De La Salle University- Manila
Gokongwei College of Engineering

PROLOGI
Programming Logic and Design

Project Proposal

PCCM: Password Checker, Generator, and Manager

Brent Calado
Ezekiel Santiago
James Tepace

Project Description

  The program aims to test the strength of user-generated passwords, as well as create computer-generated passwords if the user-generated passwords are weak and store them through the program's password managing capabilities. Through this project, certain issues regarding network security may be resolved.

  To be more specific, the program starts by first asking for the user's login credentials for a specific app or website. The login credentials include the username/email/phone number and password. After acquiring the credentials, the program will then move on to check the password strength. Its basis will be on specific criteria such as the length, complexity, and use of special characters of the password. If the program deems the password as weak, it will then display a prompt for the user if they want the program to generate a new password for them or if they want to generate the password themselves. Subsequently, once the new password is generated, the program will then store the log-in credentials of the user and manage present and future

The technical objectives includes:
1. Developing an algorithm to test the strength of the user-generated passwords
2. Implementing encryption and other security measures to protect user data from unauthorized access.
3. Providing feedback to users about the strength of their passwords and suggestions for improving them.
4. Creating a password management system that can securely store and retrieve passwords.

The project will be done in several phases, starting with research and analysis of existing password management systems and techniques. Next, the requirements and specifications for the system will be defined, and the design. Then, the system will be implemented and tested, and any bugs or issues will be fixed.

IPO
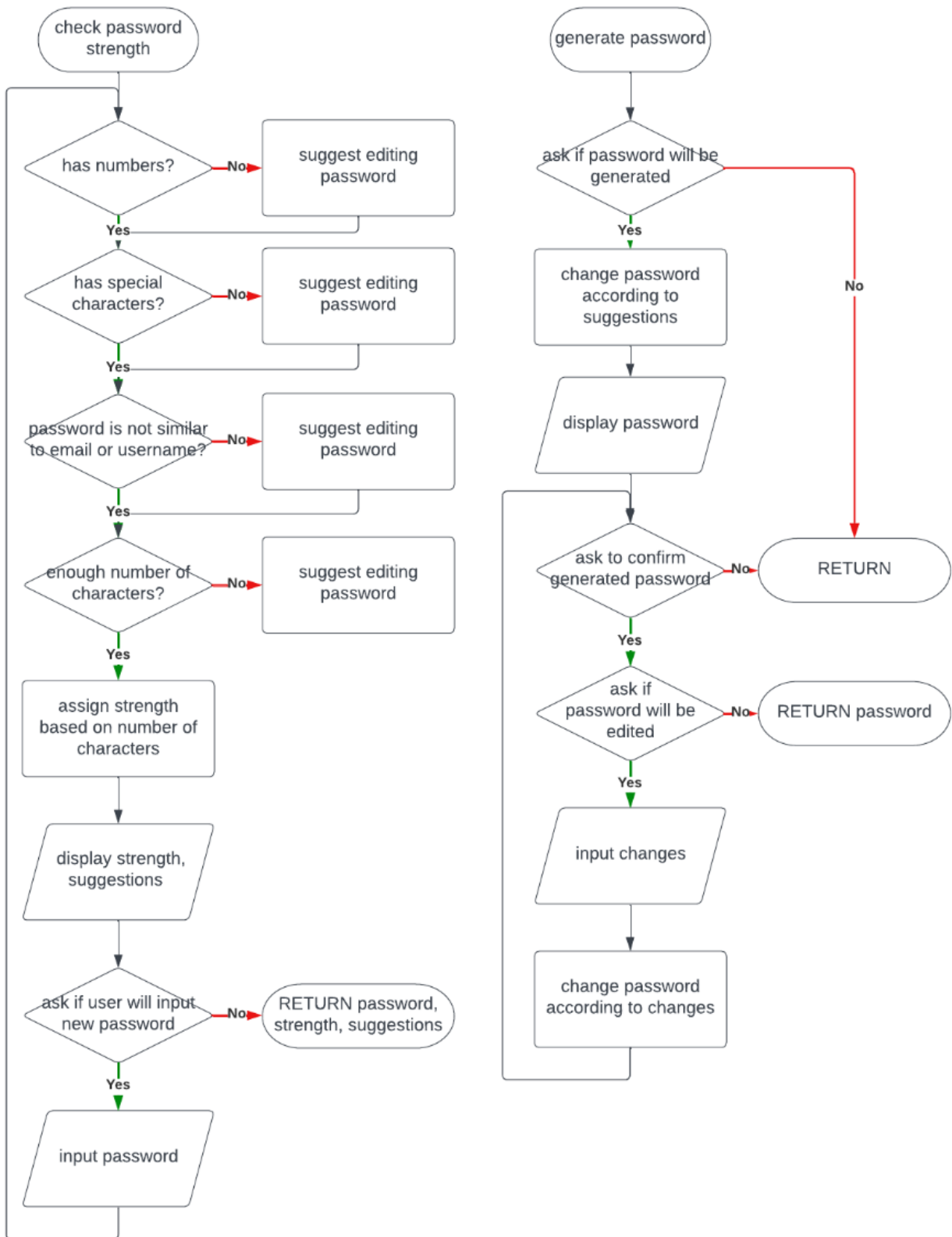
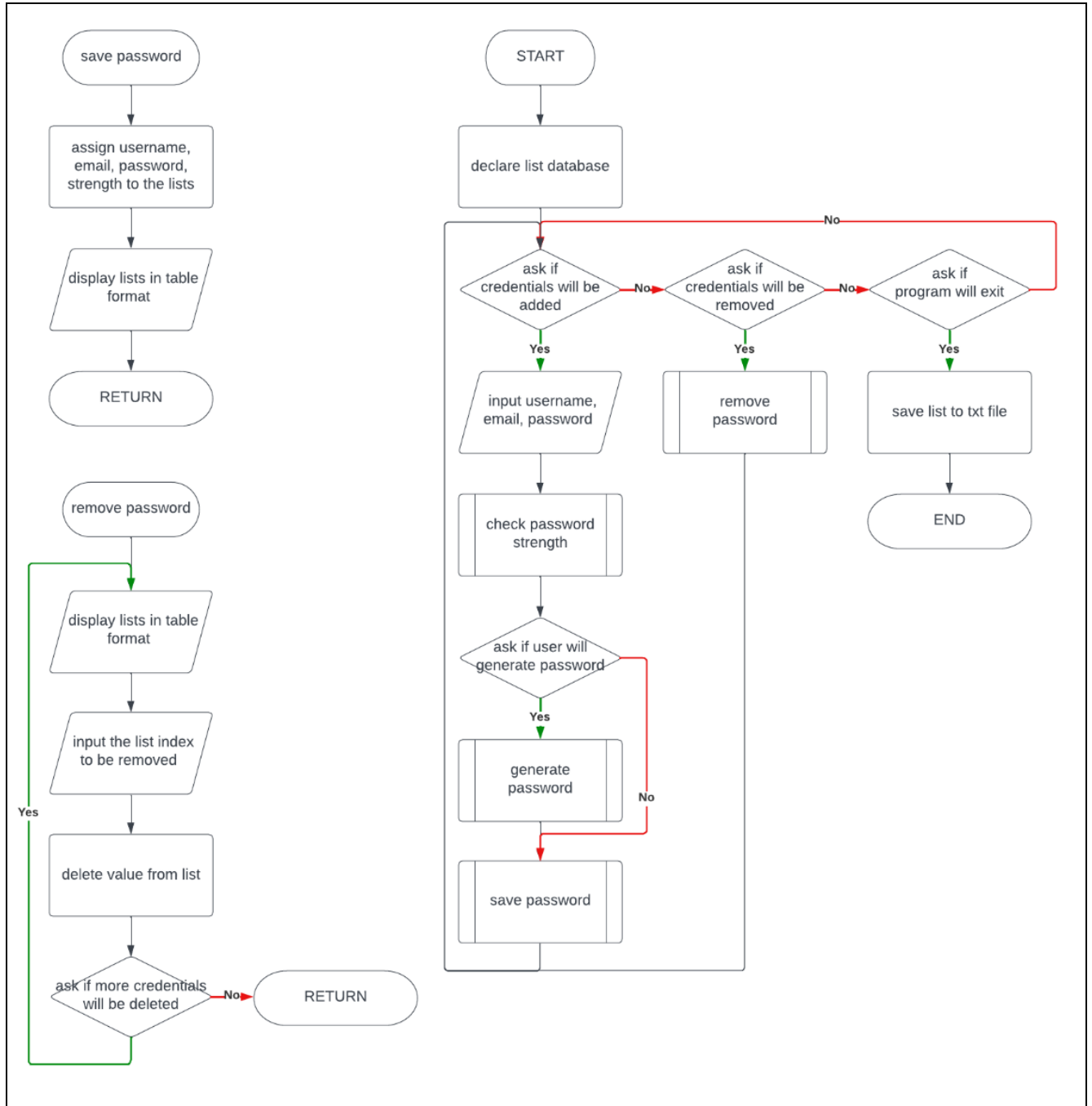| INPUT	PROCESS	OUTPUT
- username - email - password	- determine password strength - suggest changes to it depending on what it needs - generate new password according to the suggestions - create a table out of all inputted credentials - give the option to edit	- password strength - password suggestions - table of credentials - save file

	this table - save the table into a text file	

Methodology

In summary, this program will be done in Python and will utilize decisions, repetitions, modules/functions, lists, string manipulation, and file manipulation. These processes are subject to change, but the general idea is as follows (separate modules are in **bold**):

- The main module will be a loop where the program will display three options to choose from:
- **save a credential**
    - it will ask for the username, email, and password
    - the password will be checked if it follows the set rules:
        - are there special characters?
        - are there numbers included?
        - is it too similar to the username and email?
        - is the length enough?
    - if any of these are satisfied, suggested changes will be displayed
    - the overall strength of the password will also be determined by its length
    - finally, the program will ask to update the password for which this module will be repeated
    - if not, the program will ask if a password will be automatically generated or if it will just be saved as it is.
    - **generate password**
        - if the user chooses to generate a password,
        - the program will use the suggestions it provided to change the password
        - the program will display the new password and will ask for confirmation if it will be saved, re-generated, or further edited by the user
    - **save**
        - Saving the credentials is as simple as assigning them to separate lists with the same indices to keep track of their information.
- **remove a saved credential**
    - The program will display the table of the saved credentials so far
    - it will ask for the index no. of the entry to be deleted
    - the values of that index will be deleted from all the list databases.
    - the program will also ask if another entry will be deleted for which this module will be repeated
- **exit the program.**
    - This option will save the info of the lists into a text file before exiting the main program loop.

check password strength

```
check password strength
        │
        ▼
   ┌─────────────┐   No    ┌──────────────────┐
   │ has numbers?│─────────│ suggest editing  │
   └─────────────┘         │    password      │
        │ Yes              └──────────────────┘
        ▼
   ┌─────────────┐   No    ┌──────────────────┐
   │ has special │─────────│ suggest editing  │
   │ characters? │         │    password      │
   └─────────────┘         └──────────────────┘
        │ Yes
        ▼
   ┌──────────────────┐  No  ┌──────────────────┐
   │ password is not  │──────│ suggest editing  │
   │ similar to email │      │    password      │
   │ or username?     │      └──────────────────┘
   └──────────────────┘
        │ Yes
        ▼
   ┌──────────────────┐  No  ┌──────────────────┐
   │ enough number of │──────│ suggest editing  │
   │ characters?      │      │    password      │
   └──────────────────┘      └──────────────────┘
        │ Yes
        ▼
   ┌──────────────────┐
   │ assign strength  │
   │ based on number  │
   │ of characters    │
   └──────────────────┘
        │
        ▼
   ┌──────────────────┐
   │ display strength,│
   │ suggestions      │
   └──────────────────┘
        │
        ▼
   ┌──────────────────┐  No  ┌─────────────────────┐
   │ ask if user will │──────│ RETURN password,    │
   │ input new        │      │ strength, suggestions│
   │ password         │      └─────────────────────┘
   └──────────────────┘
        │ Yes
        ▼
   ┌──────────────────┐
   │ input password   │
   └──────────────────┘
```

generate password

```
generate password
        │
        ▼
   ┌──────────────────┐   No
   │ ask if password  │────────────────────┐
   │ will be generated│                    │
   └──────────────────┘                    │
        │ Yes                              │
        ▼                                  │
   ┌──────────────────┐                    │
   │ change password  │                    │
   │ according to     │                    │
   │ suggestions      │                    │
   └──────────────────┘                    │
        │                                  │
        ▼                                  │
   ┌──────────────────┐                    │
   │ display password │                    │
   └──────────────────┘                    │
        │                                  ▼
        ▼                          ┌──────────────┐
   ┌──────────────────┐   No       │              │
   │ ask to confirm   │────────────│   RETURN     │
   │ generated        │            │              │
   │ password         │            └──────────────┘
   └──────────────────┘
        │ Yes
        ▼
   ┌──────────────────┐   No   ┌──────────────────┐
   │ ask if password  │────────│ RETURN password  │
   │ will be edited   │        └──────────────────┘
   └──────────────────┘
        │ Yes
        ▼
   ┌──────────────────┐
   │ input changes    │
   └──────────────────┘
        │
        ▼
   ┌──────────────────┐
   │ change password  │
   │ according to     │
   │ changes          │
   └──────────────────┘
```

save password

save password

↓

assign username, email, password, strength to the lists

↓

display lists in table format

↓

RETURN

remove password

remove password

↓ (green)

display lists in table format

↓

input the list index to be removed

↓

delete value from list

↓

ask if more credentials will be deleted —No→ **RETURN**

Yes (loops back up)

START

START

↓

declare list database

—No→ (red, to ask if program will exit)

↓

ask if credentials will be added —No→ **ask if credentials will be removed** —No→ **ask if program will exit**

Yes↓ (green)   Yes↓ (green)   Yes↓ (green)

input username, email, password   remove password   save list to txt file

↓   ↓

check password strength   **END**

↓

ask if user will generate password —No→ (red, to save password)

Yes↓ (green)

generate password

↓ (red)

save password

Schedule of Activities

	Task	Start Date	Duration	End Date	Assigned to
1	Task	Start Date	Duration	End Date	Assigned to
2	Project Description (Proposal)	3/19/2023	4	3/20/2023	James
3	IPO(proposal)	3/19/2023	2	3/20/2023	Ezekiel
4	Methodology(Proposal)	3/20/2023	2	3/20/2023	Ezekiel
5	Password Checker Codes	3/29/2023	0	4/2/2023	Ezekiel
6	Password Generator Codes	3/29/2023	0	4/3/2023	James
7	Password Manager Codes	3/29/2023	0	4/3/2023	Brent
8	Schedule of activities	3/20/2023	1	3/20/2023	James
9	References Finder(Proposal)	3/20/2023	2	3/20/2023	Brent
10	Introduction(Project Documentation)	3/28/2023	0	3/30/2023	TBA
11	Background of the Study(Project Documentation)	3/28/2023	0	3/30/2023	TBA
12	Conceptual Framework/IPO(from Project Documentation)	3/29/2023	0	3/30/2023	TBA
13	Problem Statement(Project Documentation)	3/30/2023	0	4/1/2023	TBA
14	RRL(Project Documentation)	3/30/2023	0	4/1/2023	TBA
15	Hierarchy Chart(Project Documentation)	4/1/2023	0	4/1/2023	TBA
16	Flowchart(Project Documentation)	4/1/2023	0	4/1/2023	TBA
17	Pseudocode(Project Documentation)	4/3/2023	0	4/1/2023	TBA
18	Results(Project Documentation)	4/4/2023	0	4/6/2023	TBA
19	Discussion of Results(Project Documentation)	4/5/2023	0	4/6/2023	TBA
20	Analysis, Conclusion and Future Directives(Project Documentation)	4/5/2023	0	4/7/2023	TBA
21	User's Manual(Appendices Project Documentation)	4/7/2023	0	4/9/2023	TBA
22	Presentation video	4/13/2023	0	4/15/2023	All
23	References Finder(Project Documentation)	3/30/2023	0	3/31/2023	TBA
24					

Editable Excel Link:
https://docs.google.com/spreadsheets/d/1XhQrJhap4hgownAV43tyJQhRnHBMnNEg/edit?usp=share_link&ouid=11106615169439799307&rtpof=true&sd=true

References

GeeksforGeeks. (2023, January 10). *Create a Random Password Generator using Python*. https://www.geeksforgeeks.org/create-a-random-password-generator-using-python/

K, H. (2021, December 24). *How to Create a Password Manager using Python GUI - Huixin K. - Medium*. Medium. https://hxkoey.medium.com/how-to-create-a-password-manager-using-python-gui-92f1f59ef06b

M. (2020, March 25). *Make a password manager with python: Making the basic mechanisms*. DEV Community. https://dev.to/muhimen123/make-a-password-manager-with-python-making-the-basic-mechanisms-46im

Social Network for Programmers and Developers. (n.d.). https://morioh.com/p/b84ed5a0de73

Utilities in Python: Password Management. (n.d.).
https://rowannicholls.github.io/python/utilities/passwords.html

Vineela, M. (2019, June 30). *Check the password strength in Python*. CodeSpeedy.
https://www.codespeedy.com/check-the-password-strength-in-python/#:~:text=match