++ is a general-purpose programming language created as an extension of C programming language. The modern C++ has object-oriented, generic and functional features. It was designed with performance, efficient and flexibility as its highlights. Because of all these features we used C++ as major language for the program.

4.2.1.2 Fstream

Fstream, data type represents the file stream, and has the capabilities of both ofstream and ifstream. Objects of this class maintain filebuf object as their internal stream buffer, which performs input/output operations on the file they are associated with. File streams are associated with files either on construction, or by calling member open. This library was used in this project for the file management of user data and save files.

4.2.1.3 SFML

Simple and Fast Multimedia Library (SFML) is a cross-platform software development library designed to provide Application Programming Interface (API) to various multimedia components in computer. It is written in C++ and is composed of five modules. Some of which were used in this project.

4.2.2 Environmental Tools

4.2.2.1 IDE

There are many options to choose for integrated development environment (IDE). We chose Code::Blocks as it is very easy to setup and accessible to each member of our team Code::Blocks is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. It has not been set in stone and if any problem had raised, we had changed into using Dev C++ for a while but we used Codeblock for the most of it..

4.2.2.2 Discord

Discord is a VoIP, instant messaging and digital distribution platform designed for creating communities. Users communicate with voice calls, video calls, text messaging, media and files in private chats or as part of communities. We used this for communication among the members by text, audio or video means.

4.2.2.3 Microsoft Teams

Microsoft Teams is a proprietary business communication platform developed by Microsoft, as part of the Microsoft 365 family of products. Teams primarily competes with the similar service Slack, offering workspace chat and videoconferencing, file storage, and application integration. We used Microsoft Teams for file sharing(codes) and for communication.

5. RESULT AND ANALYSIS

5.1 Program Execution Flow

5.1.1 Loading Screen

This is the first page of the program consisting of our logo design.



Figure 0-1: Loading screen

5.1.2 New User Registration

Whenever a new user tries to add account details in the program, he/she is guided to this page to create a username and a strong Master Password/Passphrase. After sign up he/she is asked to add details which is then saved after encryption process.

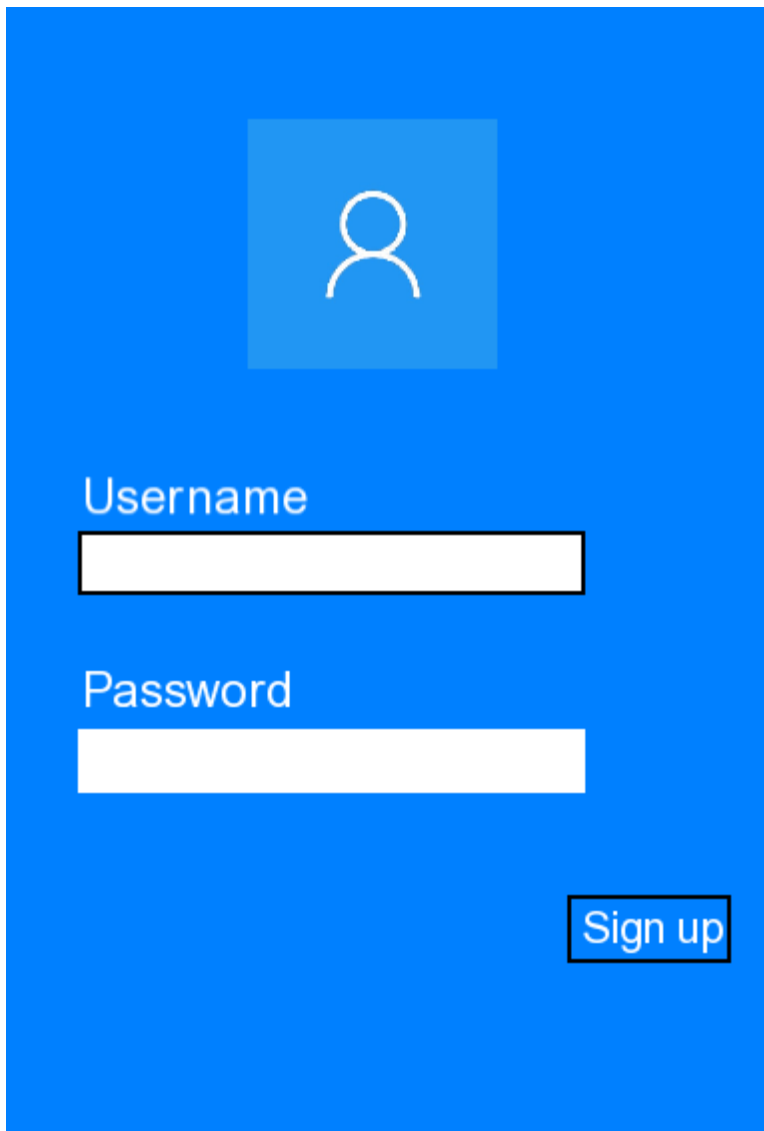
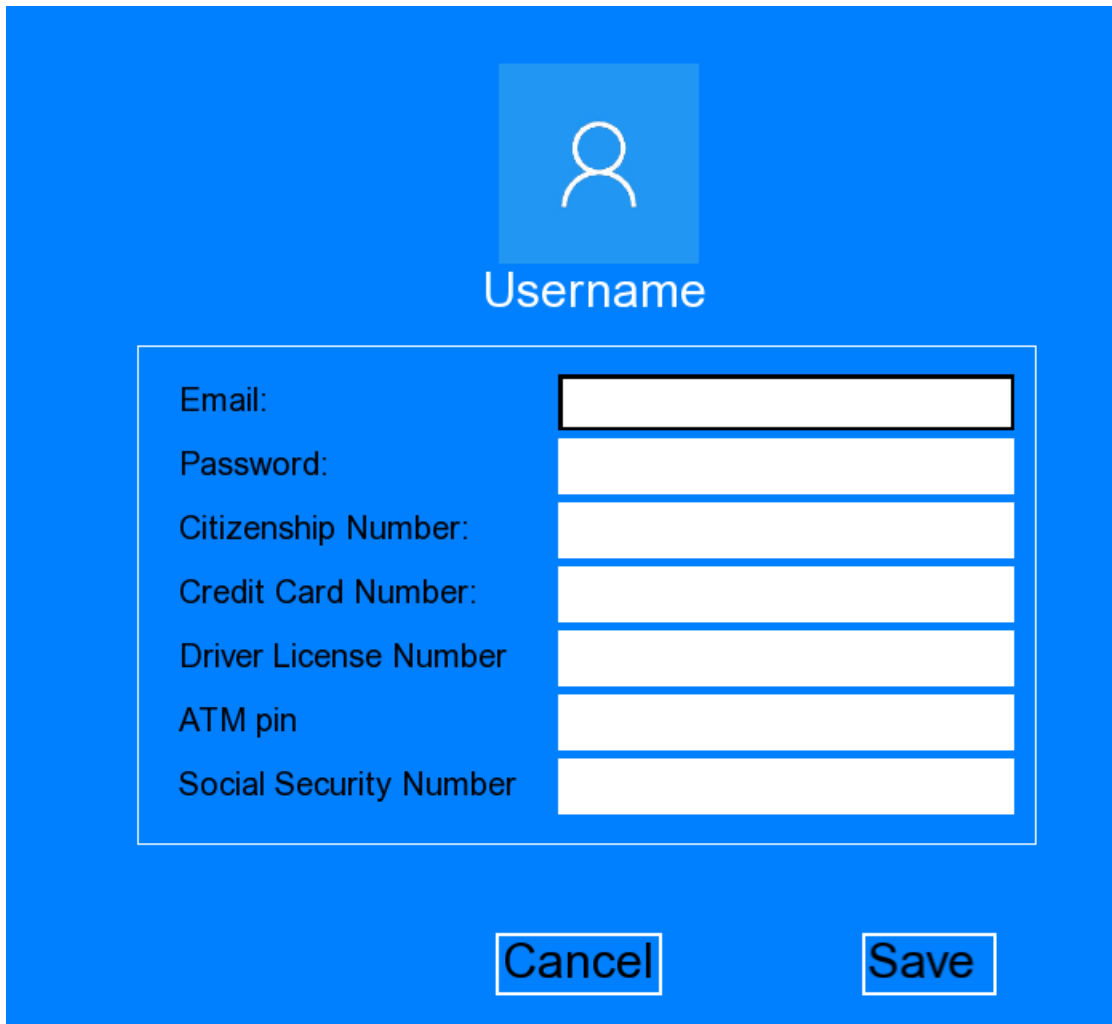
A registration form on a blue background. At the top center is a light blue square containing a white user icon. Below this, the text 'Username' is followed by a white input field. Further down, the text 'Password' is followed by another white input field. In the bottom right corner, there is a blue button with a black border and the text 'Sign up' in white.

Figure 0-2: New User Registration

5.1.3 New User Details

After sign up user can enter his/her details which is then saved in separate file after encryption process After hitting save the user is again redirected to home screen.

The image shows a user registration form titled "New User Details" on a blue background. At the top center is a light blue square containing a white person icon. Below this icon is the word "Username" in white text. The main form area is a white rectangle with a thin blue border. Inside this rectangle, on the left side, are labels for "Email:", "Password:", "Citizenship Number:", "Credit Card Number:", "Driver License Number", "ATM pin", and "Social Security Number". To the right of each label is a white input field with a black border. At the bottom of the form are two buttons: "Cancel" and "Save", both with black borders and white text.

Username

Email:

Password:

Citizenship Number:

Credit Card Number:

Driver License Number:

ATM pin:

Social Security Number:

Cancel Save

Figure 0-3: New User Details

5.1.4 Existing Users

This screen shows all the user account created in the program with individual different password/passphrases. After choosing your specific account one is directed to the screen where they have to enter Master Password to get access to their details.

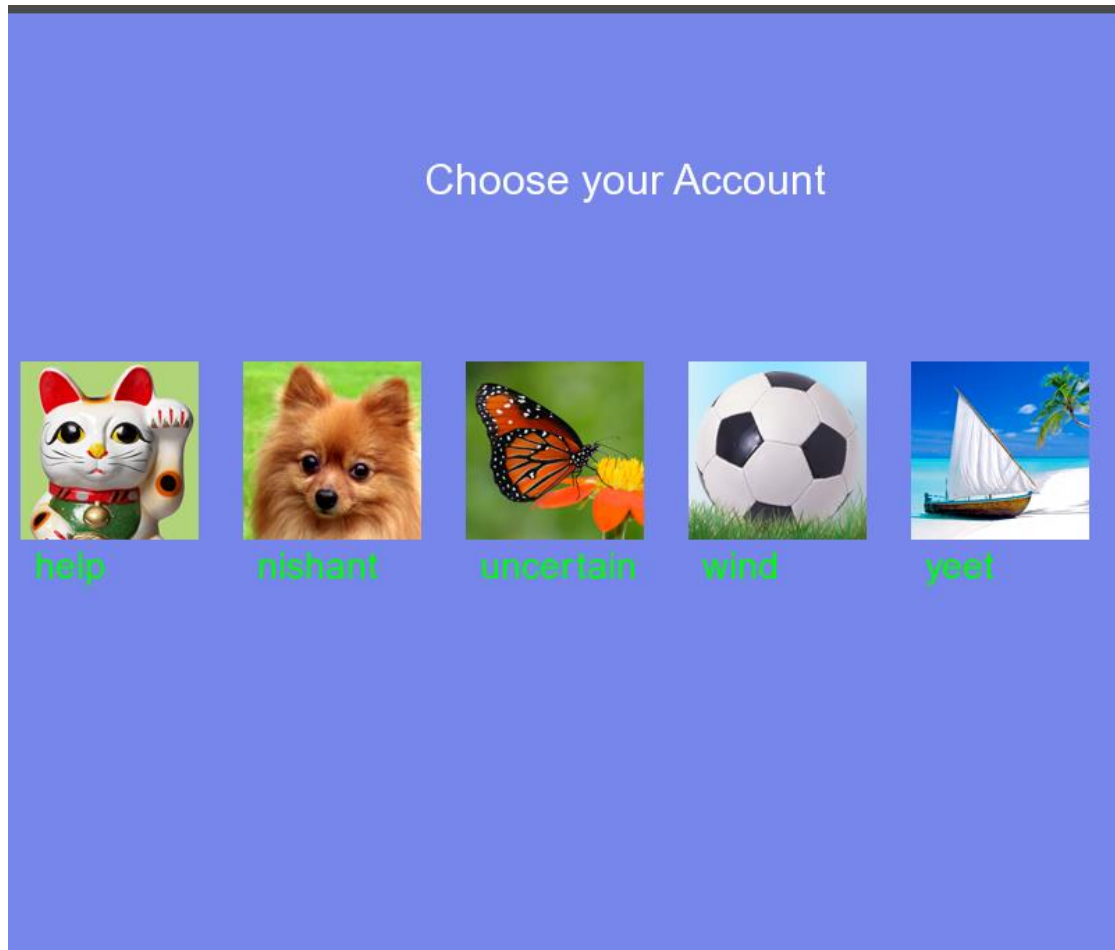


Figure 0-4: Existing Users

5.2 Analysis

Password Manager in today's date has been more important than the internet itself. If you are in internet or let's, say if your computer is plugged in you are vulnerable to unimaginable cyber threats. Our program mainly encourages the use of Password Managing software and healthy password practices. The program provides some sense of security but with the use of available resources and limited time period we are only able to use the pre medieval method for encryption and decryption making us still vulnerable to those who are willing to spend hours guess the correct encryption key and what more but with the use of Passphrase instead of just single word system of Password the program is somewhat more effective in data hiding due to strong master key.

6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion and Limitations

In these days, it is known that authentication based on just passwords is insecure. Already different websites like LinkedIn and eHarmony were attacked and private information was stolen. In the year 2014 a list of Gmail accounts was made public. One of the problems is that users tend to select easy to remember password and use it for all accounts everywhere in the internet. This vulnerability is easily exposed in the web and leads to successful attacks

To resolve these problems there are password managers. A password managers is a software or a device which stores all the passwords. By using this password manager, the user needs to remember only one password, which is used to access all the other ones. There are multiple password managers with different features like: encryption, two-factor authentication, generate passwords, etc. There are some web based password managers like: LastPass, FinalKey, Dashlane and some of them can store the passwords locally.

In the end we can say that we managed to create a prototype, that can be used in every day and has a high security by having multiple layers and is easy to use. Being easy to use let any kind of users to use it, from IT professional to simple computer users. Therefore this password manager can be used by everyone. But various attacks can be launched by hacker groups or governmental attacker, which have all the required resources and knowledge. Therefore the user needs to take in consideration that the encrypted passwords are not 100% safe. The chapter Future Work presented some features that can be added to improve the usability and security of the password manager.

6.2 Future Enhancement

This chapter presents some future enhancement that can be added to the program.

6.2.1 Phone application

The application can be ported on multiple OS. For example it can be written for iOS and for Windows phones. In this way there can be multiple users. This will not be so hard because this application is straight forward with just further knowledge mobile application development.

6.2.2 Advance Encryption Methods

Encryption is a critical tool used to protect data, establish trust, and maintain regulatory compliance. Advanced Encryption Standard is a symmetric encryption algorithm that encrypts fixed blocks of data (of 128 bits) at a time. The keys used to decipher the text can be 128-, 192-, or 256-bit long. The 256-bit key encrypts the data in 14 rounds, the 192-bit key in 12 rounds, and the 128-bit key in 10 rounds. Each round consists of several steps of substitution, transposition, mixing of plaintext, and more. AES encryption standards are the most commonly used encryption methods today, both for data at rest and data in transit.

References

- [1] A. Burgher, "A short history of passwords," *We live security*, 2017.
- [2] S. Jain, "Geeksforgeeks," [Online]. Available: <https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/>. [Accessed July 2021].
- [3] J. Andress, "Cryptography," *The Basics of Information Security*, 2014.
- [4] N. Maccha, "Network Security and Cryptography," 2020.