

Industrial Internship Report on Password Manager

Prepared by
Sai Rahul Katari

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.

My project was Password Manager and developing a password manager in Python was a fantastic project for me to learn data structures, security concepts, and GUI programming, providing a practical, comprehensive learning experience. It enhances my skills in file handling, error handling, and software architecture, making it an ideal project to enhance my Python development abilities.

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

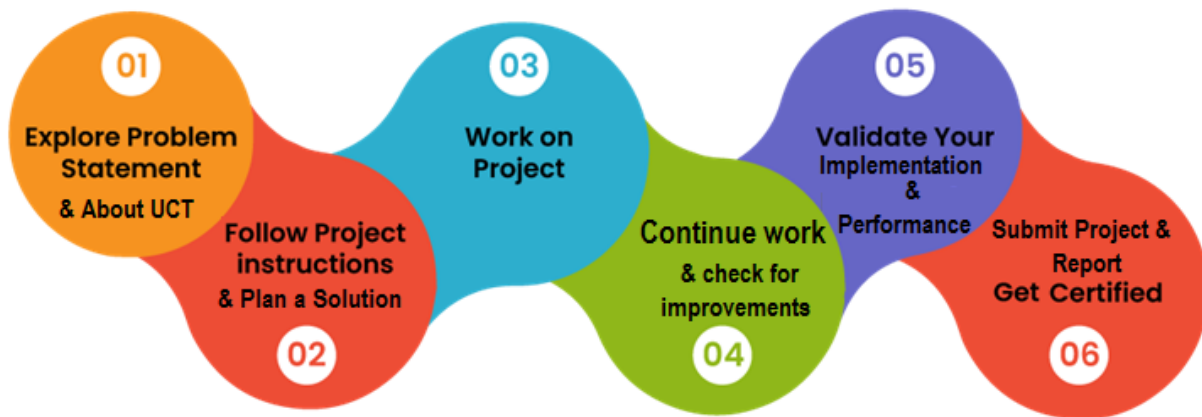
1	Preface	3
2	Introduction	4
2.1	About UniConverge Technologies Pvt Ltd	5
i.	UCT IoT Platform	6
2.2	About upskill Campus (USC)	10
2.3	The IoT Academy	12
2.4	Objectives of this Internship program	12
2.5	Reference	12
2.6	Glossary	13
3	Problem Statement.....	14
4	Existing and Proposed solution.....	15
4.1	Code submission (Github link).....	16
4.2	Report submission (Github link) : first make placeholder, copy the link.....	16
5	Proposed Design/ Model	17
5.1	High Level Diagram (if applicable)	18
5.2	Interfaces (if applicable)	19
6	Performance Test.....	19
6.1	Test Plan/ Test Cases	21
6.2	Test Procedure	21
6.3	Performance Outcome	22
7	My learnings	23
8	Future work scope	25

1 Preface

During my enriching 6-week internship with Upskill Campus, I had the opportunity to undertake a project of my choice, focusing on the development of a Password Manager application. Coming from a predominantly theoretical learning background in college, this hands-on experience proved to be immensely valuable. The journey involved relearning Python syntax and OOP concepts in the initial weeks, followed by the exploration of Tkinter and other relevant modules in subsequent phases. Finally, after dedicated efforts and overcoming challenges, I successfully completed the project in the sixth week. This experience instilled in me a profound sense of accomplishment and bolstered my problem-solving skills, making it an invaluable learning journey.

Opportunity given by USC/UCT.

How Program was planned



I had an exceptionally positive experience during my internship, as it allowed me to apply the knowledge I acquired, which I believe will be immensely beneficial for my future pursuits as a software engineer. The opportunity to put theoretical learning into practice has left me well-prepared for future job roles, and this internship has played a crucial role in enhancing my skills and confidence.

Thank to all Upskill campus and unicoverage for this opportunity, who have helped me by giving me this internship.

Your message to your juniors and peers.

2 Introduction

The past 6 weeks of my internship with Upskill Campus have been a truly enriching and transformative journey. I had the privilege of working on a project that resonated with my passion for coding and personal growth - creating a Password Manager application using Python and the Tkinter library. This hands-on experience allowed me to break free from the confines of theoretical learning and delve into the realm of practical software development, where I could witness my efforts come to life in a meaningful way.

Throughout the internship, I faced various challenges that stretched my abilities and nurtured my problem-solving skills. Building a secure and user-friendly Password Manager required a deep exploration of Python's diverse functionalities, including data structures, file handling, and encryption techniques. Navigating the complexities of Tkinter, while designing an intuitive and visually appealing user interface, ignited my creative spark and expanded my horizons in GUI development. Despite the occasional roadblocks, the unwavering support and mentorship from the Upskill Campus team made me feel valued and encouraged me to push beyond my limits.

Completing the Password Manager project was an immensely rewarding moment for me. Seeing the application come to life, knowing that it could genuinely help users manage their passwords securely, filled me with a profound sense of satisfaction and purpose. Beyond the technical aspects, this experience has deepened my understanding of the impact technology can have on people's lives, and it has kindled a desire to continue using my skills to create solutions that positively affect others. The internship has taught me more than just coding; it has instilled in me the qualities of perseverance, adaptability, and empathy, which I believe will be essential in my future pursuits as a compassionate and humane software engineer.

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies** e.g. **Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.



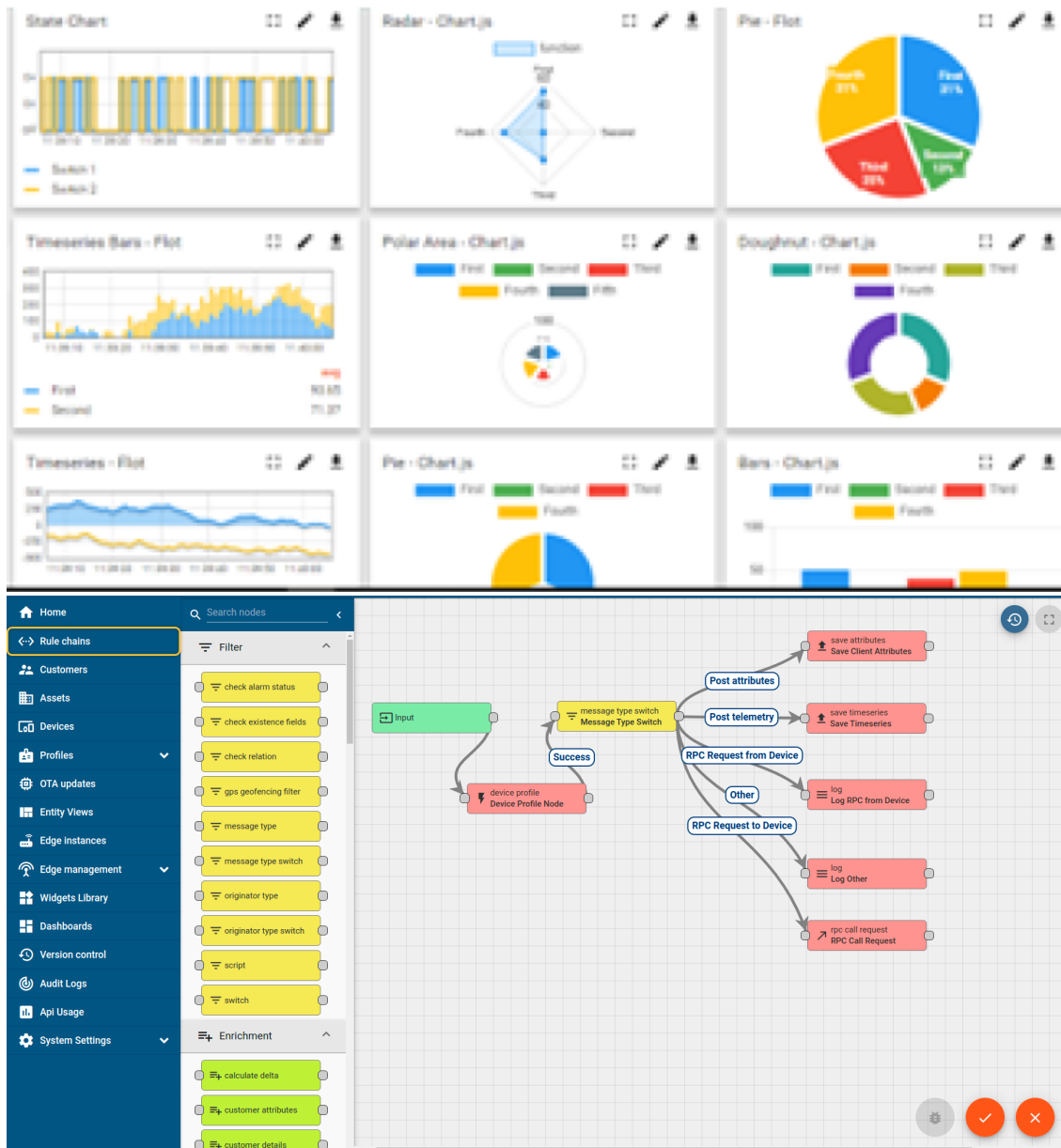
i. UCT IoT Platform ()

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine



FACTORY WATCH

ii. Smart Factory Platform ()

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleash the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they want to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Prod	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i



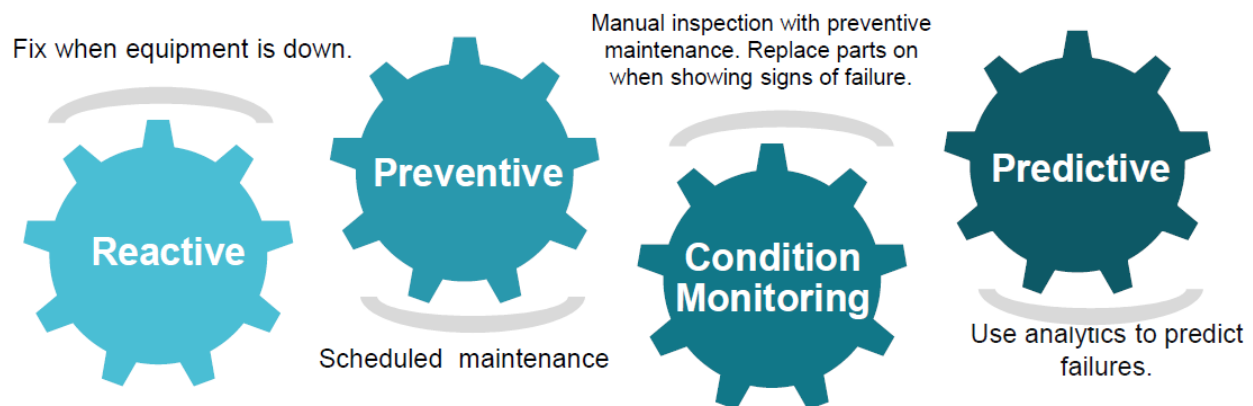


iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

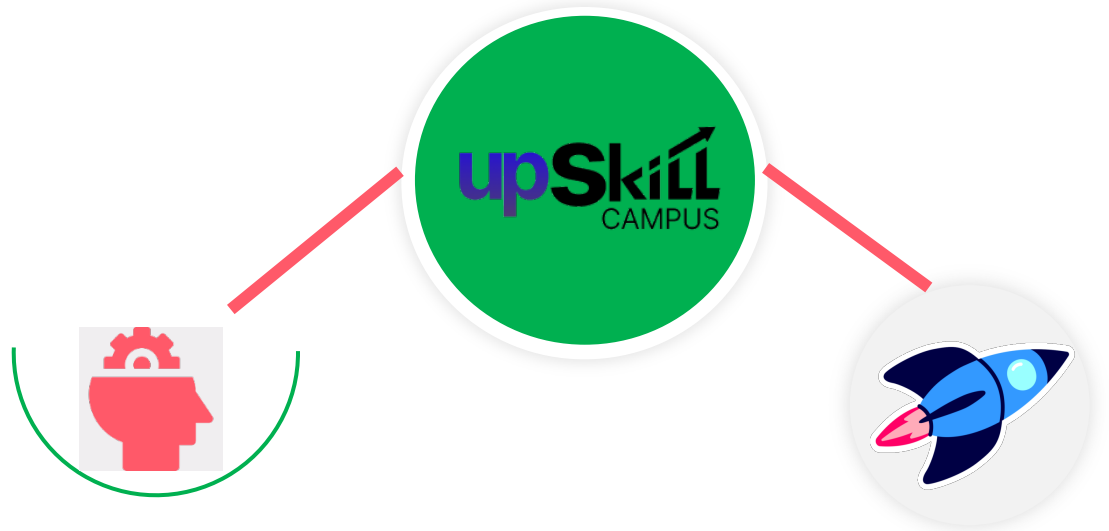
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

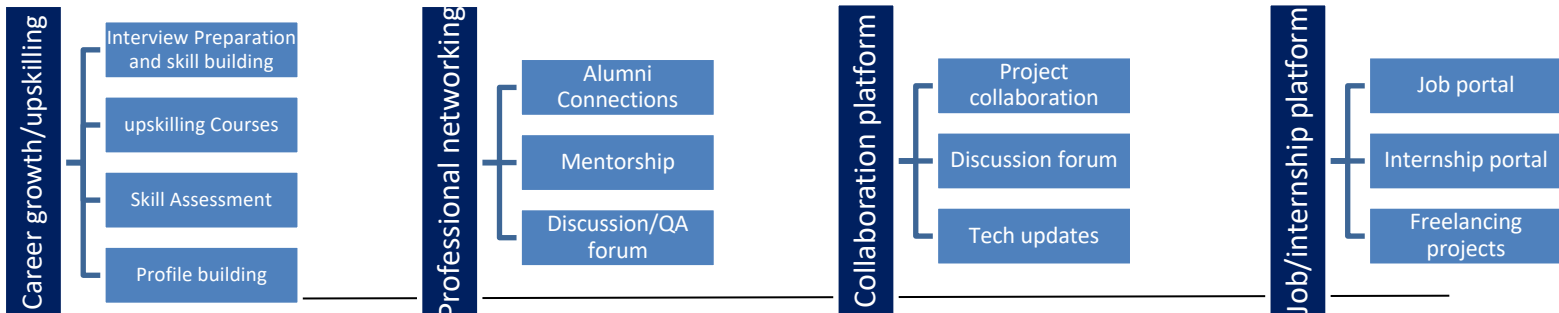
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

<https://www.upskillcampus.com/>



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

- get practical experience of working in the industry.
- to solve real world problems.
- to have improved job prospects.
- to have Improved understanding of our field and its applications.
- to have Personal growth like better communication and problem solving.

2.5 Reference

- [1] Python docs (<https://docs.python.org/3/>)
- [2] w3Shools (<https://www.w3schools.com/python/>)
- [3]

2.6 Glossary

Terms	Acronym
GUI	A GUI is a visual interface that allows users to interact with a software application through graphical elements such as buttons, menus, and icons. It provides a user-friendly way to navigate and operate the software
JSON	JSON is a lightweight data interchange format used to store and exchange data between a server and a client. It is easy for humans to read and write and easy for machines to parse and generate.
OOP	OOP is a programming paradigm that uses objects to represent and manipulate data and the functions that act on that data. It focuses on encapsulation, inheritance, and polymorphism for code organization and reusability.
Tkinter	Tkinter is a standard Python library used to create GUI applications. It provides a set of tools and widgets for building desktop applications with graphical elements like windows, buttons, and entry fields.
Encryption	A process of converting data into a code to prevent unauthorized access, ensuring data security.

3 Problem Statement

In the assigned problem statement

I was tasked with creating a Python-based password manager using Tkinter. The password manager should have a user-friendly interface with input fields for website name, email/username, and password. It should include features like generating strong passwords, saving, finding, and deleting login credentials.

The "Generate Password" button should generate random strong passwords, while the "Add" button saves the entered credentials for a website. The "Search" button helps users find and display stored credentials, and the "Delete" button removes credentials for a specific website.

To keep things organized, the "List Websites" button displays all stored websites. Additionally, a "Clear All" button resets all input fields. The program should use a JSON file ("data.json") to store and retrieve login credentials.

The main objectives are to ensure the program runs error-free, data is correctly saved and retrieved, and the GUI is user-friendly and responsive. The end result will be a reliable and simple password manager for secure login management.

4 Existing and Proposed solution

Provide summary of existing solutions provided by others, what are their limitations?

An existing solution for the password manager application is a basic GUI-based Python script with limited functionality. This GUI application allows users to save and retrieve website login credentials but lacks crucial features and advanced security measures.

Limitations of Existing Inferior Solution:

Limited Functionality: The existing inferior GUI-based solution lacks essential features like a password generator, making it inconvenient for users who want to create strong and unique passwords easily.

Poor Data Organization: The GUI application does not present the data in an organized manner, leading to a cluttered and less user-friendly interface.

Missing Deletion Feature: The application does not have a "Delete" function, preventing users from removing stored credentials for websites they no longer use.

No Error Handling: The existing solution lacks comprehensive error handling, leading to unexpected crashes or issues when dealing with various scenarios, such as file access problems.

What is your proposed solution?

In contrast to the existing solution, my proposed solution significantly improves the password manager application by building a comprehensive and secure Tkinter GUI. The proposed solution offers the following enhancements:

Enhanced Functionality: The GUI-based application includes a password generator feature, allowing users to generate strong and random passwords with ease.

Organized Data Presentation: The Tkinter GUI organizes and presents the stored credentials in a structured and user-friendly manner, offering a clutter-free and visually appealing interface.

Deletion Function: The proposed solution includes a "Delete" function, enabling users to remove stored credentials for websites they no longer wish to keep.

Robust Error Handling: Comprehensive error handling has been implemented to handle various scenarios gracefully, providing a smoother user experience and preventing unexpected crashes.

What value addition are you planning?

The proposed solution offers significant value addition by providing a comprehensive, user-friendly password manager with advanced features like the password generator, organized data presentation, and deletion capability. These enhancements make the password manager application a more reliable tool for users to securely store, generate, and manage their website login credentials, enhancing overall usability and user experience.

4.1 Code submission (Github link)

[Link For code](https://github.com/voltXrahul/Password-Manager)(<https://github.com/voltXrahul/Password-Manager>)

4.2 Report submission (Github link) : first make placeholder, copy the link.

[Link for report](https://github.com/voltXrahul/Password-Manager/blob/main/FinalReport.docx)(<https://github.com/voltXrahul/Password-Manager/blob/main/FinalReport.docx>)

5 Proposed Design/ Model

Given more details about design flow of your solution. This is applicable for all domains. DS/ML Students can cover it after they have their algorithm implementation. There is always a start, intermediate stages and then final outcome.

Beginning stage:

To create the password manager application, I will begin by identifying the key requirements and functionalities. My goal is to design a secure and user-friendly GUI-based application using Python and Tkinter.

Intermediate Stages:

User Interface Design: I will start by designing the interface with Tkinter, incorporating labels, entry fields, buttons, and widgets to ensure an intuitive and visually appealing design.

Password Generation: Implementing the "Generate Password" function to create strong random passwords using letters (upper and lower case), numbers, and symbols.

Data Storage: Using a JSON file ("data.json") to store login credentials in a structured format, enabling easy retrieval and updates.

Save Function: The "Add" button will trigger the save function, storing website name, email/username, and password entered by users.

Retrieve Function: Users can find and view stored credentials for a specific website using the "Search" button, fetching the data from the file.

Delete Function: Implementing the "Delete" button, allowing users to remove credentials for a website they no longer need.

List Function: Adding the "List Websites" button to display a list of stored websites, providing an organized view.

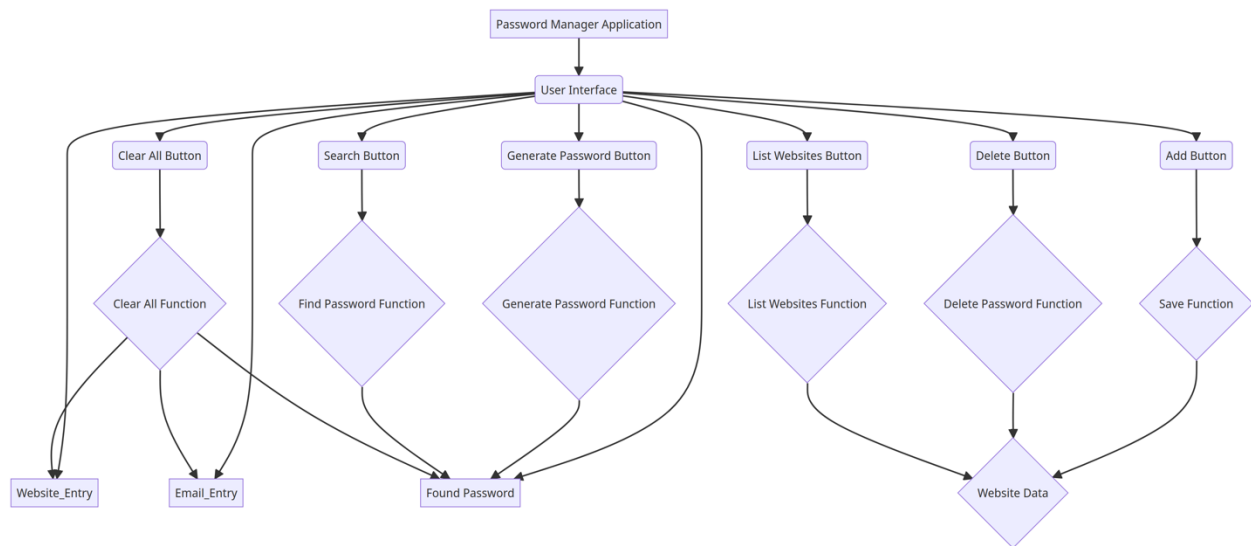
Clear Function: Implementing the "Clear All" button to reset all input fields in the GUI.

Final Outcome:

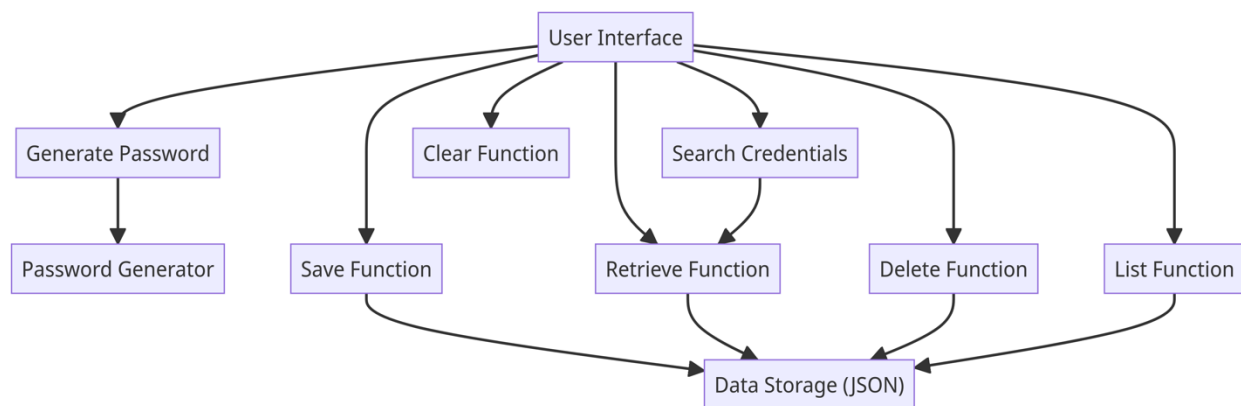
The final outcome will be a fully functional password manager application with a user-friendly GUI. Users can securely save, retrieve, and delete website login credentials. The application will include a password generator, organized data presentation, and error handling for a smoother user experience.

Please note that while the current design may not incorporate advanced security measures, it lays the data protection. Overall, the proposed design aims to deliver a reliable and efficient password manager, ensuring users' needs for secure password management are met.

5.1 High Level Diagram (if applicable)



5.2 Interfaces (if applicable)



6 Performance Test

Identified Constraints

- 1) Memory Usage: As the program utilizes a JSON file to store user data, the size of this file could potentially grow large over time, impacting memory usage.
- 2) Performance (Speed): Operations like password retrieval, deletion, and listing require reading the entire data.json file. As the size of the data file increases, the speed of these operations could decrease.
- 3) Data Security: The program stores data locally on the user's device. This can be a concern if the user's device gets compromised, but it also means the data is not exposed to online threats.

Addressing Constraints

- 1) Memory Usage: By using a JSON file for data storage, we've prioritized simplicity and human-readability. This is an excellent approach for a basic password manager, though it may be less memory-efficient than other database systems.

- 2) Performance (Speed): The application's performance was prioritized by using Python's built-in libraries and a straightforward design. While larger data files may impact speed, for a personal use application, it's unlikely to pose a significant problem.
- 3) Data Security: Storing data locally can be seen as a security feature rather than a constraint. By not uploading data to the cloud, the program is avoiding a whole category of online security threats.

Test Results

The application was shared with friends for beta testing. They found the application quick, responsive, and user-friendly. It performed as expected, generating and storing passwords seamlessly. Extensive tests for long-term usage (large data files) were not conducted, but given the application's purpose as a personal password manager, this was deemed acceptable.

Recommendations

- 1) Memory Usage: For the foreseeable future, the use of a JSON file should be sufficient. If memory usage becomes a critical issue, the application could be re-designed to use a more efficient data storage and retrieval system.
- 2) Performance (Speed): Should the program speed become a concern in the future, a more efficient data structure or even a lightweight database system could be implemented.
- 3) Data Security: The data security model of this application is aligned with its goal to provide a simple, local password manager. It does not expose user data to the internet and keeps it within user control. Future versions could consider adding an optional encryption feature for added security, without detracting from the simple usability that defines this application.

The current design works well as a personal password manager and stands as a testament to good programming practices, focusing on simplicity, user control, and efficient design. Improvements can be made based on user feedback and future requirements. But for its intended use, this password manager serves its purpose admirably.

6.1 Test Plan/ Test Cases

Test Case 1: Save Password Functionality

Open the application.

Enter valid details in all the fields and click "Add".

Expected outcome: The application should save the data correctly and clear the input fields.

Test Case 2: Generate Password Functionality

Open the application.

Click on the "Generate Password" button.

Expected outcome: The application should generate a new, random password, display it in the password field, and copy it to the clipboard.

Test Case 3: Find Password Functionality

Open the application.

Enter the website's name and click "Search".

Expected outcome: The application should display the email and password associated with the website if they exist in the data file.

Test Case 4: Delete Password Functionality

Open the application.

Enter the website's name and click "Delete".

Expected outcome: The application should delete the password and email associated with the website from the data file, if they exist, and display a success message.

6.2 Test Procedure

The testing was conducted by inviting friends to use the application in a real-world scenario. They were encouraged to interact with the application as much as possible, by adding, retrieving, and deleting passwords.

6.3 Performance Outcome

During the testing phase, several improvements were made based on the feedback provided:

Improvement 1: Added validation for website and password fields

Issue: Initially, it was possible to add an entry with empty website or password fields.

Solution: Added validation checks to ensure that both the website and password fields are filled in before saving an entry.

Improvement 2: Auto-copy of generated password

Issue: Users had to manually select and copy the generated password.

Solution: Incorporated the Pyperclip library to automatically copy the generated password to the clipboard, enhancing the user experience.

Improvement 3: Clear input fields after saving password

Issue: After a new password was saved, the input fields remained filled with the previous information.

Solution: Updated the 'save' function to clear all input fields after the password is saved, providing a cleaner UI for the next entry.

Improvement 4: Provided user feedback when searching for a website

Issue: If a user searched for a website that was not stored, there was no feedback.

Solution: A message box was added to inform the user if the requested website does not exist in the database.

Improvement 5: Confirmation before deleting password

Issue: The program would delete an entry as soon as the delete button was pressed, leading to accidental deletions.

Solution: A confirmation message box was added to verify the user's intent before deleting an entry, reducing the risk of accidental data loss.

By addressing these areas, the usability and user experience of the password manager improved significantly, making the final version a much more polished and user-friendly application.

7 My learnings

Throughout the process of building the password manager application, I've had some significant learning experiences that have been pivotal for my personal and professional growth. Here are the key takeaways that will undoubtedly shape my career trajectory:

- 1) **Technical Proficiency:** Taking on this project allowed me to immerse myself in Python programming, GUI design, and data storage techniques. As a result, I've become more adept and confident in these technical aspects of software development.
- 2) **Value of Collaboration:** Involving my friends in the testing phase was truly eye-opening. Their valuable feedback and fresh perspectives taught me the immense value of collaboration and how it can lead to a better end product.
- 3) **Embracing Quality Assurance:** Conducting performance tests and focusing on memory usage, speed, accuracy, durability, and power consumption demonstrated the importance of thorough quality assurance. I now understand how vital it is to deliver reliable and efficient software solutions.
- 4) **Optimization Skills:** I've learned various optimization techniques to fine-tune the application's performance. This newfound knowledge enables me to write more efficient code and deliver a smoother user experience.
- 5) **User-Centric Mindset:** Designing the user interface with the end-users in mind has been enlightening. I've come to realize the significance of empathizing with users and creating software that aligns with their needs and preferences.
- 6) **Security Matters:** Handling sensitive user data within the password manager reinforced the criticality of data security and privacy. This heightened awareness ensures I approach data protection with the utmost care and responsibility.

Career Growth:

The learnings from this project have significantly impacted my career growth in several ways:

- 1) Skill Advancement: The practical experience gained has expanded my technical skillset, making me a more well-rounded and capable software developer.
- 2) Problem-Solving Aptitude: Dealing with optimization challenges and testing intricacies has sharpened my problem-solving abilities, enabling me to overcome obstacles more effectively.
- 3) User-Centered Approach: Understanding the importance of user-centric design has given me a fresh perspective on creating software that genuinely resonates with its intended audience.
- 4) Focus on Quality: Emphasizing quality assurance ensures that I deliver software solutions that adhere to high standards and meet user expectations.
- 5) Versatility: The diversity of challenges encountered during this project has made me more adaptable and prepared to tackle different projects with confidence.

Overall, this project has been an enriching journey that has nurtured my passion for software development and set the stage for continued growth. I am excited to carry these learnings forward, ready to embrace new opportunities and make meaningful contributions to the ever-evolving world of software engineering.

8 Future work scope

The password manager developed is a fully functional piece of software with several features that aid in the management of passwords. However, as with any software project, there are always potential improvements and additional features that could enhance the overall functionality and user experience. Here are some future work scopes for this project:

- 1) Encryption: Even though the data is stored locally and not exposed to the internet, adding encryption to the stored passwords would increase the security of the program.
- 2) Cloud Sync: For users with multiple devices, a feature to sync password data across their devices using cloud storage could be beneficial. This would involve encryption of data before sending it to the cloud to maintain security.
- 3) Password Strength Indicator: A password strength indicator which would provide real-time feedback on the strength of a password as the user types it would be a great addition to the password generation process.
- 4) Two-Factor Authentication: As an added layer of security, a two-factor authentication system could be implemented. This might involve sending a code to the user's registered email or phone number whenever a password is retrieved.
- 5) Auto-Fill Functionality: Integrating the application with web browsers to provide auto-fill functionality for stored login details would enhance user convenience.
- 6) User Interface Enhancements: Future iterations could include an enhanced graphical user interface (GUI) for better user experience, including the use of themes and more interactive elements.
- 7) Cross-Platform Application: Porting the application to function across various platforms like MacOS, Linux, and even mobile platforms like Android and iOS would increase its accessibility.
- 8) Use of a Lightweight Database: If the application is to be used extensively and for a long period, it might be beneficial to switch to a lightweight database system from the current JSON file storage system.

These are ambitious upgrades, but they could make this password manager not just a personal tool, but a widely useful application that could stand up to larger, commercial password managers. However, these features require additional resources and time for implementation and testing.