TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Project Proposal
On
Password Manager

Submitted By:

Aashish Kumar Sah (THA076BCT001)

Arun Subedi (THA076BCT010)

Grishma Raj Khanal (THA076BCT016)

Nishant Uprety (THA076BCT023)

Submitted To:

Department of Electronics and Computer Engineering
Thapathali Campus
Kathmandu, Nepal

July, 2021

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.

Keywords: Modified Ceaser cipher, Passphrase, Security, SFML

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

IOE	Institute of Engineering
SFML	Simple and Fast Multimedia Library
MIT	Massachusetts Institute of Technology
UI	User Interface
GCC	GNU Compiler Collection
API	Application Programming Interface
IDE	Integrated Development Environment
I/O	Input/output
GUI	Graphical User Interface
VoIP	Voice Over Internet Protocol
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator

1. INTRODUCTION

1.1 Background Introduction

The growth of online services has vastly increased the number of passwords a user has to remember while accessing various online accounts. Although it is essential to create unique and strong password for each account, remembering all those passwords is becoming burden for users. A password manager is a " computer program that allows user to store, generate and manage their passwords for local and online services ".

A password manager simply assists in generating and retrieving the login information of various accounts and automatically enter them into the forms. The automatic form filling feature fills the login information for a particular URL whenever it loads resulting fewer manual errors. As password manager can identify the right URL for a particular login ID and password pair automatically, they are capable of protecting credentials from phishing sites.

Nowadays most of the browser like chrome, safari has added the password manager that can store and generate random password for us. These types of password manager use yet another password commonly known as master password to protect our passwords. If we use Google's chrome browser to store our password and share them across devices, our password will be stored by Google and protected by password for our Google account. Whereas Apple's iCloud keychain relies primarily on our device password and unlocking feature to protect the data on regular basis.

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 incase 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. Herby 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.

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 Problem Definition

There are two main problems here. First is to generate a strong passwords/passphrase for different online accounts and secondly to store these passwords safely retrievable only by the owner.

The project aims to generate, store and retrieve the login passwords using encryption and decryption. We propose to develop an efficient fast and easy to use software which also provides user with a random but strong and secure passwords and passphrases.

1.4 Objectives

The main objectives of our project are listed below

- To design and develop a clean and easy to use password managing software.
- To develop a program that also generates random but strong passwords and passphrases for user.

2. LITERATURE REVIEW

2.1 Previous Projects

2.1.1 Introduction

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.

2.1.2 Existing systemlimitations

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.

2.2 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.

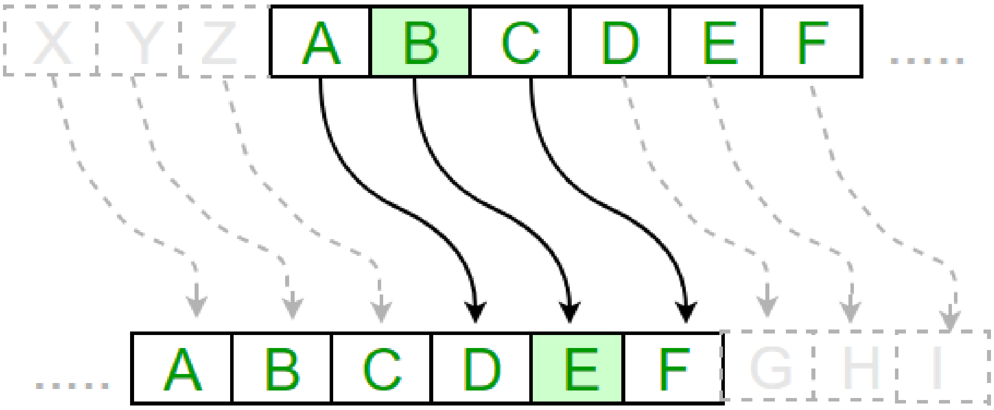


Figure 2-1:Ceaser Cipher technique **[2]**

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]

2.3 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. PROPOSED SYSTEM ARCHITECTURE

This project mostly revolves around the safety and easy accessibility of user data. So, the major entities are file handling, encryption/decryption and SMFL graphics. The programs are 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 userfile.dat and the saved passwords for each users will be stored in separate .dat files after encryption. The password generator will be able to generate required length of random string using different symbols, letters and words for extra security. The different classes made for specific purposes in this application can be summarized as:

Table 3-1: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

3.1 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 logins. 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.

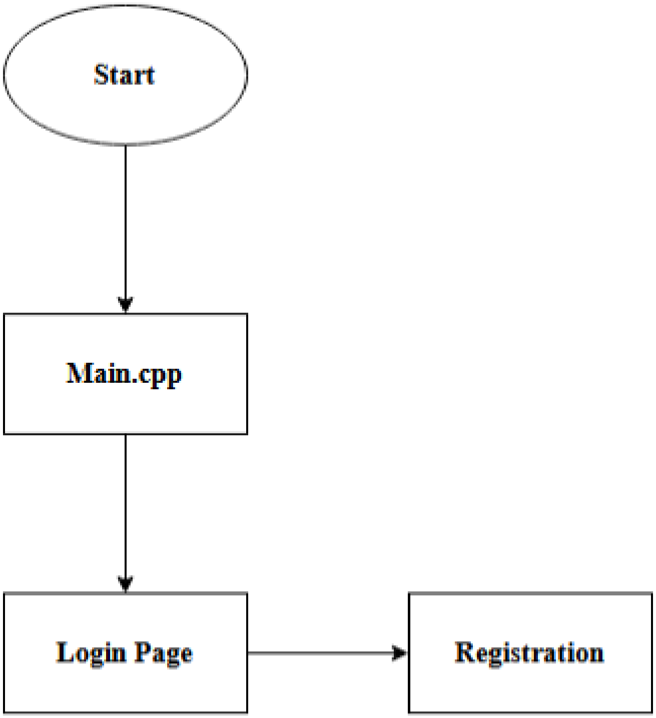


Figure 3-1:Basic layout of application

3.1.1 Login Class

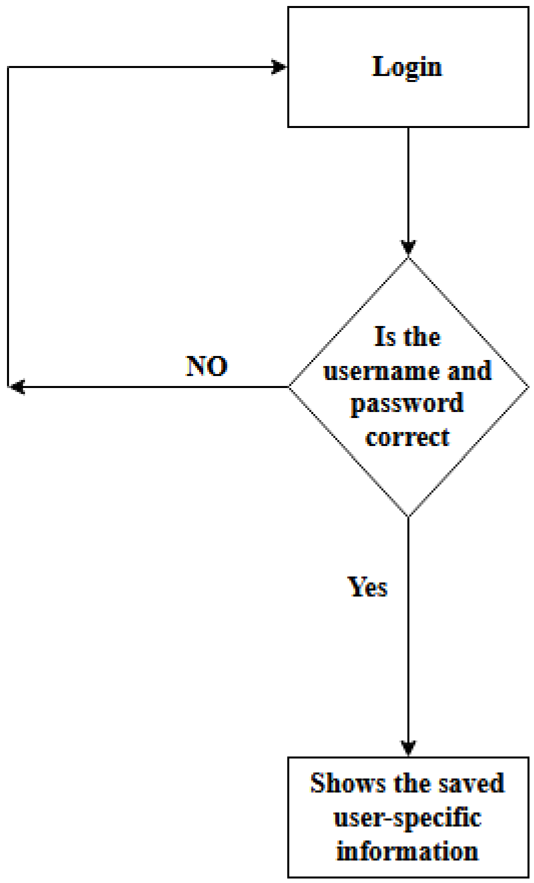

Login
NO
Is the username and password correct
Yes
Shows the saved user-specific information

Figure 3-2:Login Class model

3.1.2 Registration Class

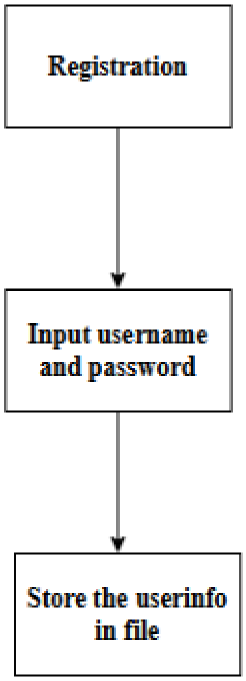


Figure 3-3:Registration Class model

3.1.3 UserInfo Class

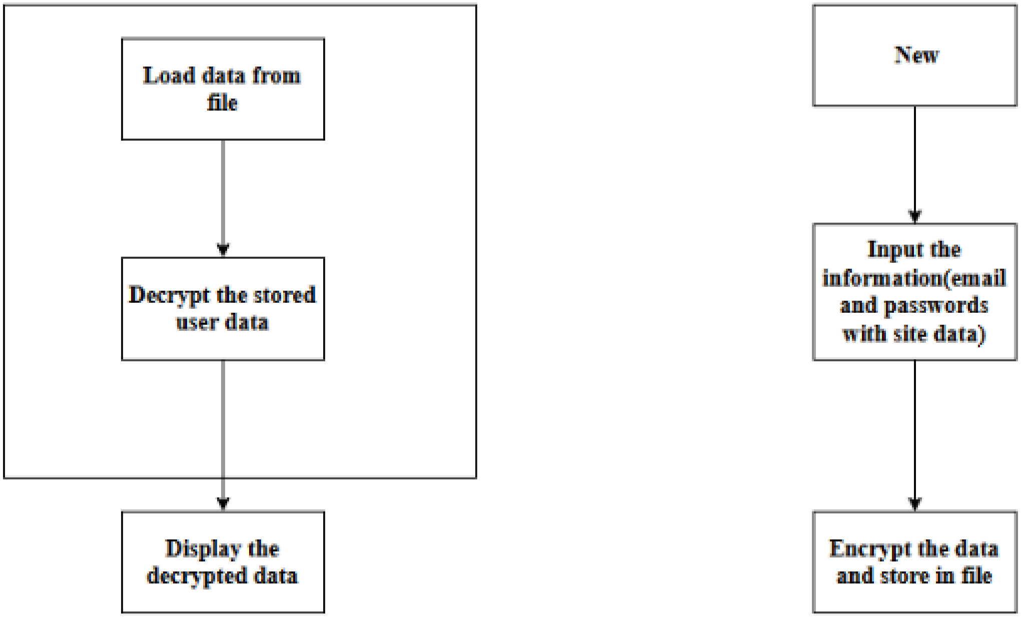

Load data from file
Decrypt the stored user data
Display the decrypted data
New
Input the information(email and passwords with site data)
Encrypt the data and store in file

Figure 3-4:Userinfo Class model

3.2 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 used in the project are classified mainly in two categories: Programming Language and Environment tools.

3.2.1 Programming language and libraries

3.2.1.1 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 are using C++ as major language for the program.

3.2.1.2 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 will be essential in this project for the file management of user data and save files.

3.2.1.3 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 are used in this project.

3.2.2 Environmental Tools

3.2.2.1 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 arises we may change the IDE for more suitable one.

3.2.2.2 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.

3.2.2.3 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.

4. 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.

4.1 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.

4.1.1 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.

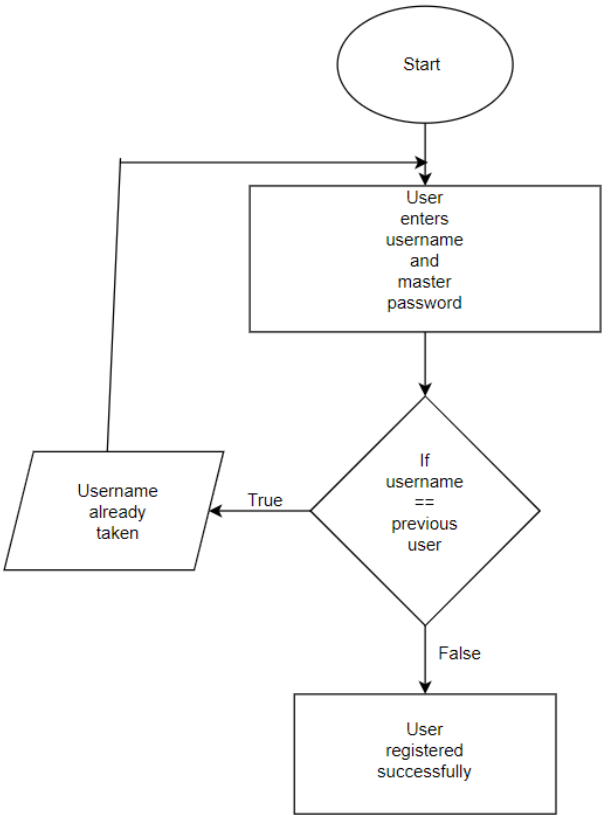

Start
User enters username and master password
If username == previous user
True
Username already taken
False
User registered successfully

Figure 4-1:Block diagram for first time registration

4.1.2 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.

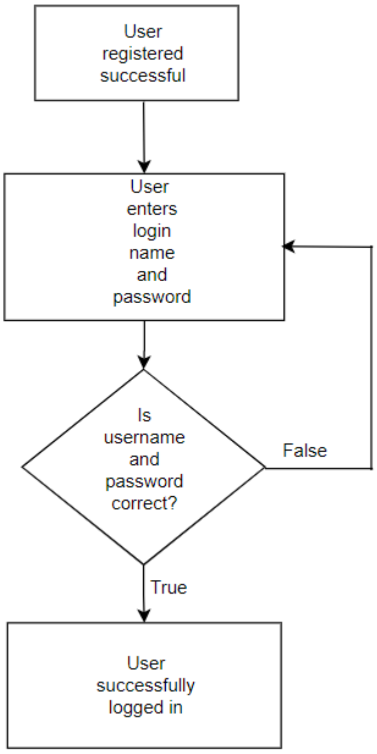

User registered successful
User enters login name and password
Is username and password correct?
False
True
User successfully logged in

Figure 4-2:Block diagram for Login

4.1.3 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.

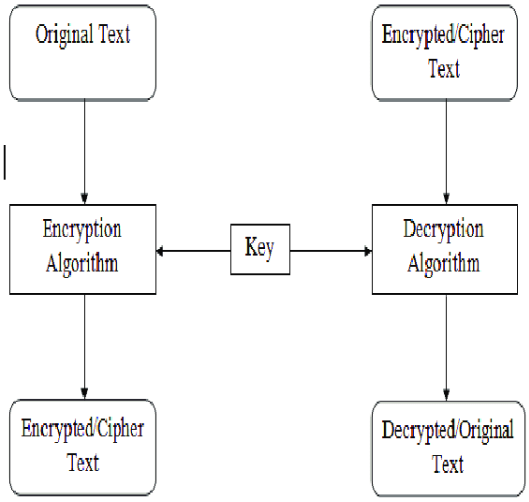


Figure 4-3:Conversion of encrypted password to decrypted and vice versa

4.1.4 Update User data

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

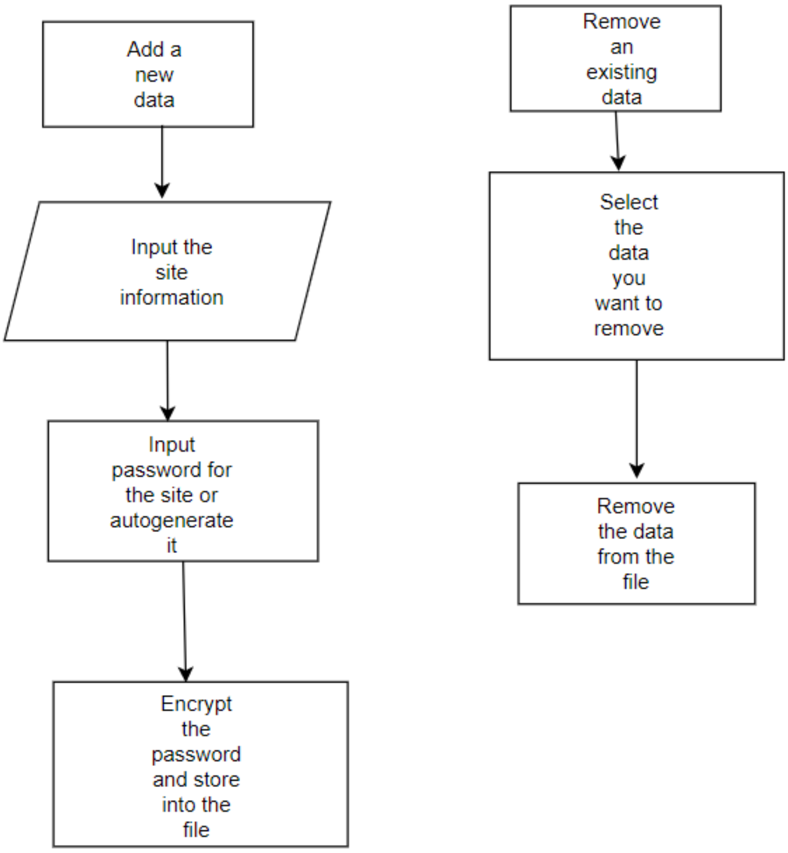

Add a new data
Input the site information
Input password for the site or autogenerate it
Encrypt the password and store into the file
Remove an existing data
Select the data you want to remove
Remove the data from the file

Figure 4-4:Block diagram for Updating User data

4.1.5 Signing out

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

4.2 Identifying subsystems

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

4.2.1 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.

4.2.1 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.

4.2.2 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.

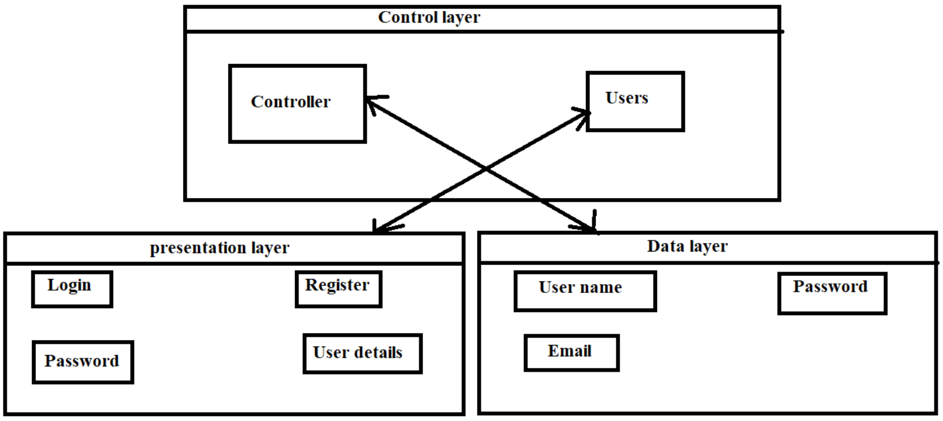


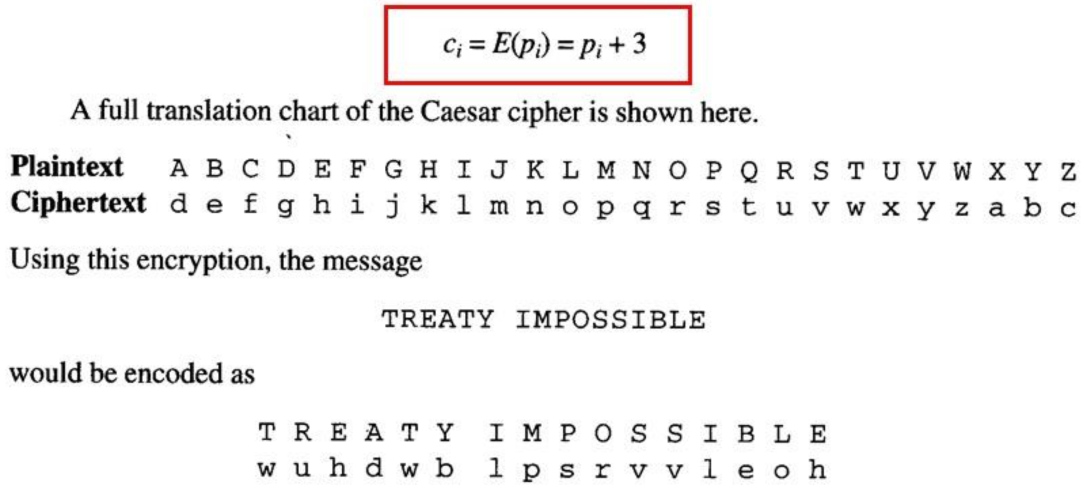
Figure 4-5:Block Diagram of control flow between different layers of system

4.3 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.



$$\boxed{c_i = E(p_i) = p_i + 3}$$

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

5. SCOPE AND APPLICATIONS

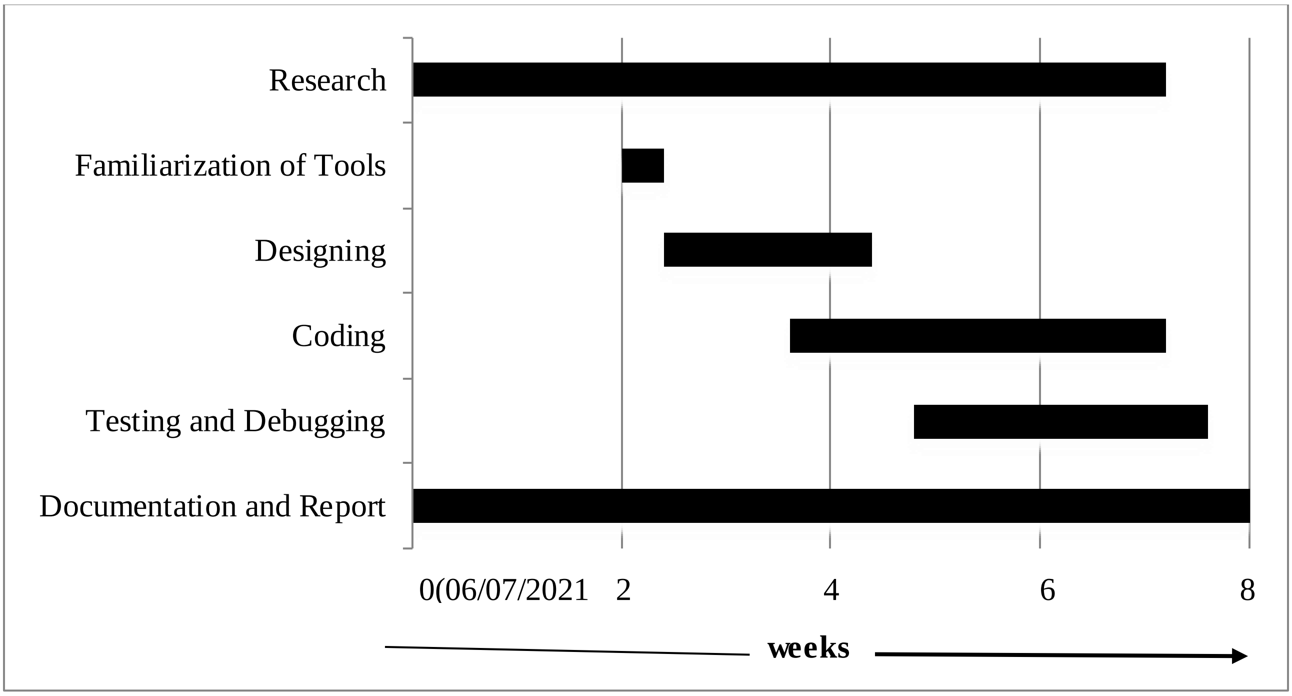
Our project aims for:

- The password manager will do the work of creating unique and complicated passwords needed to help protect the online accounts.
- The program helps to keep tracks of all the passwords created beforehand for different websites.

6. TIME ESTIMATION

A Gantt chart is a graphical representation of a project that shows each activity task as a horizontal bar whose length is proportional to its time for completion. A Gantt chart for the project deliverables within time frame. This project Gantt chart is shown below:

Table 6-1:Gnatt chart



7. FEASIBILITY ANALYSIS

The development of software is disturbed mainly by scarcity of hardware resources, software resources and time constrain. So, it is necessary for us to think about future outcomes while checking the feasibility of the program at the very beginning of the development of the program. The three considerations involved in the feasibility analysis are:

7.1 Economic feasibility

This procedure is to determine the benefits and the expected savings from the developed program. Here, we compare the cost of production which includes the cost increased or decreased due to the use of external resources as well as the earnings that may result in benefit for the group. If anything is not found as the plan, then we edit the proposed system and make sure that the economic feasibility of the program is maintained.

As for now, in our project we do not expect any feasibility costs to be spent on as in this program we have only used the open source resources as they are easily available.

7.2 Technical feasibility

Technical feasibility focuses on the existing resources such as hardware, software. It also focuses on the extent to which the available resources can be used and if the budget is found as a serious restriction in the completion of the project then the project is judged to be not feasible. Here in the case of our project we have used CODEBLOCKS to write the code and used the Windows OS as a platform whereas we have only used the open resources available. We need members to have knowledge of database, structure and sfml library in C++.

7.3 Operational feasibility

People have been known to like the programs that take less space, processing time, that is easy to configure/install and that is entertaining as well as does not stress them out. So, keeping that in mind here in our project, we have made our program easy to configure as well as easily executable regarding the comfort of future clients. The technical background required for the sensors of the game or the I/O are the basic devices that a computer system needs to function properly. So, regarding the operational feasibility the is feasible if the user has basic knowledge of Code block and the program.

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," in *The Basics of Information Security*, 2014.

[4] N. Maccha, "Network Security and cryptography," Nikhil Maccha, 2020.