الجمهورية الجزائرية الديمقراطية الشعبية
The People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة محمد بوضياف بالمسيلة
University Mohamed Boudiaf of M'sila

كلية الرياضيات والإعلام الآلي
Faculty of Mathematics and Informatics

قسم الإعلام الآلي
Department of Computer Science

Domain: Mathematics and Computer Science

Thesis Presented to Fulfill the Partial Requirement
for Master's Degree in Computer Science

Specialty: Information Systems and Software Engineering

Prepared By: Elbar Zakaria Mabkhout, Khier Yacine

Supervised By:

Dr. Loucif Hamza

Prof. Linda Belabdelouahab-Fernini

ENTITLED

Cryptool Account Manager

Jury Members

Dr. Chikouche Noureddine	President
Dr. Loucif Hamza	Supervisor
Prof. Linda Belabdelouahab-Fernini	Supervisor
Dr. Tharafi Abdellah	Examiner

Academic Year 2023/2024

Dedications

"I dedicate this thesis to my beloved parents, my sisters, and to all those who are dear to me.

Elbar Zakaria Mabkhout

"This thesis is dedicated to my beloved family, especially my mother, who was very supportive and helpful during stressful times. Thank you for all your support."

Khier Yacine

Acknowledgments

We are deeply grateful to our thesis supervisor and advisor, Dr. Loucif Hamza and Prof. Linda Belabdelouahab-Fernini for their unwavering support and inspiration, which have been pivotal to the success of this project. We especially appreciate their dedication and assistance throughout this challenging study.

Special thanks to our friends and families for their patience and understanding during the stressful moments of our research.

Table ofContents

List of Figures

List of tables

List of acronyms

General Introduction

With the fast evolution that the world is witnessing nowadays, and the huge leap in data processing and storage services, many businesses have completely migrated to the digital space or 'The Cloud' to be able to provide different services to users efficiently without having to be constrained by local resources. In most cases, for a user to be able to access an online service, he must create an account to subscribe to that service whether it is free or paid. Every account consists of a username which is usually an email and a password that must be difficult to guess so that no one except the user can access his account. Due to the large number of services that require an account and the password complexity required for account security, many users tend to use an online password manager that stores all their credentials on an external server. But on the other hand, this will make it a target for attackers and will increase the chances of data getting breached.

In this thesis we are going to implement an application "digital safe" that will enable users to save their credentials easily, having different and complex passwords without the need to remember them, while securing all data on a file that is encrypted using a custom encryption algorithm and saved on the user's local machine (offline) which makes it extremely difficult for an intruder to access.

The application is developed on the windows operating system, and the main algorithm for the data encryption, which is the core of the app, is a custom encryption algorithm augmented using the popular hashing algorithm SHA-256. The test of the encryption algorithm may not be scalable due to time constraints and the lack of professional testing tools.

This thesis is structured into six main chapters. Chapter one deals with the literature review and overview of other password manager tools and an analysis of some of them with a brief review of some encryption algorithms. Chapter two tackles the conceptual foundation and principles of secure credentials and local file storage. After that, the design phase and algorithm architecture are presented in chapter three. This is followed by the fourth chapter with an overview of the implementation process and selection of tools and technologies used in the development process. As for chapter five, it presents the testing and evaluation process, which details the evaluation of the use cases and usability. Finally, the last chapter shows a discussion with a comparison between the implemented tool and the existing solutions, its strengths, limitations, and future enhancements.

CHAPTER 1
Literature review

1.1. Introduction

With the increasing number of online businesses, social media platforms and other online services, users have to manage numerous passwords daily and that can be stressful and overwhelming. A survey performed by "Bitwarden" brought results from 2400 individuals from the US, UK, Australia, France, Germany and Japan. The survey's findings key takeaways show that "A majority of respondents continue to use memory (54%) and pen and paper (33%) for password management, underscoring a reliance on outdated and potentially insecure practices". [1]

1.2. Overview of some password manager tools

1.2.1. What is a password manager?

A password manager is a tool that stores mainly passwords hence the name, all in one place making the user able to access all his private credentials safely. The core element of a password manager is the implementation of various encryption methods and techniques to prevent the unauthorized access of the user's credentials additionally providing a safe place to store and manage these credentials easily.

Some password managers provide various features such as password generator which suggests randomly generated complex passwords for the user to sign up with on an online service. Some password managers have autofill enabled so when the user starts typing the app suggests to paste stored credentials. Other features include synchronization of credentials across devices and the ability to manage different types of credentials such as credit card numbers, bank account numbers, and addresses. Most password managers provide an organized user interface to handle different credentials and give the users various methods of accessing and monitoring their data.

1.2.2. Types of password managers

When password managers basically do the same functionality, they come in 3 main different types. The first one is cloud-based password managers, which is the most popular choice for individuals. Cloud password managers store and maintain their user's sensitive data in their own servers making it accessible for the users from anywhere in the world using any device that has internet access. The second type is the oldest and most popular one: the desktop password manager. This type of password manager keeps the user's data inside their local machines which limits the accessibility of the data to a single machine, but on the other hand, it increases the security by limiting the potential for a data breach.

The third type is Single Sign-On or "SSO" password managers. This type differentiates from the first two by using one SSO instead of multiple usernames and passwords for each account. While it's a security requirement to use diversity in passwords, SSO uses attributes that are related to the user (ID, IP, Location…) and shares it across trusted sites.

1.3. Challenges in credential storage and management

1.3.1. Data breaches

As companies invest more heavily in their digital infrastructure, cyber-attacks and data breaches have also increased. Statistics from IBM show that 'The global average cost of a data breach in 2023 was $4.45 million, a 15% increase over 3 years.' [2]

A data breach is the release of confidential or sensitive information into an unsafe environment. This usually happens when cybercriminals get unauthorized access to users' data on the cloud because of a cyber-attack, user error, or service vulnerabilities. Data breaches know no bounds and often happen on a large scale affecting global corporations across different sectors and in multiple locations. These breaches can stem from various vulnerabilities affecting all sectors and companies even large corporations, despite their substantial investments in cybersecurity, are not immune, as evidenced by high-profile breaches affecting millions of individuals' personal and financial information.

This is an overview of the number of known breach incidents during each month from the end of 2023 to the first quarter of 2024.

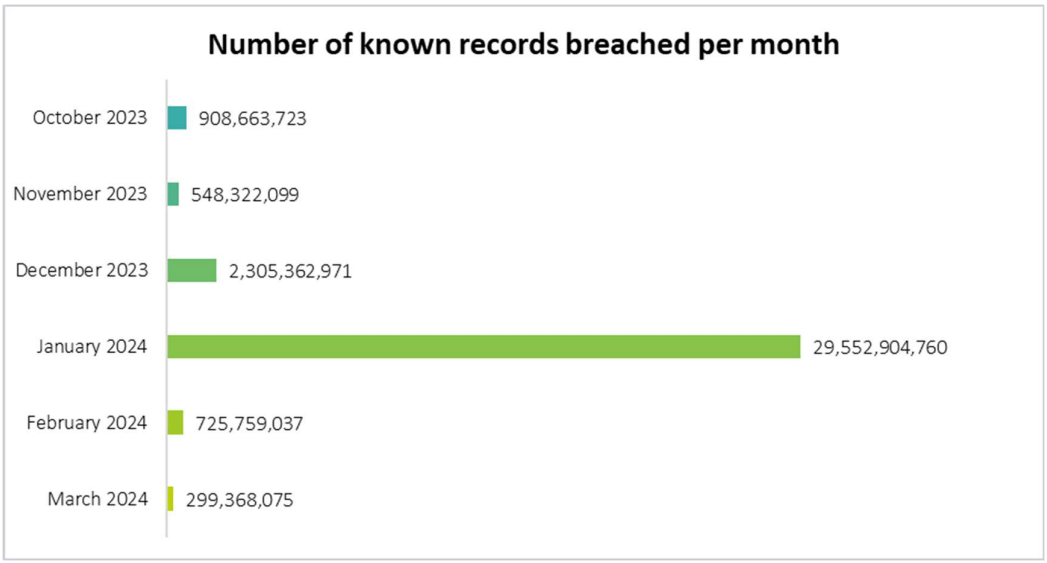

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.1 :** Number of data records breached in Q4 2023 and Q1 2024. [3]

1.3.2. Single point of failure

Users are always suffering from issues regarding the process of managing their passwords, one of which is using a weak password that usually includes some personal information that is publicly available on social media. This is a serious concern when it comes to attackers trying to perform social engineering techniques on the target to steal passwords. In addition, using the same weak password across multiple accounts increases the likelihood of it being cracked in a

data breach and an attacker can easily access other accounts that have sensitive information such as banking details or credit card numbers.
Furthermore, recovering from that type of attack is much more complicated, especially when the user loses access to some of the accounts. When data gets leaked, if multiple accounts share the same password, it makes it harder to identify which service or website was breached initially. This complicates the process of securing accounts and the user needs to update the password for each one, which can be time-consuming and frustrating.

1.3.3. Phishing attacks

Phishing attacks are one of the most common cyber-attacks, where attackers try to deceive victims to steal their credentials or manipulate them into sending their private data. Attackers usually masquerade as legitimate organizations or companies that the users trust.

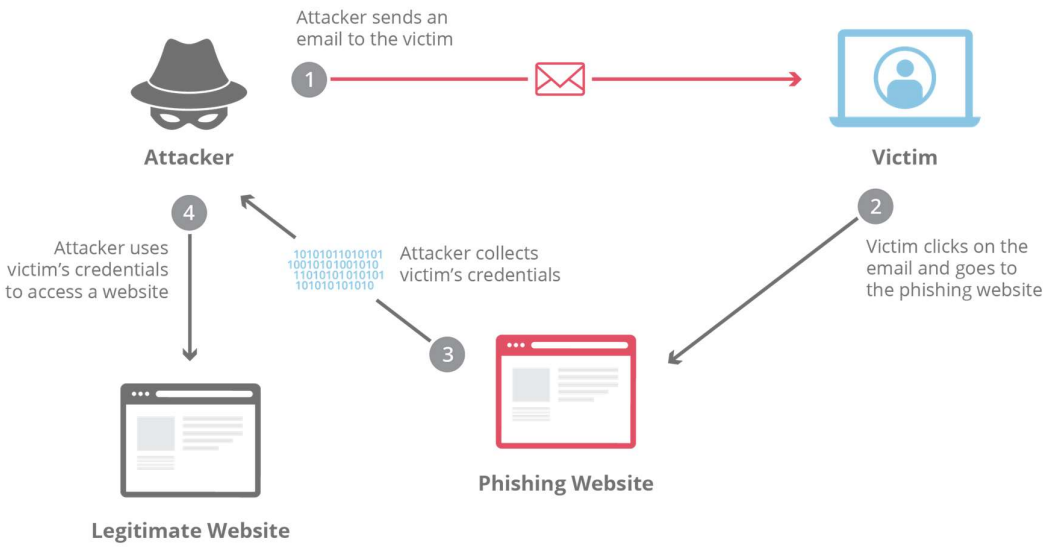

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.2:** Phishing attack diagram. [4]

Attackers use various methods like sending fraudulent emails or SMS messages to convince and trick victims into actions like:

- Downloading files or games containing malware;
- Sharing personal information such as (credit card numbers, bank account details, address, phone number … etc.);
- Access their machines to perform large-scale attacks.

Phishing attacks are the most popular among attackers because it is cheaper to carry out than to do an actual hacking of a network or a system. Typically, attackers use phishing to gain access to large data and corporations by exploiting employees using emails; for example, the attacker tries to deceive a specific employee who has access to the company's financial

information by sending a crafted phishing email and getting access to that employee's credentials then uses the gathered data to perform a large-scale attack on the company.
Phishing attacks can vary depending on the attacker and the victim. Therefore, phishing attacks are divided into many types, we are just going to mention the two most common types.
The first type is Bulk email phishing attack which is the most popular and causing damages to individuals as well as corporations all over the world. An attacker creates an email message that appears to be legitimate from a well-known company or organization. Then he sends a bulk of messages to millions of recipients. In a bulk email phishing attack, the attacker relies on the numbers: The larger the number of recipients the more likelihood to have victims to fall into the trap, and also the more legitimate the message appears the more recipients who are likely to be customers or members.
The second type is Spear phishing, which is a narrower attack that targets a specific individual, usually someone with privileged access to sensitive data or a network of large resource. In a spear phishing attack, the attacker tries to gather information about the target's social media where people publicly congratulate coworkers and colleagues. Then the attacker can send a message containing personal details or financial information to the target followed by a credible request. Most spear phishing attacks that are aimed at high-level executives or wealthy individuals are called "Whale phishing attacks".

1.4. Analysis of existing solutions

There are so many password managers available on the market, that it is unrealistic to analyze them one by one. Also, all password managers share the same core functionality, which is using encryption to secure users' credentials, but each one differs in the algorithm, interface, and dealing with the storage and access of credentials locally or on the cloud. With all this in mind we surveyed to know how most people deal with storing their credentials and what tools they use.

1.4.1. Survey overview

We conducted a survey on how most people access and store their credentials. The survey was posted on April 18, 2024, and gathered insights from 89 individuals from Mohamed Boudiaf University of M'Sila and other universities in Algeria. The survey shows login habits and credentials storing and accessing.
We categorized the questions for the sake of simplicity and conciseness. We did some research and found that most people either rely on a browser to store their credentials, or write them down on a notepad, or a piece of paper, or use a standalone password manager tool.
The questions included how often users type their credentials and what tools they use to store them and whether they use the same password for multiple accounts on the internet. That helped us define the risks and the issues regarding the protection and security of private data.

Here are the key takeaways from the survey:

- The majority of the respondents use Google Password Manager as a way to store their credentials mostly due to its simplicity and popularity.
- 32.6 % of the respondents still use a notepad to store their passwords in a traditional way, which raises concerns and compromises their credential security as their data is in plain text and at risk of access by any intruder.
- Only 15% of respondents use a password manager to store and secure their credentials which requires a critical need for enhanced awareness and education about taking serious actions to develop better security habits.

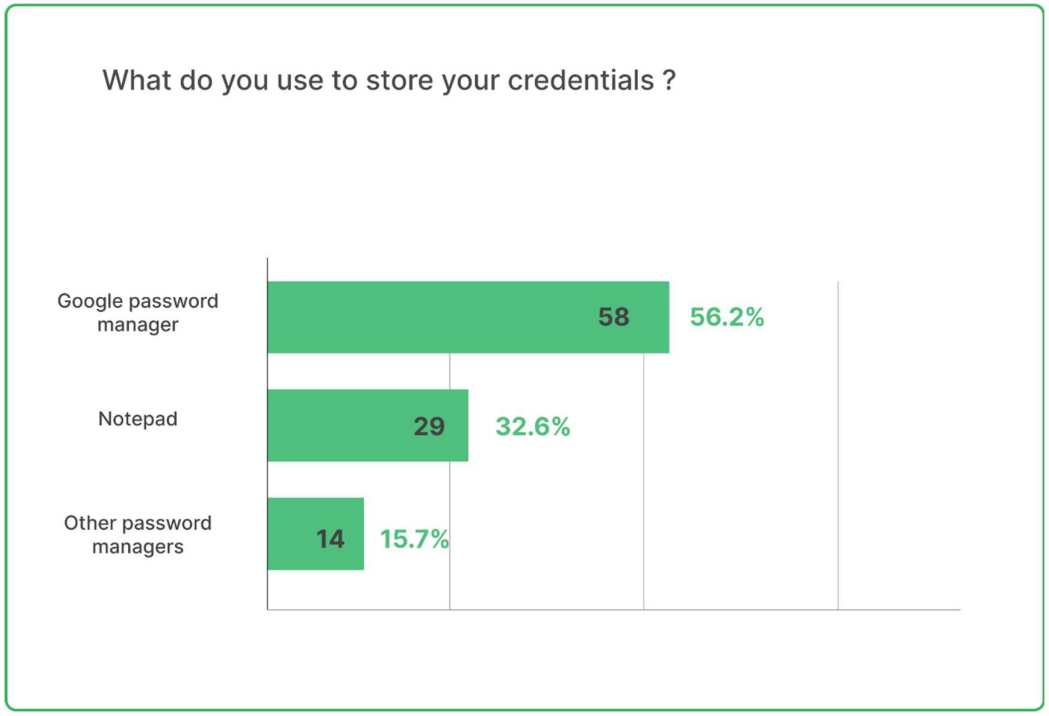

Fig. *Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.3:** Private data security survey April 2024.*

- A third of respondents use the same password for multiple accounts which is a risky habit that can increase their credentials to likely being leaked in a data breach and all of their accounts can be stolen by attackers just from a single account.

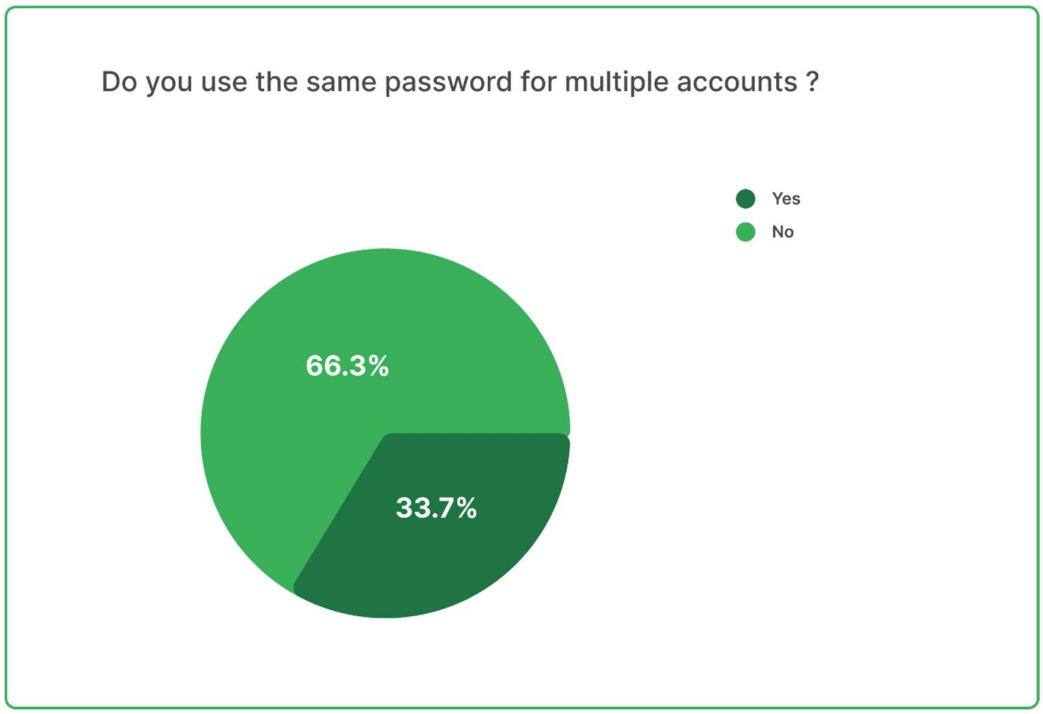

Fig. *Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**4:** Private data security survey April 2024.*

After our survey results, we decided to categorize existing solutions into two, the first being Google Password Manager a popular browser-based password manager that the majority of people use. The second type of solution includes all other third-party password managers which have similar functionalities and interfaces so they all fall into the same category.

We did our research and collected some advantages and disadvantages of both solutions.

1.4.2. Google Password Manager

Google is a large multinational technology and data company that has the most popular internet browser in the world “Chrome”. According to “StatCounter” global statistics between April 2023 and April 2024, chrome had a global market share of 65.42 %. [5]. Chrome has a built-in password manager which allows the browser users to save their credentials once they sign in to their accounts on the web and they can log in without having to remember their passwords.

Google Password Manager has some advantages and features that make users prefer it over other tools, we mention some of them:

- The integration in the browser, since Chrome is the most popular browser in the world most users don’t feel the need to have another application to store their credentials and then have to log to their accounts from Chrome anyway.

- The simplicity of storing and accessing credentials makes it a preferred easy option to secure private data.
- Google synchronization of data between all devices that have a Google account on them makes the process of accessing credentials from anywhere in the world and on any device.
- Google's reputation and popularity as a large company with top-of-the-line cutting-edge technologies drive users to trust their data.
- Many websites and services support Google sign-in so when a user wants to create an account in a new service or web application, the service provides the option to register using the user's Google account which removes the burden of thinking of a new password to sign in with.
- Data breach detection function that was added recently, Google claims that it monitors "the dark web" to detect any users' passwords that were leaked during a data breach and sends a notification to the users to change their passwords.
- Features like a password generator that suggests a random strong password to users when they try to write a new password.
- The autofill feature that auto-fills the input of the credentials once the user clicks inside the text input field.

Despite Google password manager's advantages and features, it has some serious security and privacy compromises and disadvantages:

- The large number of Chrome users all across the world has made it a target of cyber-attacks and data breaches. In the latest data breach that happened in 2018, Google confirmed that the bug impacted approximately 52.5 million users in connection with a Google+ API; [6]
- The poor security method used in storing the accounts file involves putting the encrypted key in a file named Local State. The key used in encryption is encrypted using Data Protection Application Programming Interface (DPAPI) which is a cryptographic application programming interface built-in Windows 2000 and later versions. [7] The DPAPI is used in local machine scope so anyone that has physical or remote access to the user's machine can decrypt the file and get the passwords in a matter of seconds. This can be done using multiple methods one that is common and used by most attackers is sending a phishing email to the user. The message could contain the script that fetches for the "Local State" file, extracts the encryption key, decrypts it using the machine's DPAPI then uses the generated key to decrypt the user's credentials stored in the "Login Data" file.
- Google is known for extensive data collection on its users and one of the latest incidents is the settlement of a lawsuit that accused the company of tracking users' data even while they were using "incognito mode", which is supposed to keep users' data safe. The settlement was a result of a 2020 lawsuit that accused Google of collecting users' private

data. The lawsuit claimed “Indeed, even when Google users launch a web browser with “private browsing mode” activated (as Google recommends to users wishing to browse the web privately), Google nevertheless tracks the users’ browsing data and other identifying information.”; [8]

- Vendor lock-in inside Google’s ecosystem. Google Password Manager works only with Google apps and services and platforms like “Android” which makes it very daunting for a user to switch to another browser or platform and get their credentials synchronized;

- Google Password Manager stores only usernames and passwords and cannot store other types of data which is a lacking feature especially when private data is not always a password or a username, it can be any important personal or financial information like a bank account number or a credit card details or even a crypto wallet passphrase;

- While the simplicity of Google Password Manager is a feature, it is also a drawback. The lack of a user interface and integration into the browser can sometimes be frustrating, especially when a user needs to log in to a desktop application. They have to open Chrome, go to settings, search for Google Password Manager, find the desired account credentials, and copy them—all this just to get their credentials.

- Google fully supports the occupation of Palestine and the genocide that the Zionists are carrying out against the Palestinian people.

1.4.3. Third-party password managers

This category of password managers includes Cloud password managers and desktop password managers. Nowadays most modern password managers have desktop and mobile applications and even web interfaces and browser extensions, so they cover a very large area of platforms. Some of the advantages of using a cloud password manager are:

- Unlike Google Password Manager, cloud password managers are standalone applications with a full user interface and a bunch of functionalities and customizability;
- Cloud password managers store credentials on their servers with some of them supporting the local storage and access of data.
- They use strong encryption and hashing algorithms to secure users’ credentials in their servers and implement various security measures like 2-factor Authentication and biometrics to ensure that only the user can access their data.

Cloud Password Managers are safer than Google’s and have more features, but on the other hand, they have a few disadvantages, some of them are:

- Everything connected to the internet is vulnerable to attacks and is not 100% safe and there were incidents where Password Managers got breached, one is where a known Password Manager named ‘LastPass’ suffered a sequence of attacks. On March 2023 the CEO acknowledged that “the threat actor exploited a vulnerability in third-party

software, bypassed existing controls, and eventually accessed non-production development and backup storage environments." [9]

- Most Cloud Password Managers are paid services that require the user to have a subscription plan and the cost can go higher over time with more features or different plans which puts a burden on the user. There are some free tools but the user is limited to a maximum number of credentials that can be stored and they do not provide solid security or features.
- It's essential to keep in mind when utilizing these Cloud Password Managers users entrust their sensitive information to that service provider. Despite reputable password managers employing robust encryption methods to safeguard data, users must understand that these tools have full access to their information.
- Cloud Password Manager creates a lock-in to their services so that when a user wants to move to another application or another service they will find it complicated to migrate and transfer their credentials to the other application.
- Cloud Password Managers are internet-dependent hence the name "Cloud", so the user needs a stable internet connection to use them and there might be an inconvenience in places that do not have full access to the internet, or for example when a user needs quick access to his social security number and have to connect to a network.

1.5. Conclusion

In conclusion, this chapter provides a comprehensive overview of password managers and some of the challenges that are compromising the security of internet users like data breaches and phishing attacks. Through the examination of users' habits regarding their private information security and storage. Then the critical analysis of existing solutions and the challenges they face in modern cybersecurity. The insights from this literature review will inform the development of a secure credential storage solution tailored to address these challenges effectively.

CHAPTER 2

Conceptual Foundations for Cryptool Account Manager

2.1. Introduction

Before the development of Cryptool Account Manager which is our proposed solution aimed at securely storing and managing user credentials. In this chapter, we explore fundamental principles guiding secure credential storage, including encryption algorithms, authentication mechanisms, and considerations for local file storage. By establishing a robust conceptual framework, we lay the groundwork for designing a secure and user-centric solution that addresses the challenges inherent in credential management and tries to solve some of the issues that users are still suffering from regarding their data security.

2.2. Principles of Secure Credential Storage

When developing a system or an application, various principles should be applied to ensure security and minimize the attack surface area by preventing unnecessary access to different levels of the application and making correct decisions about the security of the application being developed.

2.2.1. Least privilege and separation of duties

Least privilege is a security principle that states that users should have the least access or permissions to do their job, like giving the bare minimum of permissions to the user to enable users to access only the resources they need eliminating unnecessary or excessive access to functionalities. By applying the least privilege principle in the process of securing credentials it limits the potential of an attacker to have access to sensitive data or systems.

Separation of duties is a principle used in internal control in organizations, this principle states that no single individual has access to all aspects of a system, duties should be separated so no one controls all the transactions. By applying separation of duties, it ensures the flow of the work preventing the risk of potential fraud or errors maintaining the integrity of data.

2.2.2. Hashing and hashing functions

Hashing is the process of transforming a string of characters into a shorter fixed-length key that represents the original string. That key is called a hash. [10] The hashing process is irreversible so it is impossible to find the original text starting from the hash.

The term “hash” originated from the physical world, where the standard meaning of “hash” in the English language is “chop and mix”, that means that the hashing function chops string into different parts and then mixes those parts to produce a hash that has a fixed number of bits.

Hashing has three main components, a key which is the string or data that will be fed into the hash function. This latter then maps the key into indexes in the hash table.

Hashing is used in various areas and domains for example it is used in search engines to index documents to make the querying process much faster. And for cryptocurrencies like the SHA-256 used to hash the Bitcoin blocks, and for a digital signature to ensure the integrity of data

transferred. One of the important applications of hashing is in password security. Some websites do not store any valuable personal data like addresses or credit card numbers. Therefore, owners of those sites may think that there is no need to protect passwords. However, due to password reuse, an attacker can use passwords retrieved from a breached site to retrieve valuable data from a more secure one. In a study that was done by J. Bonneau and S. Preibusch in 2010 on 150 websites, they found a cleartext password in a welcome email was observed at 16 sites. [11]

Nowadays, even with the increase in cybersecurity measures and the implementation of different hashing and encryption algorithms to secure passwords and prevent attackers from accessing them, so many websites are still taking poor security measures regarding the protection of their users' passwords and private information. In 2021 many large data breaches happened, one of the most notable is when 8.3 million passwords stored in plaintext were leaked when a hacker breached the "DailyQuiz" builder's database. The data was collected then by "The Record" which contains details about 12.8 million users, including plaintext passwords, emails, and IP addresses for 8.3 million accounts. [12]

The large volume of credentials being stolen and compromised in different cloud services and websites raises a serious concern about data storage management and security and how to use hashing and encryption to achieve confidentiality, integrity, and availability in secure storage solutions. So, how hashing can be used to store credentials in databases and prevent attackers from accessing plain text passwords?

Here comes the importance of hashing in storing passwords in the cloud. By using a hashing function that takes the password as input adds a salt to it and outputs a hash which then will be stored in the database.

A salt is a random string added to the password before the hashing process, using the salt adds the randomness element to the password making it more resistant to guessing and strengthening it like adding a secret ingredient to a recipe hence the name 'salt'. Adding a salt to a password makes it harder for an attacker to guess and brute-force the password.

We can summarize the process of password hashing and salting in steps as follows:

- The user enters their password as a plaintext in the input field.
- The password then will have a salt concatenated to it to add strength.
- The server stores the salt related to that specific password before the hashing algorithm transforms the plaintext into a hash.
- Then the plaintext (password + salt) goes as input in the hashing function which will produce a fixed-length hash.
- The hashed (password + salt) will then be stored in the server's database.
- The next time when the user tries to log in the server hashes the password and a simple string comparison can determine whether or not the provided password is correct.

2.2.3. Secure Hashing Algorithm 256-bit overview

Secure Hashing Algorithm 256 (SHA-256) is a hashing algorithm from the family of Secure Hashing Algorithms. It is the successor of the SHA-1 family after its deprecation by the National Institute of Standards and Technology (NIST) in 2011 [13]. SHA-256 stands out as an important cryptographic algorithm that is widely employed in cybersecurity applications. The SHA-256 produces a 256-bit long hash hence the name, making it resistant to brute-force attacks which play a crucial role in maintaining data integrity that's why it is the main hashing function used in Bitcoin cryptocurrency blocks to hash transactions. One of the properties that make SHA-256 so resilient is that if you change just one letter in the input, the output changes drastically. We are not going to dive deep into the details of the SHA-256 algorithm as it is not the project's main subject.

2.3. Overview of encryption algorithms

The world has seen an increase in cloud users and services. This year statistics show that the end-user spending on public cloud services has reached a market of $678 Billion. [14] That means a vast number of public transactions that are carried out over public wired or wireless networks must have comprehensive communication that ensures the security, authentication, and integrity of the data. To transfer data securely, it must be encrypted and secured so that it cannot be tampered with by unauthorized parties during a transaction. Here comes the important role that encryption algorithms play in encrypting data preventing malicious attackers from getting access to it.

Encryption algorithms are mathematical functions that transform plaint text into an unreadable string called cipher text that is difficult to decrypt using a key. Unlike hashing functions the output of which cannot give you the original text back, encryption algorithms use an encryption key which is used to encrypt and decrypt the cipher text so it can be read by authorized people or organizations. There are two main types of encryption algorithms based on their key a symmetric and asymmetric algorithm.

2.3.1. Symmetric and asymmetric encryption algorithms

Symmetric encryption algorithms use a single key to encrypt and decrypt data, using the same key ensures the speed of the encryption process and security of the data.

On the other hand, asymmetric algorithms use two different keys to encrypt and decrypt data, a private and public key, the private key is used to decrypt and the public key is used to encrypt. Asymmetric encryption algorithms are more complicated and slow compared to symmetric algorithms, are not efficient to use. [15] In practice, they are used together because symmetric encryption is of great advantages in terms of speed, and computation time; however, public key encryption has better key management than private key encryption. [16]

Since we need to encrypt data securely on a local machine, the optimal option is to use a symmetric algorithm like AES which is a fast and safe algorithm that is resilient to password attack.

2.3.2. The Advanced Encryption Standard algorithm overview

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm which means that it uses the same key for encryption and decryption. It was published by (NIST) in 2001. [17]

Encryption with AES is based on a secret key with 128, 192, or 256 bits. AES operates by employing a substitution-permutation network (SPN) structure, which involves multiple rounds of substitution and permutation operations applied to the input data. AES processes data in fixed-size blocks, typically 128 bits in length, and employs a key expansion algorithm to generate round keys from the initial encryption key. Each round of AES encryption consists of four main operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey. [18]

2.4. Considerations for Local File Storage

The process of storing credentials or private information locally has been a topic for discussion for many years, especially in terms of security and access, and even with cloud storage solutions resolving many issues. It still has disadvantages like internet dependency, costs, and data privacy because some private data must be kept secret from the outside world. For this reason, some considerations must be made for local file storage and access on different operating systems.

2.4.1. File format

There are so many file types and formats on different platforms and operating systems for storing data. The most popular local storage solutions are databases, which are a collection of data organized in tables. Databases enable the access and easy manipulation of different data types like executing queries on tables, having relationships, and securing data. However, a database is more suitable for large data or one that requires extensive processing and managing, but storing uncomplex small-size data it adds an unnecessary performance overhead and implementation complications.

The best solution we opted for is using a text-based specific file format which will store the encrypted data as a continuous string making it fast, easy to manage, and transfer to any device or cloud service.

2.4.2. Security risks and backup

Storing data locally has advantages especially when you need to store sensitive data that has to be available all the time. It also has disadvantages like the possibility of data loss and damage to the physical storage unit or drives. The best way to solve local storage issues is by

providing an option for cloud upload of the encrypted file. It ensures the privacy of the data while keeping it safe from the possibility of data loss and enabling the user to choose preferred cloud options while taking advantage of the benefits of using local storage for security and accessibility giving the user freedom of choice.

2.5. Authentication Mechanisms

Authentication is a key element when it comes to accessing private data. There are various authentication methods used in different applications and systems to ensure that only the owner of the data can access it.

Using a password to access data is a traditional method to prevent unauthorized entry. However, relying on passwords alone has become insecure, especially if the password is stolen. Therefore, an additional layer of protection is necessary to verify that the user attempting to access the data is indeed who they claim to be.

2.5.1. Password-based authentication

Password-based authentication is widely used and familiar to users, making it easy to implement and use. Different password rules like complexity and length make the password stronger and harder to guess, but as the hardware gets faster and the introduction of compute units that can perform trillions of calculations per second, password alone has become vulnerable to brute-force attack, so other methods must be added on top to provide more security.

2.5.2. One Time Password

One Time Password (OTP) is type of a 2-factor Authentication (2FA) which is a security method used to add a layer of security by demanding another type of verification from the user. OTP is a login verification code that is only used for one login. OTP code is valid only for a short period, usually 30 or 60 seconds, after which it expires and becomes unusable. This time-based aspect adds an extra layer of security, as attackers have a limited window to intercept the authentication code. This drastically reduces the risk of unauthorized access, even if the password is compromised.

OTP systems generate a unique password for each authentication attempt, ensuring that the password cannot be reused to gain unauthorized access. Once the OTP is used, it becomes invalid, mitigating the risk of replay attacks. OTPs are commonly generated using cryptographic algorithms or hardware tokens, providing an additional layer of security compared to static passwords. [19]

2.6. Conclusion

In this chapter, we discussed different conceptual foundations that were crucial in the process of developing a Cryptool Account Manager taking into consideration the principles of

secure credential storage. In addition, we mentioned a couple of security principles, followed by an overview of hashing and its importance in credential security. After that, we mentioned encryption algorithms and their types. Then, we tackled considerations regarding local file storage. In the end, we closed the chapter with the authentication methods that will be implemented to ensure the security of our application.

CHAPTER 3

The methodological framework for creating Cryptool Account Manager

3.1. Introduction

Developing robust and secure systems or applications requires good planning and designing is recommended to ensure high compatibility and compliance to the global standards. This chapter delves into various facets of the design process, starting with the overall architecture and its high-level components and interactions, passing to the different technologies and development tools chosen and design specifications for Cryptool Account Manager.

3.2. Design phase tools and technologies

During the design phase of the Cryptool Account Manager, a careful selection of tools and technologies has been made considering important aspects of the application design process.

3.2.1. Design and concept

Unified Modeling Language (UML) is a general-purpose visual modeling language intended to provide a standard way to visualize the design of a system. Created by Rational Software in 1997 and later adopted by the Object Management Group (OMG) [20] as the standard for visualizing the design of our system.
The tool we opted to use for designing and modeling the different system flowcharts and diagrams is LucidChart, a web diagramming application produced by Lucid Software Inc. in 2008. LucidChart is a solid tool that facilitates the process of drawing, revising, and sharing charts and diagrams.

3.2.2. Prototyping

Figma, a free web application for interface design was chosen as the tool for prototyping and designing the various user interface elements. The extensive support of plugins and tools for design, prototyping, and animation made it the best choice for the prototyping phase and it was used in the design process to edit diagrams.

3.2.3. Project resources management

Miro, a digital collaboration platform is designed to facilitate remote communication and project management. It was used for managing different resources from websites, documents and media files. It supports various resources such as links, media, and documents. This made the process of organizing project resources easy and efficient.

3.3. Design of Cryptool Account Manager

3.3.1. Flowchart of Cryptool custom algorithm

The flowchart of Cryptool Account Manager illustrates the overall algorithm of the application from a high-level standpoint. It outlines the sequential processes and security stages that constitute the core functionality of the application. By visualizing the algorithm, we can better understand the logical flow and interactions within the system.

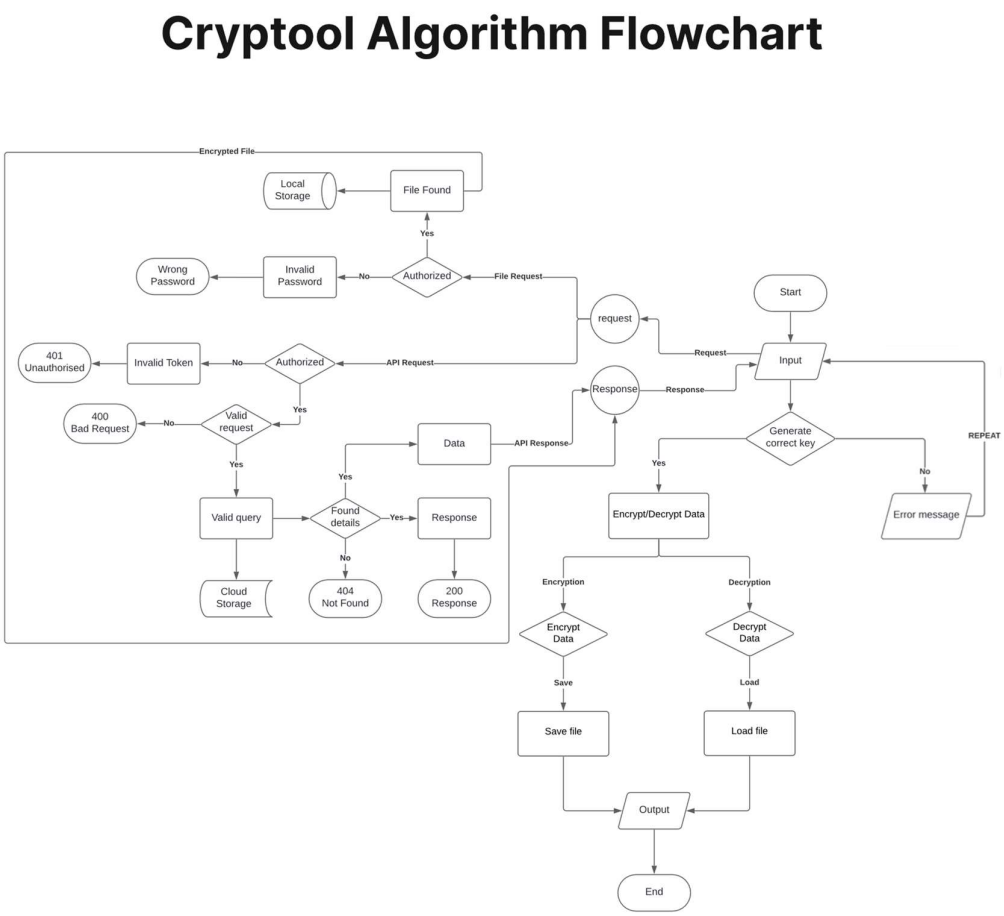

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**1:** Cryptool Algorithm Flowchart.

3.3.2. Use-case diagram

The Use-case diagram of Cryptool Account Manager defines all the functionalities of the application and all the interactions between different actors to have a better understanding of the mechanisms and methods that the application could provide to the user.

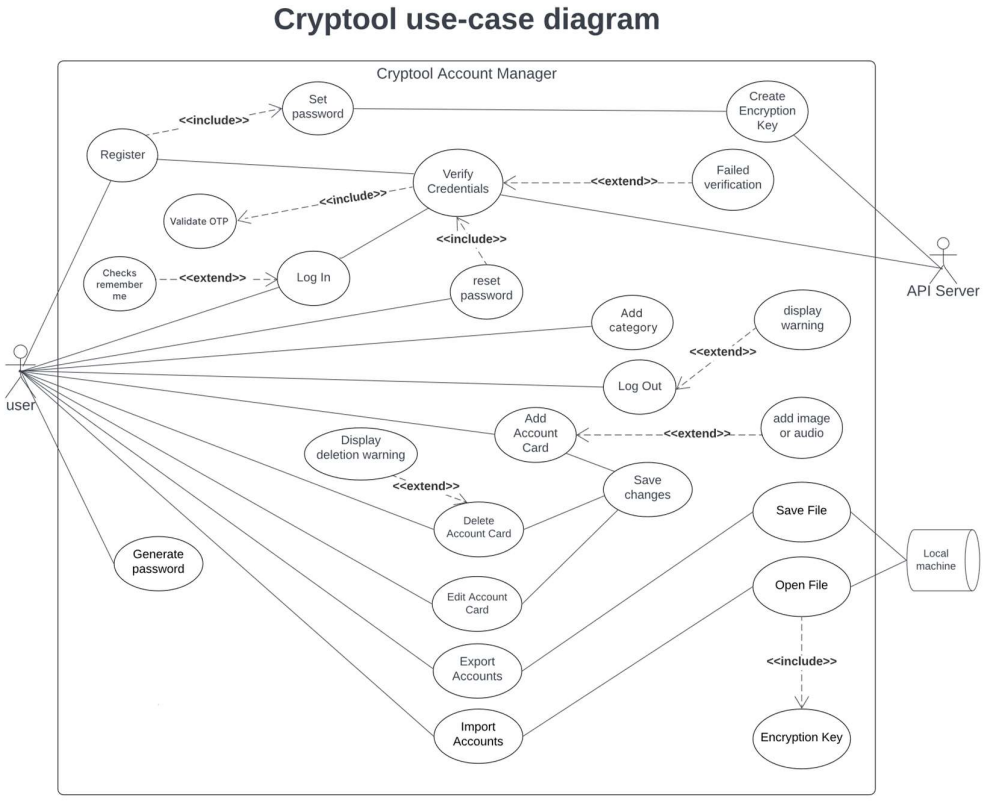

Fig. *Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**2:** UML use-case diagram of Cryptool Account Manager.*

The primary actor in this diagram is the user and the secondary actor is the server, we start by listing the main use-cases as follows:

Register:

Use Case Name	Register
Actors	User, Server
Description	Allows new users to register in Cryptool
Preconditions	The user must enter a valid email and password
Postconditions	The user is successfully registered
Main Flow	1. The user enters their email and password.

	2. The server verifies the user's information. 3. The server sends OTP to the user. 4. The user enters the OTP 5. The server verifies the OTP. 6. the application generates an encryption key from the password. 7. The user account is created.
Alterative Flows	1. The password is invalid. 2. The app prompts the user to enter the correct password. 3. The email is not valid. 4. The app prompts the user to enter a valid email. 5. The OTP expires. 6. The app sends an error message "expired OTP".
Exceptions	- Invalid user email and/or password.
Includes	Set password
Extends	

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.1:** Use-case description (Register)

Login:

Use Case Name	Login
Actors	User, Server
Description	Allows new login to their account
Preconditions	The user must enter their email and password
Postconditions	The user is successfully logged in
Main Flow	1. The user enters their mail and password. 2. The app verifies the user's information. 3. The app and server generate an OTP. 4. The server sends the OTP to the user's email. 5. The user enters the OTP and confirms. 6. The app verifies the OTP. 7. The user is logged in.
Alterative Flows	1. User credentials are not valid or the OTP expires 2. The app prompts an error message.
Exceptions	- Invalid user email and/or password. - Expired OTP code
Includes	
Extends	The user checks the remember me option.

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.2:** Use-case description (Login).

Reset password:

Use Case Name	Reset password
Actors	User, Server
Description	Allows the user to change their password.
Preconditions	The user must be logged in to their account
Postconditions	The password resets successfully
Main Flow	1. The user requests an OTP. 2. The app and server generate the OTP. 3. The server sends the OTP to the user's email. 4. The user enters the OTP and confirms. 5. The app verifies the OTP. 6. The password is reset.
Alterative Flows	
Exceptions	The OTP expires.
Includes	Verify credentials
Extends	

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.3:** Use-case description (reset password)

Add an account card:

Use Case Name	Add account card
Actors	User
Description	Allows users to add a new account card.
Preconditions	
Postconditions	The new account card is added
Main Flow	1. The user adds card information 2. The app saves changes
Alterative Flows	1. The user does not enter the card title 2. The app prompts the user to enter the card title - the app does not save the card
Exceptions	The card title is empty
Extends	Add image or audio files

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.4:** Use-case description (add an account card)

Edit an account card:

Use Case Name	Edit account card
Actors	User
Description	Allows users to edit an account card.
Preconditions	One card is created at least
Postconditions	The card is edited
Main Flow	1. The user edits the card's information 2. The app saves changes
Alterative Flows	1. The user deletes the card title 2. The app prompts the user to enter the card title and does not save changes.
Exceptions	The card title is empty
Includes	
Extends	

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**5:** Use-case description (edit an account card).

Delete an account card:

Use Case Name	Delete account card
Actors	User
Description	Allows users to delete an account card.
Preconditions	One card is created at least
Postconditions	The card is deleted
Main Flow	1. The user deletes card information 2. The app sends the card to the trash folder 3. The user deletes the card permanently – the app removes the card from the system.
Alterative Flows	1. The user selects the cancel option.
Exceptions	
Includes	
Extends	Display deletion warning

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**6:** Use-case description (delete an account card).

Add category:

Use Case Name	Add category
Actors	User
Description	Allows users to add a category.
Preconditions	
Postconditions	The folder is created
Main Flow	1. The user creates a category 2. The user names the category 3. The app adds the category
Alternative Flows	1. The folder name is empty 2. The app does not add the folder.
Exceptions	
Includes	
Extends	

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**7:** Use-case description (add folder).

Export accounts:

Use Case Name	Export accounts
Actors	User
Description	Allows users to export their accounts to an external file.
Preconditions	One card is created at least
Postconditions	The file is saved on the local machine
Main Flow	1. The exports account 2. The app encrypts the account's data using the encryption key 3. The app saves the file on a local machine
Alterative Flows	1. There are no cards in the user account 2. The app prompts the user to add at least one card.
Exceptions	
Includes	
Extends	

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**8:** Use-case description (export accounts).

Import accounts:

Use Case Name	Import accounts
Actors	User
Description	Allows users to import their account data from a local machine.
Preconditions	Accounts file exists
Postconditions	The account data is imported
Main Flow	1. The user imports the account file. 2. The app opens files from the local machine 3. The app adds accounts to the accounts list.
Alternative Flows	1. The user opens a corrupted file. 2. The app prompts an error message.
Exceptions	
Includes	Encryption key
Extends	

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**9:** Use-case description (import accounts).

Generate password:

Use Case Name	Generate password
Actors	User
Description	Allows users to generate a password.
Preconditions	
Postconditions	The password is generated
Main Flow	1. The user clicks the “generate password” button 2. The app generates a password
Alterative Flows	
Exceptions	
Includes	
Extends	

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**10:** Use-case description (generate a password).

Logout:

Use Case Name	Logout
Actors	User
Description	Allows users to log out from Cryptool
Preconditions	The user must be logged in
Postconditions	The user is logged out from Cryptool
Main Flow	1. The user logs out from the app 2. The app returns to the login window
lternative Flows	1. The user selects the option to cancel. 2. The app does not logout.
Exceptions	
Includes	
Extends	Display warning (need network connection for next login)

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.11:** Use-case description (logout).

3.3.3. Sequence diagram

The sequence diagram shows a high-level overview of the interaction that happens between the user and the system sequentially illustrating the flow of messages. The Cryptool Account Manager sequence diagram involves two main sections; the first section represents the authentication process involving the external server, and the second represents the interaction between the user and the application locally. The main components are: The user, the application which includes the user interface, and the server which will handle authentication for the register and login operation.

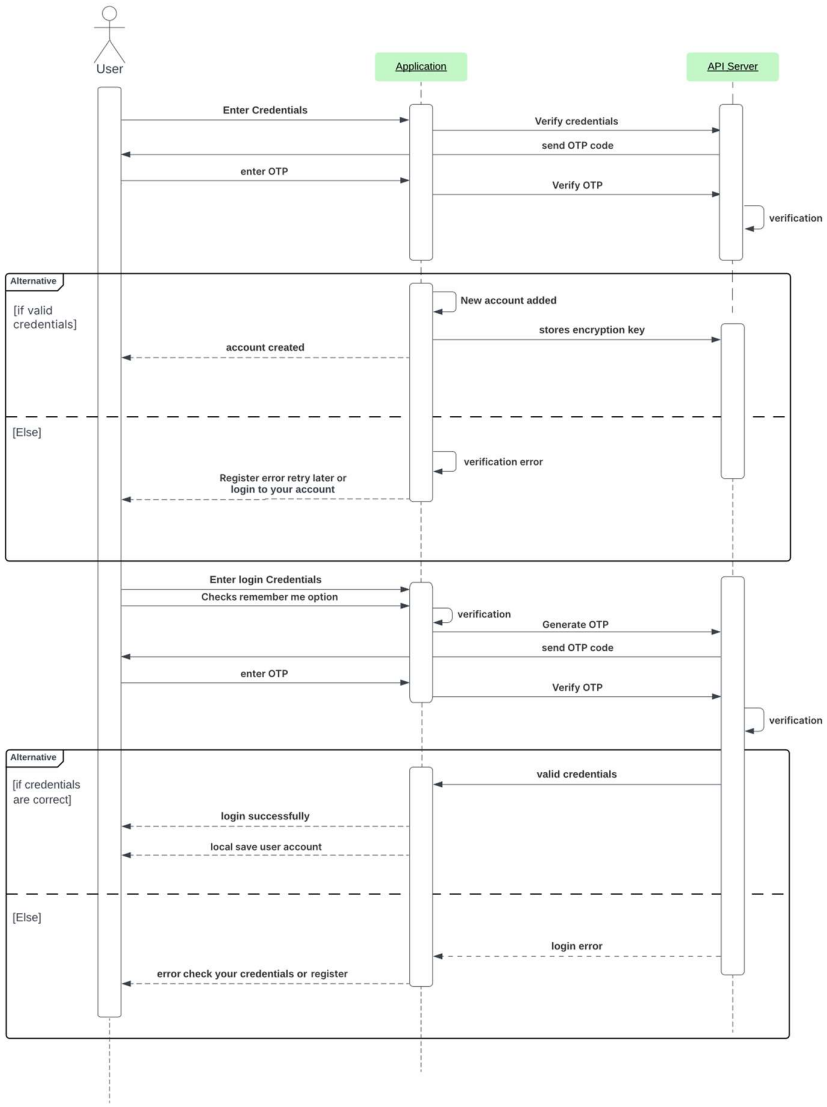

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.3:** UML Sequence diagram of Cryptool Account Manager (Authentication).

The sequence of events:

Registration

This is the sequential order of the events that will take place during the registeration process:

1. The user enters credentials (email, password)
2. The application verifies the credentials.
3. The server generates and sends an OTP to the user via email.
4. The user enters the OTP.
5. The server verifies the OTP.
6. If the credentials and OTP are valid, the app creates the account and generates and stores the encryption key.
7. If the credentials are not valid, the app sends an error message.

Login

For the log in process, there are the same main events with some changes as follows:

1. The user enters the credentials (email, password)
2. The user selects the “Remember me” option.
3. The application verifies the credentials with the server.
4. The application generates an OTP with the server.
5. The server sends the OTP to the user via email.
6. The user enters the OTP in the app.
7. The app and server verify the OTP.
8. If the credentials and OTP are valid, the app allows access to the account.
9. The application saves the user account locally (offline).
10. If the credentials or/and OTP are not valid, the app sends an error message.

The second section of the sequence diagram will represent the interaction between these three main components: the user, the application and the local machine.

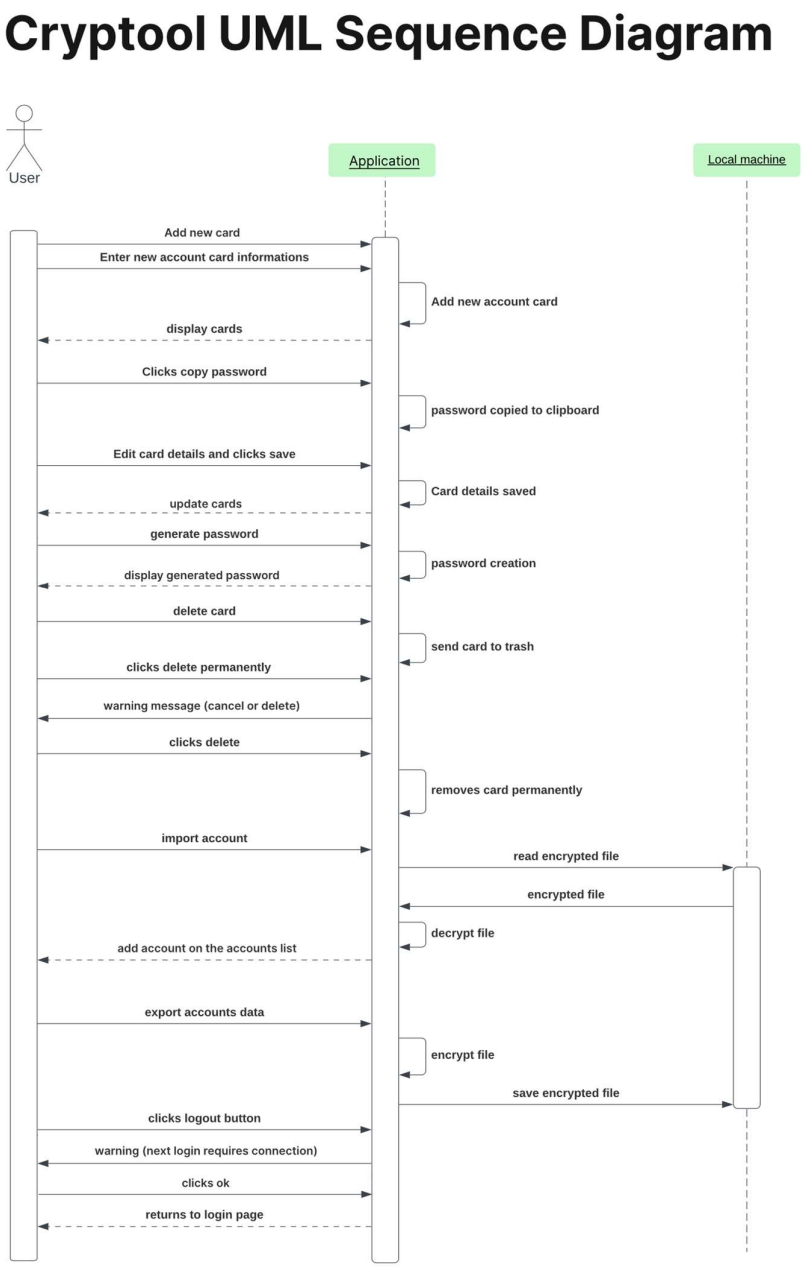

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.4:** UML Sequence diagram for Cryptool Account Manager (App).

The sequence of events:

1. The user adds a new card and enters card information.
2. The app adds the new card and displays it.
3. The user copies a password.
4. The password is copied to the clipboard.

5. The user edits card details and clicks save.
6. The app saves changes and updates the card.
7. The user generates a password.
8. The app displays the newly generated password.
9. The user deletes a card.
10. The app sends the card to the trash folder.
11. The user clicks on delete permanently.
12. The app prompts a warning message (cancel or delete).
13. The user clicks on delete.
14. The app deletes the card permanently.
15. The user imports an account.
16. The app reads the account file from the machine.
17. The app adds accounts to the accounts list.
18. The user exports an account.
19. The app encrypts the account and saves a file on the local machine.
20. The user clicks on the Logout button.
21. The app prompts a warning message (network connection is required for the next login).
22. The user clicks OK.
23. The app returns to the login page.

3.3.4. Class diagram

The class diagram of the Cryptool Account Manager shows the main classes of the application with their relationships, and their methods and attributes. It helps understand how data is managed between its components.

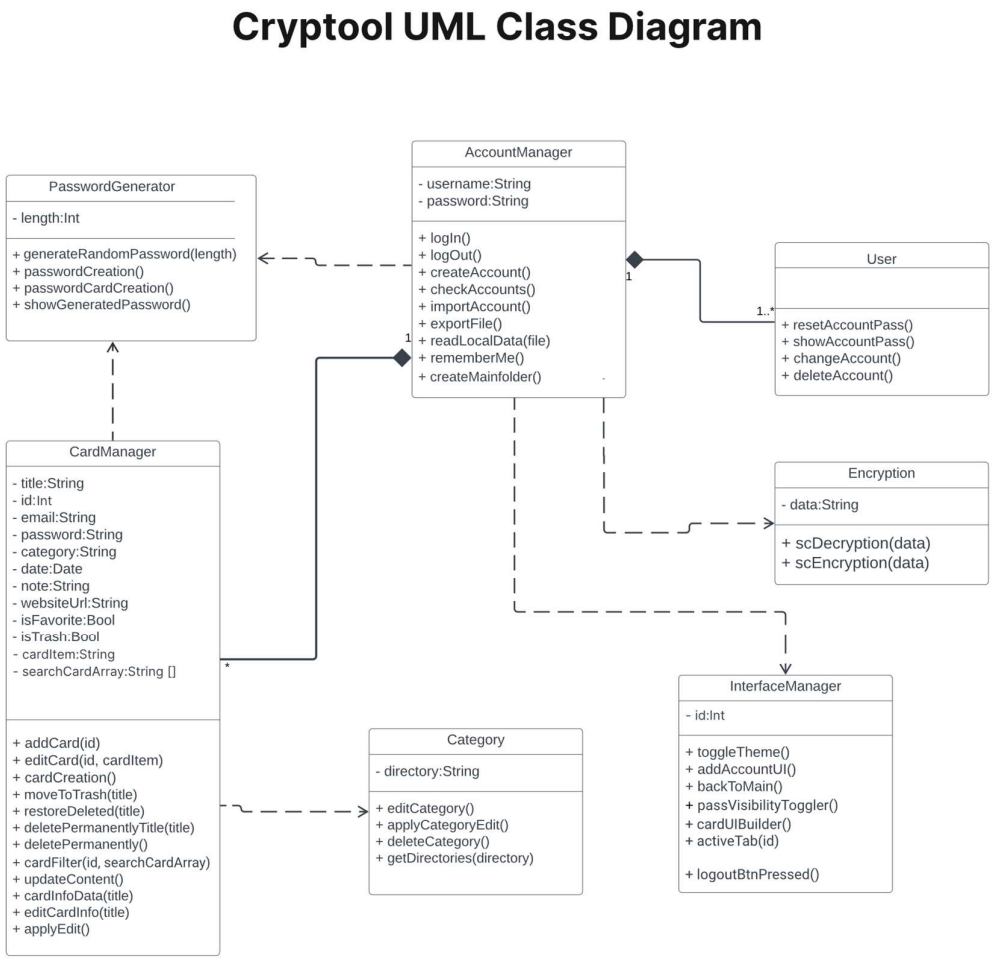

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**5:** UML Class diagram for Cryptool Account Manager.

The following description of the class diagram explains every class with its methods and attributes:

3.3.4.1. AccountManager

This is the main class of Cryptool account manager, it handles register and login operations with authentication process and import and export.

Attributes

- Username: private string type representing the user name.
- Password: private string type representing the password of the user.

Methods

- LogIn(): this method handles the login process.
- LogOut(): handles the logout process.
- createAccount(): handles the creation of a new user account.
- checkAccount(): checks if an account exists.
- importAccount(): handles the external file import.
- exportFile(): exports the account data in a file.
- readLocalData(file): reads the local file.
- rememberMe(): saves the user's account for offline access.
- createMainFolder(): this method is responsible for creating the main folder for Cryptool in the system.

3.3.4.2. User

This class handles the user account and its operations.

Methods

- resetAccountpass(): this handles the reset of the account password.
- showAccountPass(): this handles the password visibility.
- changeAccount(): this is responsible for switching different user accounts.
- deleteAccount(): this handles the account deletion.

3.3.4.3. Encryption

This class handles the data encryption process

Attributes

- data: a string type that stores the user data.

Methods

- scDecryption(data): this method is responsible for decrypting the user's data.
- scEncryption(data): this method handles the encryption of the user's data.

3.3.4.4. InterfaceManager

This class handles all operations involving user interface and interaction between different UI components.

Attributs

- id: an integer representing the id of the tabs in the sidebar component.

Methods

- toggleTheme(): this method handles the overall app theme and colors.
- addAccountUI(): this handles the UI for the user account.
- backToMain(): this method handles navigation.
- passVisibilityToggler(): this method handles the UI components for password visibility functionality.
- cardUIBuilder(): this method is responsible for building the account cards UI.
- activeTab(id): this handles the currently opened tab in the sidebar.
- logoutBtnPressed(): this method is an event listener for the logout button.

3.3.4.5. Category

This class is responsible for the category functionality in the app.

Attributes

- directory: a string that represents the system directory to open files.

Methods

- editCatgory(): this handles the editing of category information.
- applyCategoryEdit(): this method handles saving categories after editing.
- deleteCategory(): this handles the deletion of a category.
- getDirectories(directory): this gets the directory from the system file.

3.3.4.6. CardManager

This class is responsible for all processes related to account cards.

Attributes

- title: a string type representing the card title.
- id: an integer type representing the card id.
- email: a string type representing the email element in the card.
- password: a string representing the password element in the card.
- category: a string type representing the category of the card.
- date: a date type that shows the card creation date.
- note: a string type representing a note element in the card.
- websiteUrl: a string denoting the URL link to the user's registered website.
- isFavorite: a boolean type denoting whether a card is a favorite or not.

- isTrash: a Boolean type denoting whether a card is deleted (sent to trash) or not.
- cardItem: a string representing card info items.
- searchCardArray: an array of card titles.

Methods

- addCard(id): handles adding a new card with an ID number.
- editCard(id, cardItem): handles editing a specific card.
- cardCreation(): handles the creation of a newly added card.
- moveToTrash(title): handles the transfer of a card to trash.
- restoreDeleted(title): handles the restoration of a deleted card.
- deletePermanentlyTitle(title): handles the permanent deletion of a card using the title.
- deletePermanently(): handles the permanent deletion of cards.
- cardFilter(id, searchCardArray): handles the filter of cards to display.
- updateContent(): handles the update of data dynamically when changes happen.
- cardInfoData(title): handles the display of specific card details.
- editCardInfo(title): handles the editing of specific card details.
- applyEdit(): saves the changes after editing.

3.3.4.7. PasswordGenerator

This class handles the password-generation process.

Attributes

- length: an integer type representing the number of characters in the generated password.

Methods

- generateRandomPassword(length): takes the length as an input and returns the random string of characters.
- passwordCreation(): the main method that invokes the generation method.
- passwordCardCreation(): handles the generation of the password inside the card element.
- showGeneratedPassword(): handles the visibility of the generated password.

3.4. Conclusion

In conclusion, this chapter has provided a comprehensive conception of the Cryptool Account Manager architecture and the design phase, starting with a brief summary of the specific tools and technologies that were used during the design phase, then giving an overview of the overall system algorithm and its main components. Then, the architectural diagrams along with their respective explanations are presented. This chapter lays a solid ground for the development phase and guides a seamless transition from the conception to the implementation of the Cryptool Account Manager.

CHAPTER 4

Design, implementation and testing

4.1. Introduction

In this chapter, we delve into the process of designing and implementing the Cryptool Account Manager, starting with the user interface design and features as it is the layer that interacts with users and handles all their actions showcasing the meticulous attention to detail to ensure the usability. detailed explanation will be provided of the tools and technologies employed in the development process. Then, a brief overview will be given on the encryption and decryption mechanisms that Cryptool uses. In addition, there will be a description on the integration of the tool with the local file system and the password generation process. Finally, the integration of authentication mechanisms will be dealt with. These parts will cover the main and crucial aspects of the Cryptool implementation process.

4.2. User interface design and features

4.2.1. Choice of color

In the case of our software Cryptool, we deliberately chose green as the main color for several reasons. Firstly, green symbolizes security and trust, aligning perfectly with Cryptool's core purpose of safeguarding sensitive data. It creates an atmosphere of reliability and assurance, assuring users that their confidential information is protected within the digital safe.
Additionally, green also represents growth and prosperity. By associating Cryptool with this color, we aim to convey the idea of personal and digital growth, emphasizing the software's capability to help users manage their credentials and important data securely, thereby fostering their overall digital well-being.
Furthermore, green is universally recognized as a color of balance and harmony [21]. In the context of Cryptool, this choice reflects our commitment to providing users with a seamless and harmonious experience while navigating the software's interface. It promotes a sense of ease and clarity, ensuring that users can interact with the application effortlessly, enhancing their overall satisfaction and confidence in using Cryptool.
Also, green is known for its soothing effect on the eyes and its positive impact on readability. By incorporating green into Cryptool's design, we prioritize user comfort and ease of use. The color's gentle tone reduces eye strain during prolonged use, enhancing overall readability and ensuring that users can interact with Cryptool comfortably for extended periods.

4.2.2. Layout

Cryptool is divided into two main sections: the left section and the right section, each containing features that contribute to a good UI/UX experience for the user:

- **Intuitive Organization:** The left section's layout follows a logical hierarchy, with tools like All Items, Favorite Items, Trash, support, and settings, placed for easy access. This intuitive organization helps users quickly find and manage their data.

- **Efficient Navigation:** Placing the search bar at the top of the right section allows users to quickly locate specific items within their vault, enhancing navigation efficiency. The "Adding New Items" button and export menu are also conveniently located for seamless interaction.
- **Clear CTA Placement:** Placing essential actions like "Log Out" prominently ensures that users can easily exit the application when needed. This clear Call-to-Action (CTA) placement enhances usability and prevents user frustration when ending a session.
- **Consistent Interaction**: The right section dynamically updates based on the user actions or selections, providing immediate feedback and ensuring a consistent interaction experience. Whether adding new items or accessing settings, users can expect the interface to respond predictably.
- **Streamlined Workflow:** The overall layout and feature placement contribute to a streamlined workflow, minimizing cognitive load and allowing users to focus on their primary tasks of managing and securing their data effectively.

4.3. Selection of the development tools and technologies

4.3.1. Tools

4.3.1.1. VS Code

Visual Studio Code stands out as a versatile and feature-rich source code editor that caters to the diverse needs of developers, offering a seamless development experience across different platforms and programming languages. Despite some minor drawbacks, its extensive feature set, extensions, and active community support make it a top choice for developers worldwide [22].

Visual Studio Code: also known as VS code, is a free and open-source code editor developed by Microsoft, It is designed to be customizable, efficient, and optimized for different development tasks.

Key features:

- Cross-Platform: VS Code is available for Windows, macOS, and Linux, making it accessible to developers across different platforms, and also ensuring a consistent development experience across different operating systems.
- VS Code supports a wide range of programming languages and offers an extensive ecosystem.
- Intelligent Code Editing (Intellisense): It offers features like syntax highlighting, autocompletion, and code refactoring, enhancing productivity and reducing development time.

- Rich Extension Ecosystem: VS Code has a vast library of extensions that can be easily installed to tailor the editor to specific development needs, including language support, debugging tools, and themes.
- Built-in Git Integration: Git version control is seamlessly integrated into the editor, allowing developers to manage source code repositories directly within the interface.
- Integrated Terminal: VS Code includes a built-in terminal that enables developers to execute commands, run scripts, and interact with the underlying operating system without leaving the editor.
- Debugging Support: It provides robust debugging capabilities with support for various programming languages and frameworks, helping developers identify and fix bugs efficiently.
- Customizable UI: The user interface (UI) of VS Code is highly customizable, allowing the developers to adjust layout, themes, and keyboard shortcuts to suit their preferences and workflows.
- Live Share Collaboration: With the Live Share extension, developers can collaborate in real-time, share code, and debug together, regardless of their physical location.
- Performance: Despite its feature-rich nature, VS Code remains lightweight and responsive, providing a smooth and efficient coding experience even on less powerful hardware.

4.3.1.2. Postman

Postman serves as a comprehensive and versatile platform for Application Programming Interface (API) development, offering a wide range of features to streamline the API development lifecycle [23]. Despite some challenges related to its learning curve and resource consumption, Postman remains a valuable tool for developers seeking to build, test, and document APIs efficiently. Postman was originally launched as a Google Chrome extension, but evolved into a comprehensive API toolchain that offers a range of features to simplify API development workflows [24].

Key features:

- API Testing: Postman allows developers to create and execute API requests quickly and efficiently, supporting various HTTP methods, request parameters, headers, and authentication methods.
- Collections: Developers can organize API requests into collections, making it easy to manage and share related endpoints, tests, and documentation.
- Automated Testing: Postman supports the creation of automated tests for APIs using JavaScript, enabling developers to write and run tests directly within the application to ensure API reliability and consistency.

- Mock Servers: With Postman mock servers, developers can simulate API endpoints and responses, allowing them to test client-side integrations without relying on the actual backend implementation.
- Documentation: Postman offers built-in documentation generation for APIs based on collection descriptions, making it easy to create comprehensive API documentation that can be shared with team members or external stakeholders.
- Monitoring: Postman provides monitoring capabilities that allow developers to track API performance, uptime, and response times, helping identify issues and optimize API performance.
- Collaboration: Postman's collaboration features enable team members to work together on API development projects, facilitating communication, version control, and sharing of collections and environments.
- Environment Variables: Postman supports the use of environment variables, allowing developers to define and manage dynamic values across requests, making it easier to work with different environments such as development, staging, and production.
- Integration: Postman integrates seamlessly with other development tools and services, including version control systems like GitHub, CI/CD pipelines, and API monitoring platforms, enhancing the overall development workflow.
- Security: Postman prioritizes security by offering features such as encrypted data transmission, role-based access control, and audit logs, ensuring that sensitive API data remains protected throughout the development lifecycle.

4.3.1.3. MongoDB

MongoDB is a leading NoSQL database management system known for its flexibility, scalability, and performance in handling large volumes of unstructured and semi-structured data. It diverges from traditional relational databases by using a document-oriented data model, making it well-suited for modern applications that require dynamic and evolving data schemas [25]. MongoDB may present challenges in terms of operational complexity and consistency management, but it remains a popular choice for organizations seeking to harness the power of NoSQL databases for their applications.

Key features:

- Document-Oriented: MongoDB stores data in flexible, JSON-like documents, allowing developers to represent complex relationships and nested data structures easily;
- Scalability: MongoDB offers horizontal scalability through sharing, enabling seamless distribution of data across multiple servers to accommodate growing workloads and ensure high availability and performance.
- High Performance: With its memory-mapped storage engine and native support for indexing, MongoDB delivers fast read and write operations, making it suitable for real-time applications and data-intensive workloads.

- Rich Query Language: MongoDB provides a powerful query language, allowing developers to retrieve and manipulate data with ease.
- Schema Flexibility: Unlike traditional relational databases, MongoDB does not enforce a rigid schema, enabling developers to evolve data structures over time without requiring downtime or complex migrations.
- Security Features: MongoDB offers robust security features such as authentication, access control, encryption at rest, and auditing, helping organizations protect sensitive data and comply with regulatory requirements.
- Integration: MongoDB integrates seamlessly with popular programming languages, frameworks, and tools, offering official drivers and client libraries for Java, Python, Node.js, and more, as well as connectors for BI and analytics platforms.
- Cloud-Native: MongoDB Atlas, the fully managed cloud database service, allows developers to deploy, manage, and scale MongoDB clusters in the cloud with ease, offering automated backups, monitoring, and security controls.

4.3.1.4. GitHub

GitHub is a web-based platform built around Git, a distributed version control system that facilitates collaborative software development. It serves as a central hub for hosting, sharing, and managing code repositories, enabling developers to collaborate on projects, track changes, and coordinate workflows [26].

Key features:

- Code Hosting: GitHub provides a centralized platform for hosting Git repositories, allowing developers to store, access, and manage their codebases securely in the cloud;
- Version Control: With Git at its core, GitHub enables developers to track changes to their code over time, maintain multiple versions, and collaborate with others through features like branching, merging, and pull requests.
- Collaboration Tools: GitHub offers a suite of collaboration tools, including issue tracking, project boards, and wikis, to facilitate communication, coordination, and task management among team members.
- Pull Requests: Developers can propose changes to a repository by submitting pull requests, which allow for peer review, feedback, and discussion before merging changes into the main codebase.
- Community Engagement: GitHub fosters an active and vibrant developer community, enabling users to discover, explore, and contribute to open-source projects, as well as connect with like-minded developers through discussions, forums, and events.
- Security Features: GitHub provides robust security features such as access controls, vulnerability scanning, dependency management, and code scanning, helping developers identify and address security vulnerabilities and protect their code and data.

4.3.2. Technologies

4.3.2.1. Hashing

Hashing is a cryptographic technique used to convert data of arbitrary size into a fixed-size string of characters, known as a hash value or hash code. The hash function takes input data (such as a file, message, or password) and produces a unique hash value that serves as a digital fingerprint for verifying data integrity, authenticity, and identity. Hash functions are designed to be deterministic, meaning the same input always produces the same output, and one-way, meaning it is computationally infeasible to reverse the process and obtain the original input data from the hash value.

4.3.2.2. Encryption

Encryption is a process of converting plaintext (unencrypted data) into ciphertext (encrypted data) using an algorithm and a cryptographic key. The primary purpose of encryption is to ensure confidentiality by rendering data unreadable to unauthorized users. Encrypted data can only be deciphered (decrypted) using the corresponding decryption key, allowing authorized parties to access the original plaintext.

4.3.2.3. Node.js

Node.js is an open-source, cross-platform JavaScript runtime environment built on Chrome's V8 JavaScript engine. It allows developers to run JavaScript code outside a web browser, enabling server-side scripting to build scalable and high-performance network applications. Node.js uses an event-driven, non-blocking I/O model, making it lightweight and efficient for handling concurrent connections and asynchronous operations [27].

Reasons for using Node.js:

- Rich Ecosystem: Node.js has a vibrant ecosystem of npm modules, offering a vast array of open-source libraries and frameworks for building servers, RESTful APIs, real-time applications, microservices, and more. This extensive ecosystem accelerates development, facilitates code sharing, and fosters innovation.
- Performance: Node.js leverages Chrome's V8 engine, which compiles code into highly optimized machine code. This results in fast execution speeds and low latency, making Node.js suitable for building high-performance applications that can handle a large number of concurrent connections and real-time interactions.
- Community Support: Node.js benefits from a strong and active community of developers, contributors, and enthusiasts who provide ongoing support, share best practices, contribute to the ecosystem, and collaborate on improving the platform's features, performance, and security.

4.3.2.4. REST API architecture

Representational State Transfer (REST) is an architectural style for designing networked applications, and RESTful APIs adhere to this style. REST APIs are designed to be lightweight, scalable, and flexible, using standard HTTP methods and data formats to perform CRUD operations on resources. REST APIs follow a client-server model, where clients make requests to access or manipulate resources on a server using well-defined endpoints [28].

Reasons for using REST API for the Back-end:

- Simplicity and Uniformity: REST APIs are built on simple, well-known principles of HTTP, making them easy to understand, implement, and use. They provide a uniform interface for interacting with resources through standard HTTP methods, URIs, and status codes, fostering interoperability and reducing complexity.
- Statelessness: REST APIs are stateless, meaning each request from a client to the server contains all the information necessary to process the request. This architectural constraint simplifies server-side logic, improves scalability, and enhances reliability by eliminating the need to manage the session state on the server.
- Scalability and Performance: RESTful architectures are inherently scalable and performant, allowing for horizontal scaling by adding more servers or instances to handle increasing load and concurrent requests. They leverage the caching mechanisms provided by HTTP, reducing server load and latency, and improving overall system performance.
- Flexibility and Interoperability: REST APIs support multiple data formats, providing flexibility in data representation. They also enable seamless integration with various client applications, programming languages, and platforms, promoting ecosystem growth.
- Separation of Concerns: REST APIs promote a clear separation of concerns between clients and servers, with well-defined boundaries and responsibilities. Clients are decoupled from server implementations, allowing them to evolve independently and enabling teams to work in parallel on frontend and backend components without tight coupling.
- Security: REST APIs can leverage various authentication and authorization mechanisms provided by HTTP to secure access to resources and protect against unauthorized actions, where authentication helps verify the identity of a user or service, and authorization determines their access rights.

Authentication in REST API

Authentication in REST API refers to the process of verifying the identity of clients or users who make requests to access protected resources or perform privileged operations on a server. It ensures that only authorized entities are granted access to specific endpoints or functionalities within an application. Authentication mechanisms authenticate the credentials provided by clients, such as usernames and passwords, tokens, or certificates, against a trusted source, such as a database, identity provider, or authentication server.

Authorization in REST API

Authorization in REST API refers to the process of determining whether a client or user who has been authenticated is allowed to access specific resources, perform certain operations, or execute particular actions within an application. It establishes and enforces access control policies and permissions based on the authenticated user's identity, roles, privileges, and other attributes. Authorization mechanisms evaluate the permissions associated with requested resources and enforce security policies to grant or deny access accordingly, ensuring that users only have access to the resources and functionalities they are authorized to use.

4.3.2.5. OTP (One Time Password)

OTP, or One-Time Password, is a unique authentication code that is valid for only a single login session or transaction. It is typically generated and used for authentication purposes.

Reasons for using OTP:

- Security: OTP adds an extra layer of security beyond traditional password-based authentication methods.
- Dynamic: Each OTP is unique and valid only for a short period, reducing the risk of unauthorized access through stolen or intercepted codes.
- Flexibility: OTP can be delivered through various channels such as SMS, email, or mobile apps, providing users with flexibility and convenience.
- User Convenience: OTP solutions offer a balance between security and user experience, providing a relatively seamless authentication process for users.

4.3.2.6. Development Environment

Cryptool was developed on a system running Windows 11, utilizing these hardware components: an AMD Ryzen 5 3600 processor, 16GB of RAM, and an NVIDIA GeForce RTX 3060 graphics card. Cryptool's installer is approximately 80MB in size, while the installed software occupies around 280MB of disk space.

4.4. Development of the custom encryption algorithm

Cryptool's custom encryption algorithm was developed by integrating and coordinating the output of SHA-256 algorithm and AES algorithm and some specific data from the Cryptool API. This intricate process ensures that Cryptool's encrypted files are highly complex and resistant to decryption. Cryptool’s encrypted files are engineered with robust security measures to resist decryption attempts, even if an unauthorized individual gains access to both the password and the encrypted file, decryption remains unachievable. This ensures that the file remains protected and inaccessible to malicious actors, safeguarding sensitive information from unauthorized disclosure or tampering.

This is the maximum extent to which we can elaborate on how the custom algorithm operates to safeguard intellectual property.

4.5. Encryption and decryption mechanisms

Cryptool employs a straightforward encryption and decryption process. It utilizes the AES algorithm to encrypt the user's data, storing it locally on their machine. When the user accesses Cryptool, the file is decrypted, granting access to the confidential content.

The strength of Cryptool doesn't lie in the algorithms it uses, but rather in how effectively it utilizes them.

4.6. Integration with local file system

When Cryptool is installed, it creates a dedicated folder in the user's home directory. This folder serves as the repository for all Cryptool encrypted files, providing a centralized location for saving and accessing them at any time.

Why use local storage for saving files?

Using local storage for saving files serves several important purposes for Cryptool:

- Accessibility in Offline Environments: By storing files locally, Cryptool ensures that users can access their data even without an internet connection. This feature is particularly beneficial for users who may need to access their sensitive information while offline, providing them with uninterrupted access to their encrypted files.
- Ensuring Credibility and User Ownership: Local storage enhances credibility by giving users full control and ownership of their data. Since the files are stored directly on the user's device, they can trust that their sensitive information remains private and secure. This approach fosters a sense of ownership and trust, as users have complete control over their data without relying on external servers or third-party hosting services.
- Avoiding Hosting Costs and Lowering Expenses: Utilizing local storage helps Cryptool avoid the need for hosting services, which can incur high costs and overhead expenses. By eliminating the reliance on external servers, Cryptool can provide its services more affordably to users. This cost-effective approach makes Cryptool more accessible and obtainable for a broader range of users, without compromising on data security or reliability.

4.7. Integration of authentication and authorization mechanisms

4.7.1. Cryptool side

When the user submits their email and password, the API validates the credentials. If correct, Cryptool then generates a unique OTP, consisting of 6 characters, and then sends it to the user's provided email address. This OTP is valid for 1 minute before it expires. By requiring users to authenticate via OTP sent to their registered email, Cryptool adds a layer of security to verify the user's identity and ensure that they are the legitimate owner of the account.

4.7.2. API side

4.7.2.1. Authentication

The authentication API employed by Cryptool requires the user to provide a unique email along with their username and password. Upon validation, if the email is found to be unique, a new user account is created, the API then generates a token to authenticate the user, facilitating seamless access to the platform.

4.7.2.2. Authorization

For authorization, Cryptool employs a role-based system where users are assigned a single role, typically labeled as "user," which grants them specific levels of access and actions within the platform. Any actions beyond the designated permissions are considered unauthorized for the user.

4.8. Conclusion

In this chapter, we covered different aspects of the design and implementation of Cryptool, from the user interface, and the choice of color schemes to the layout. Then, we explained in detail the tools and technologies we used to implement all the functionalities. After that, a quick overview of the custom encryption algorithm used, the mechanisms of encryption and decryption, and the integration with a local file system were made. We concluded with the integration of authentication and authorization mechanisms on Cryptool's side and API's side. This technical chapter is a pivotal stage that helped in bringing our secure password management solution from a concept on paper to a real-world solution

4.9. Testing and evaluation

Testing software products is a crucial process before making them available for use by making sure all use cases of the system are achieved. In addition, it is important to monitor performance to ensure reliability and compatibility to identify, and fix potential issues and bugs before the deployment to prevent serious vulnerabilities and potential attacks.
In this chapter, we are going to present the test of some scenarios and then evaluate the usability of the Cryptool Account Manager. Next, we review various performance testing metrics and present the findings.

4.10. Test cases and scenarios

We are going to test scenarios for the main use cases of the Cryptool Account Manager which are: Register, Login, Add card, Import file, and Export File. The scenarios are the main flow highlighted in green and the alternative flows highlighted in pink.

Register:

Test cases	**Steps**	**Expected results**
Test case:	Create account	
1)	Enter a valid email and password	The server verifies email and password
2)	Enter a valid OTP code.	The app creates an account
Test case:	Invalid email	
1)	Repeat step 1) of Creating an account	
2)	Enter invalid email	The app will display an error message
Test case:	Incorrect password length	
1)	Repeat step 1) of Creating an account	
2)	Enter incorrect password length	The app prompts the user to enter the correct password length
Test case:	Expired OTP code	
1)	Repeat step 2) of Creating an account	
2)	Enter expired OTP code	The app will display an error message ”OTP expired”

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.1:** A test case for the use-case "Register".

Login:

Test cases	Steps	Expected results
Test case:	Login to account	
1)	Enter a valid email and password	The server generates and sends an OTP code
2)	Enter a valid OTP code	The app logs into the account
Test case:	Invalid credentials	
1)	Repeat step 1) Login to account	
2)	Enter an invalid email or password	The app will display an error message
Test case:	Expired OTP code	
1)	Repeat step 2) Login to account	
2)	Enter expired OTP code	The app will display an error message ”OTP expired”

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**2**: A test case for the use-case "Login".

Add card:

Test cases	Steps	Expected results
Test case:	Add card	
1)	Enter card information and select the save option	The app saves card
Test case:	Empty card title	
1)	Repeat step 1) Add card	
2)	Empty card title	The app will prompt to enter the card title – the app does not save the card.

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**3:** A test case for the use-case "Add card".

Import account file:

Test cases	Steps	Expected results
Test case:	Import account file	
1)	Select the account file in the	The opens account file and adds the

	local machine	account to the accounts list
2)	Enter account credentials	The app logs into the account
Test case:	Corrupt file	
1)	Repeat step 1) Import account file	
2)	Select the corrupt file	The app will display an error message
Test case:	Invalid account credentials	
1)	Repeat step 2) Import account file	
2)	Enter invalid credentials	The app will display an error message

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.4:** A test case for the use-case "Import account file".

Export account file:

Test cases	**Steps**	**Expected results**
Test case:	Export account file	
1)	Select the export account file	The app encrypts and saves the file on the local machine
Test case:	No account cards	
1)	Repeat step 1) Export account file	
2)	Select export account with no cards	The app displays an error message "add at least one card"

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.5:** A test case for the use-case "Export account file".

4.11. Performance testing metrics and results

4.11.1. Performance testing overview

Performance is a crucial quality measure of a software product, to ensure the efficiency when using Cryptool. During the manual testing of use cases, we recorded the system performance and resource consumption for the two states Idle and under-load to provide a comprehensive overview of the software performance and usage of system resources.

During the idle state, we noticed minimal resource utilization, with CPU usage averaging around 0.3%. RAM consumption remains relatively stable at 87MB, indicating efficient memory management. Disk activity remains negligible at 0.1MB/s, as an offline tool that will be idle most of the time and it does not impose significant demands on system resources ensuring a seamless user experience.

Under load conditions, Cryptool shows a minimal increase in resource utilization, with CPU usage rising to approximately 10%. Despite this uptick, CPU utilization remains within reasonable bounds. RAM usage also experiences a modest increase, reaching 110MB, Disk activity remains consistent at 0.1MB/s, indicating that the application efficiently manages data access without overwhelming disk I/O, with only a small noticeable glitch due to uploading media files. Overall, Cryptool shows effective balancing resource consumption to maintain responsiveness and stability.

4.11.2. Results comparison

After we get the performance results, it is necessary to compare them to existing tools to have an overview of how the implemented tool is performing against others and highlight areas of improvement to optimize Cryptool functionality and efficiency. We compared the results of CPU, memory, and disk usage against the NordPass desktop application. The test machine specifications are a Celeron B815 processor with 4GB of memory and an HDD. The results are represented by the following charts:

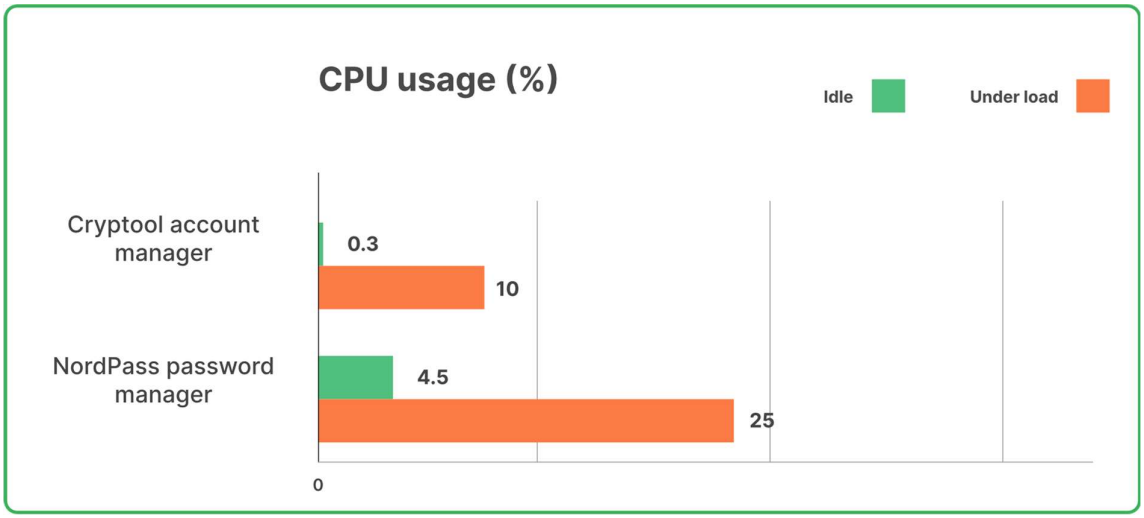

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**1:** Cryptool and NordPass Comparison (CPU usage).

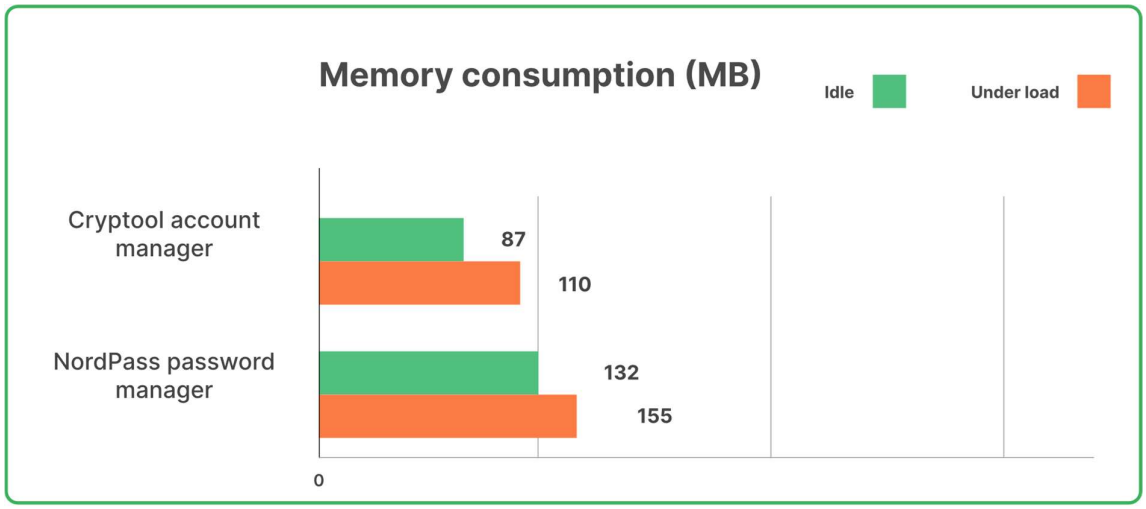

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.2:** Cryptool and NordPass Comparison (Memory consumption).

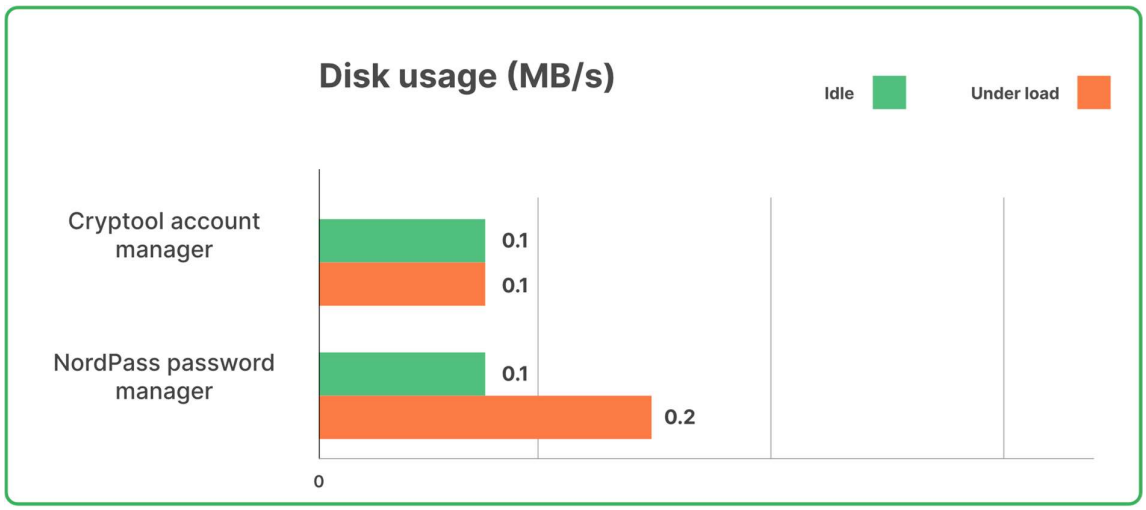

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.3:** Cryptool and NordPass Comparison (Disk usage).

4.12. Conclusion

In this chapter we provided a brief testing process in which we tested the main use cases of Cryptool, followed by a performance testing overview and a comparison between the NordPass password manager and our tool. All the testing was done manually and on-premises due to time constraints and the lack of user feedback, so the results may vary depending on the machine's software and hardware specifications.

CHAPTER 5

Discussion

5.1. Introduction

In this chapter, we discuss various aspects of the implemented tool. We start by presenting an objective comparison between the Cryptool Account Manager and other existing solutions which will serve as a basis for the next section where we summarize the strengths and limitations of the tool. Then, we move to the next section where we provide practical implications of the tool for end-users and conclude the chapter by discussing various ideas and methods for enhancing the implemented tool and outlining directions for future research or improvement.

5.2. Comparison with existing solutions

The evaluation process plays a crucial role after a system or a tool is implemented, and for that reason, we have to conduct a comparative analysis with some existing solutions. For the comparison to be fair, we picked Google Password Manager as it's the most used password manager, and for other third-party tools, we picked NordPass as a representative for them according to TechRadar's best password managers for 2024 [29]. We summarized some of the key criteria for the comparison as shown in the table below:

Features	**Cryptool Account Manager**	**Google Password Manager**	**NordPass Password Manager**
Price	Free	Free	Paid (Subscription)
Security	Strong	Poor	Strong
Two-Factor Authentication Support	Yes	Yes	Yes
Secure Notes	Yes	No	Yes
Cloud synchronization	Versatile	Limited (Google account)	Limited (NordPass account)
Password generation	Customizable	Basic	Basic
File attachments support	Yes	No	Yes

Table Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.1:** Comparison between the Cryptool Account Manager and other solutions.

5.3. Strengths and limitations of Cryptool

After the comparison, we are going to summarize the limitations and strengths of the Cryptool Account Manager to have a comprehensive understanding of its capabilities and areas for improvement.

5.3.1. Limitations

1. Testing limitations as extensive testing requires a large sample of data submitted by users.
2. For the time being, the Cryptool Account Manager is in beta version and only supports desktops.
3. Manual cloud synchronization can be inconvenient for some users.
4. Other methods of authentication have not been implemented yet.
5. Some features have not been implemented yet.

5.3.2. Strengths

1. The Cryptool Account Manager is offline and we are not involved in storing or accessing any user information as data privacy is our number one concern.
2. User data is encrypted using a strong encryption function paired with 2-factor authentication to ensure security.
3. Cryptool is free with a lifetime license and access to all the app features, without the need for recurring subscription fees.
4. Users can import and export encrypted data in a single file and share it or save it wherever they want.
5. The Cryptool Account Manager is network independent and an internet connection is needed only to check the user when he tries to sign in.
6. Cryptool is more than just a password manager, it is a personal digital safe that can secure multiple types of data like: notes, images, audio, and more.
7. Ease of use and the convenience of the user interface and access to data.
8. Features such as password creator, favorite cloud upload, and more will be added in the future with every update.

5.4. Guidelines and disclaimers

Cryptool Account Manager is developed to secure and store credentials and is going to be used by all types of users, so we have to provide a comprehensive understanding of how to utilize the application effectively, while also clarifying what the application does not claim and what users can expect from its functionalities.

5.4.1. User Guidelines

Get started with securing your data, with this brief instruction on how to use Cryptool to Manage your important information, with its convenient interface and intuitive design.

Register:

- The user clicks Add new account button to register.

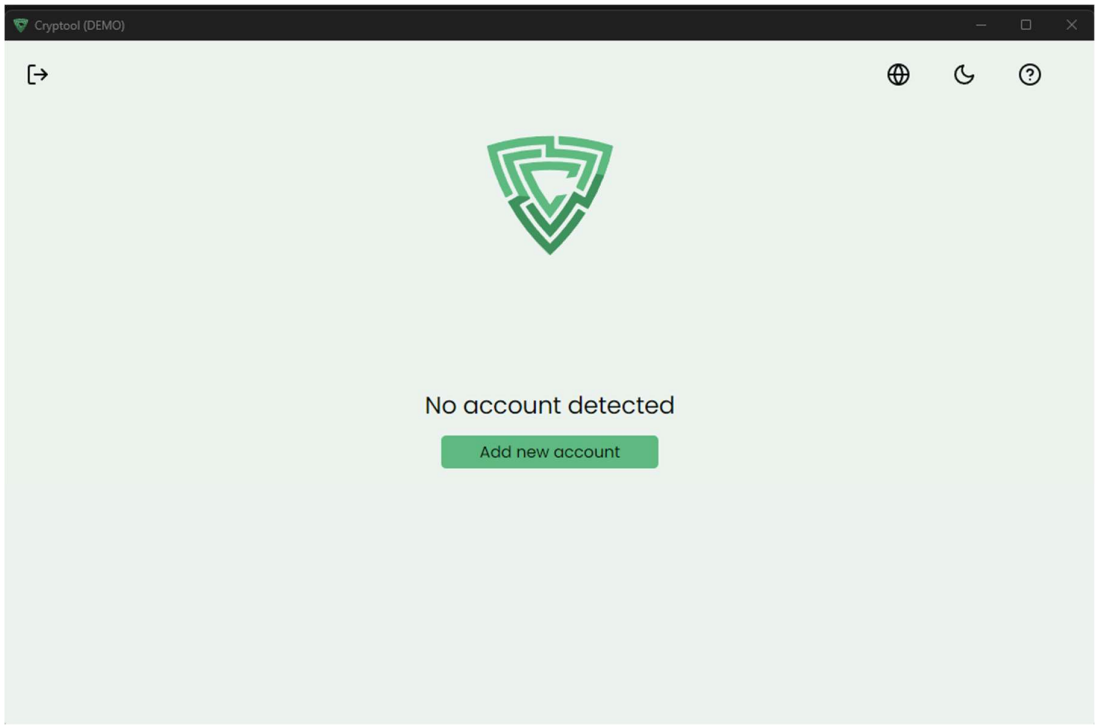

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.1:** Cryptool user guide (Register).

- The user needs to have a valid email and strong password to ensure secure access.

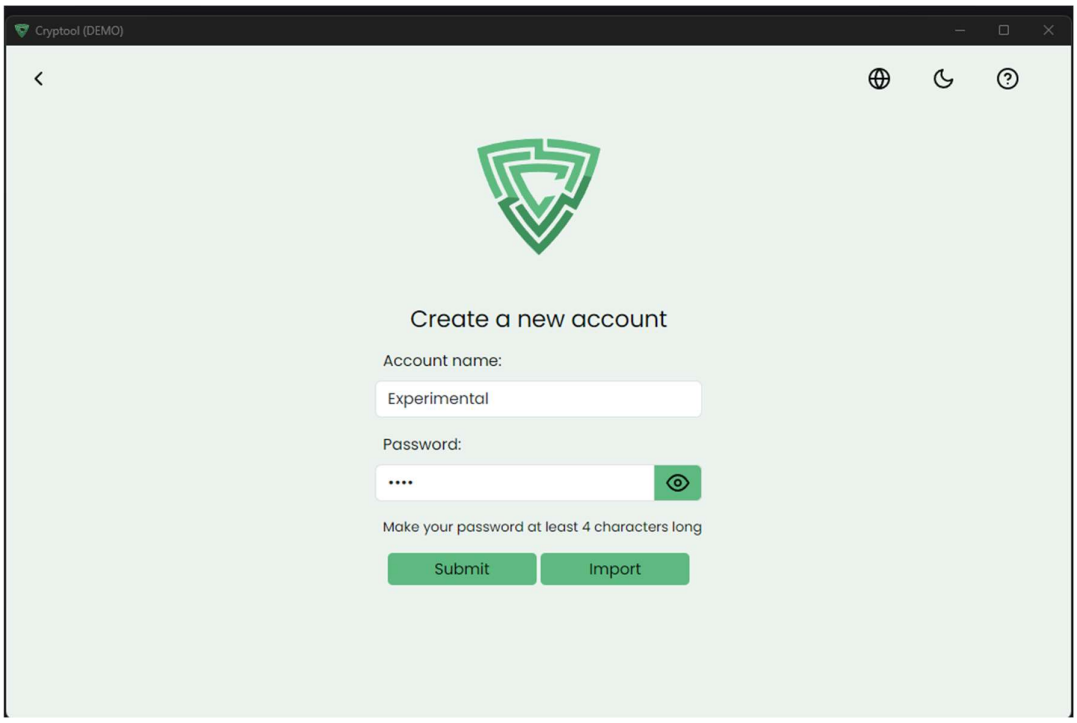

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**2:** Cryptool user guide (Register).

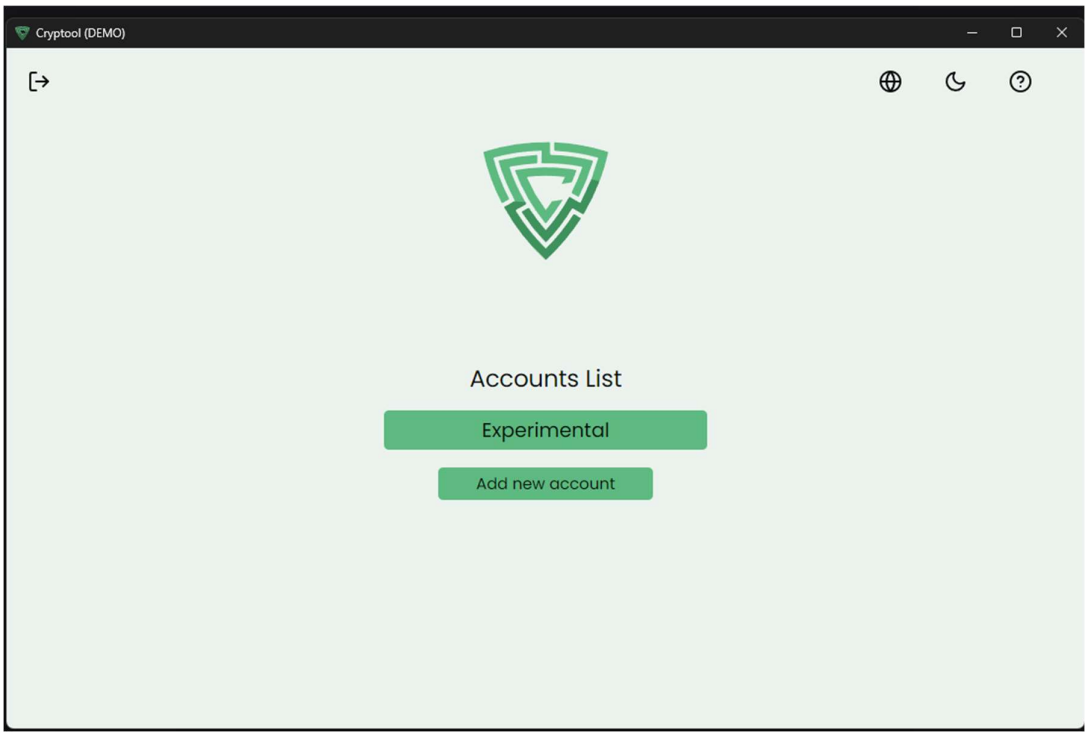

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**3:** Cryptool user guide (Register).

Import account file:

- The user could import account files from his local machine.

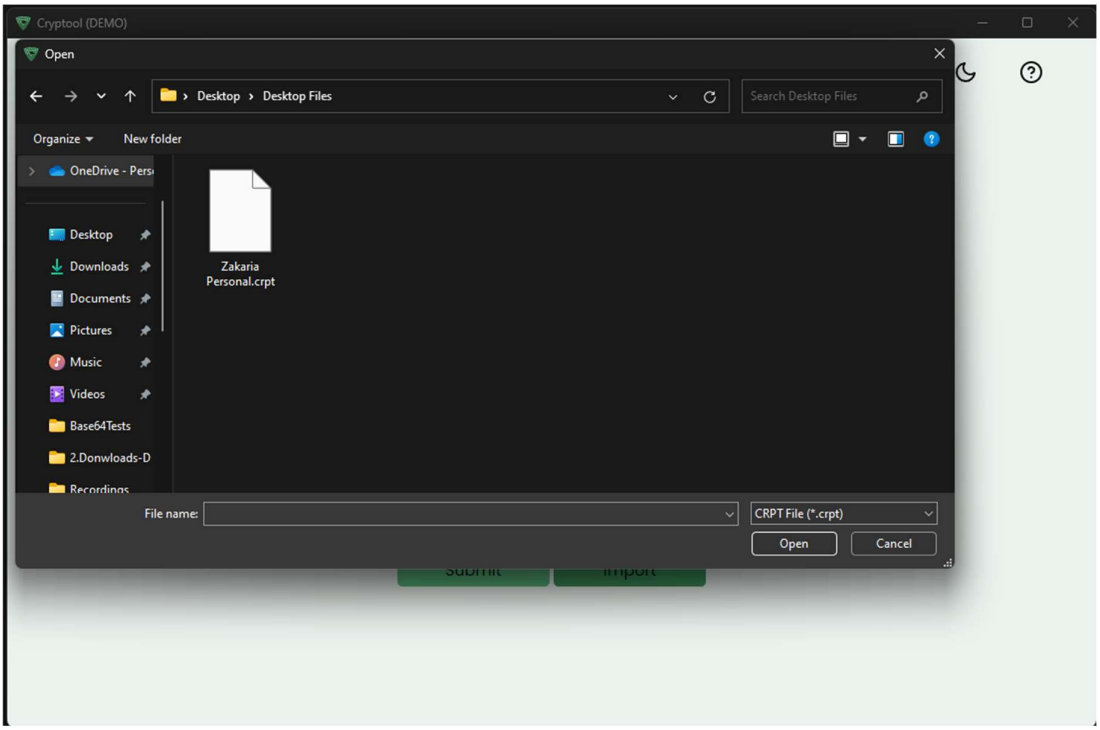

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**4:** Cryptool user guide (Import account).

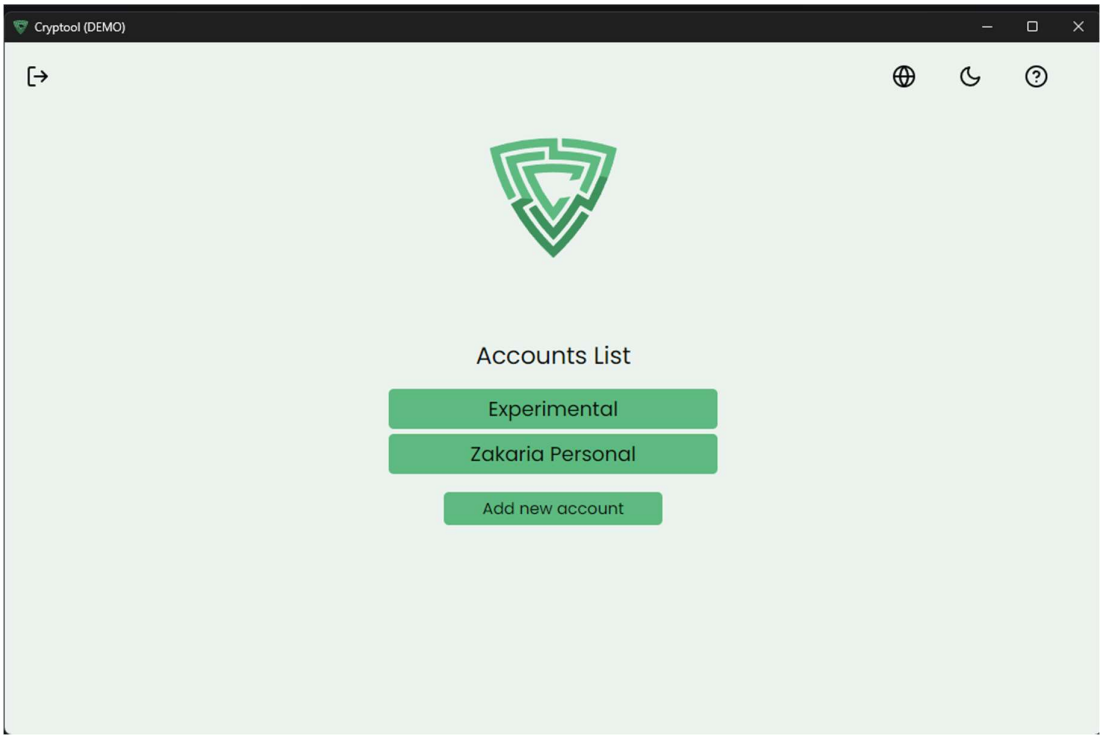

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**5:** Cryptool user guide (Import account).

Login:

- The user enters his email and password and the OTP code is sent to his email.

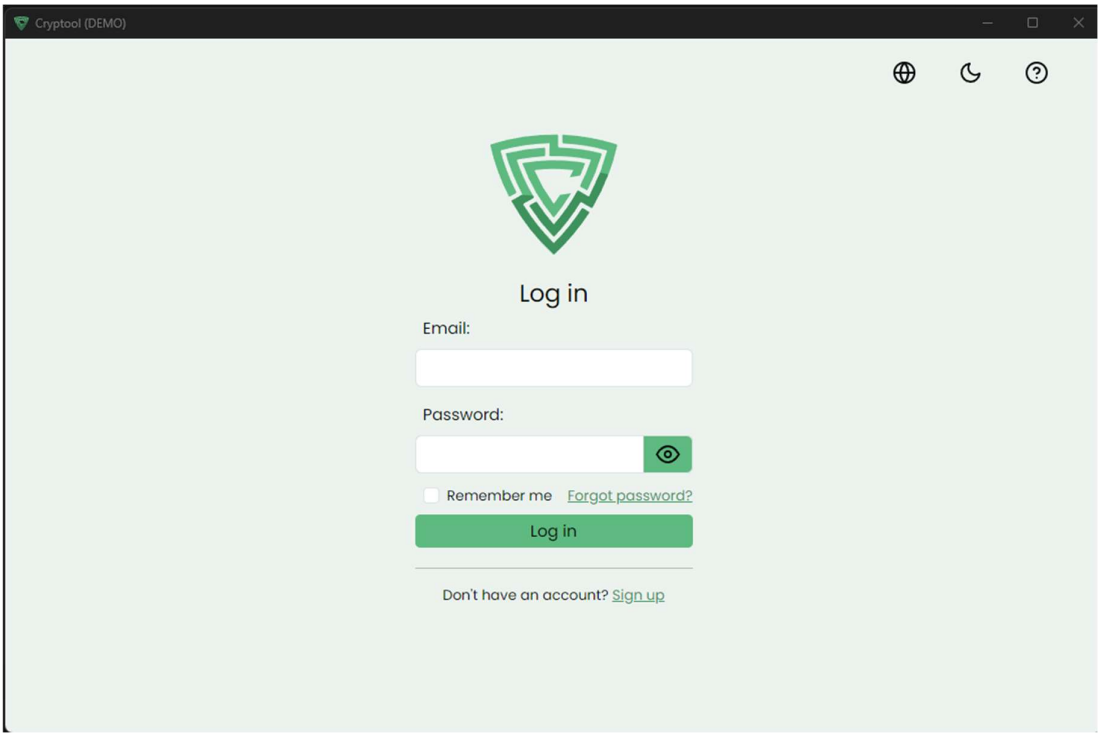

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**6:** Cryptool user guide (Login).

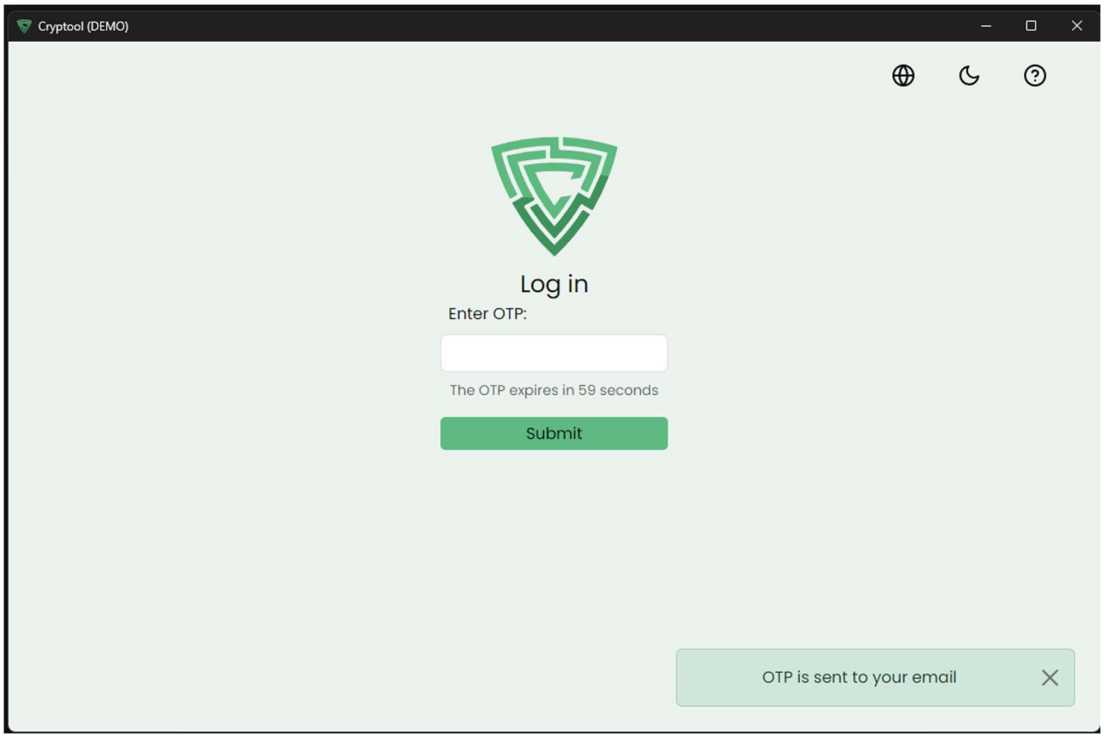

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**7:** Cryptool user guide (Login).

Adding a new account card:

- the user clicks the + button to add a new card.

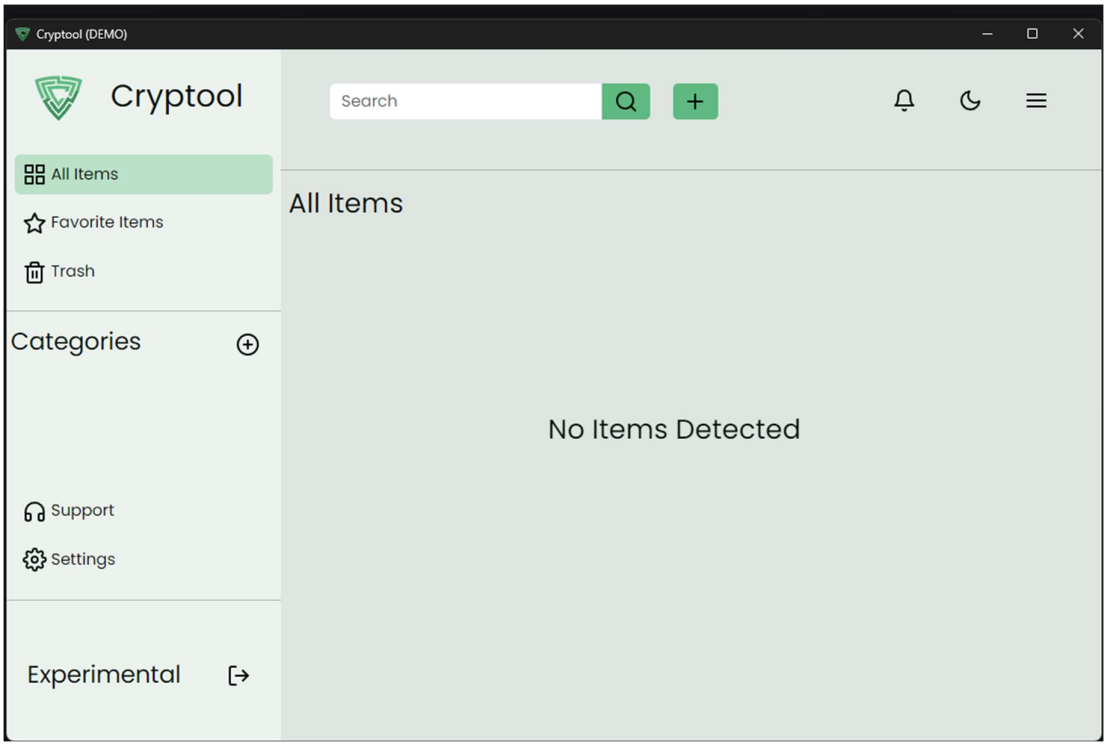

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.8:** Cryptool user guide (Add card).

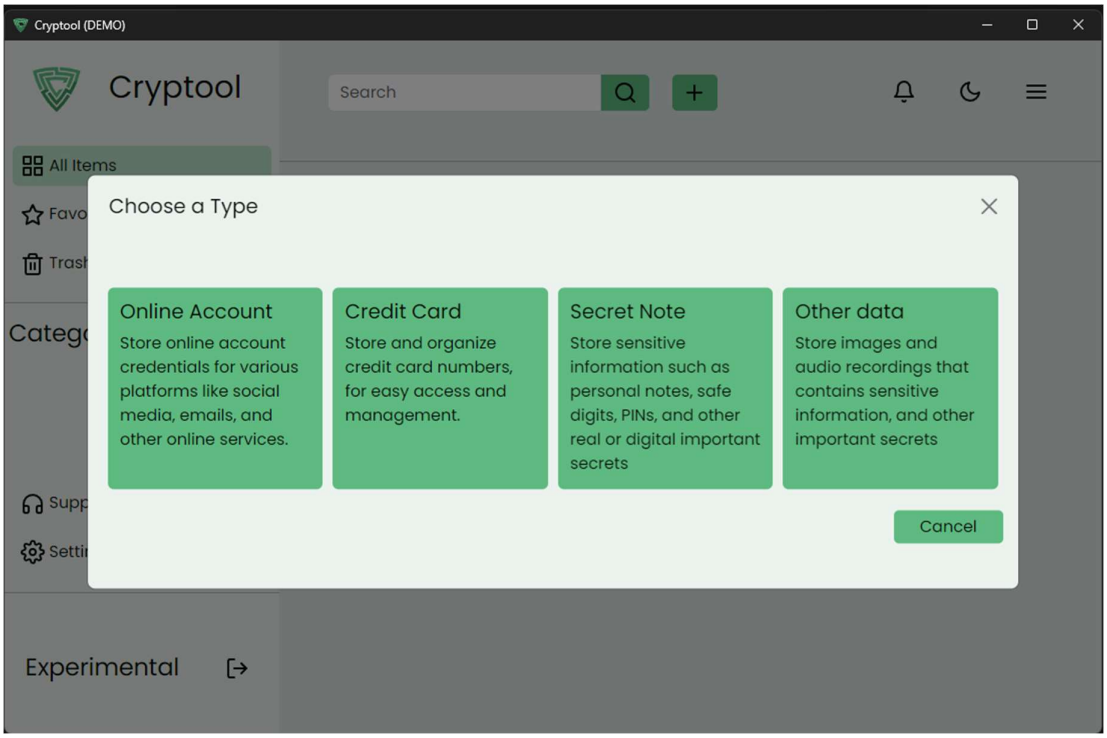

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.9:** Cryptool user guide (Add card).

- Add card information and clicks the **Add** button.

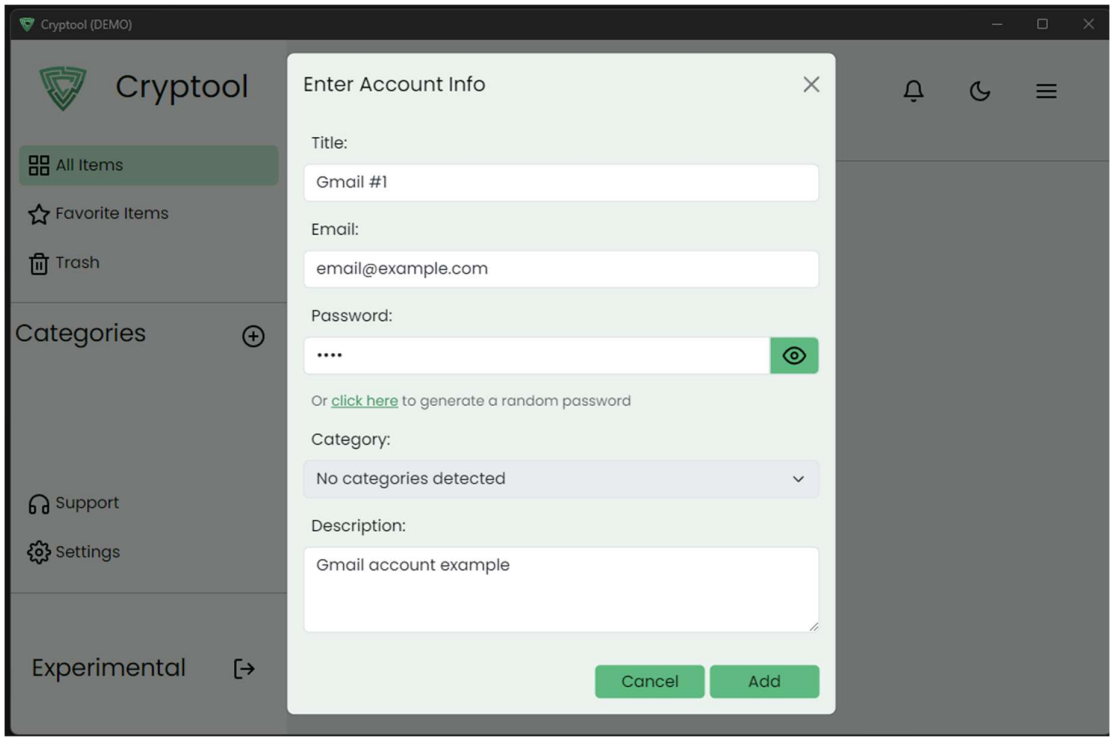

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.10:** Cryptool user guide (Add card).

- The user could display card details, add to favorite, edit, or delete a card.

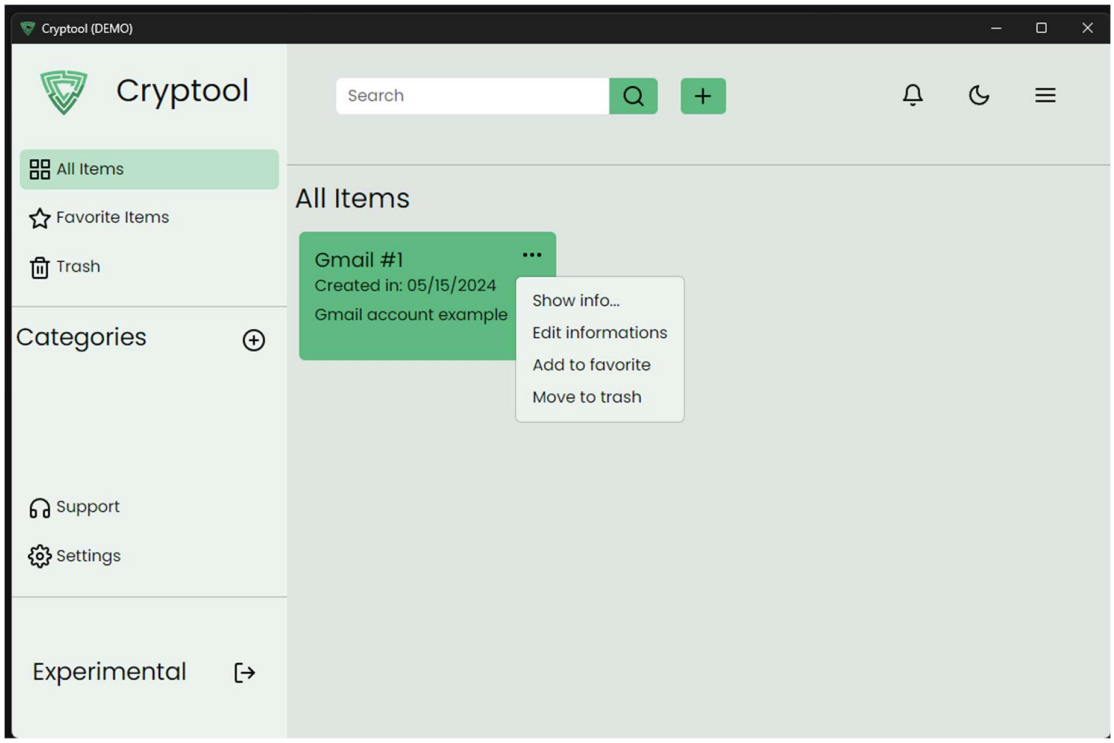

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**11:** Cryptool user guide (Add card).

Add category: the user could add a category folder to organize different cards.

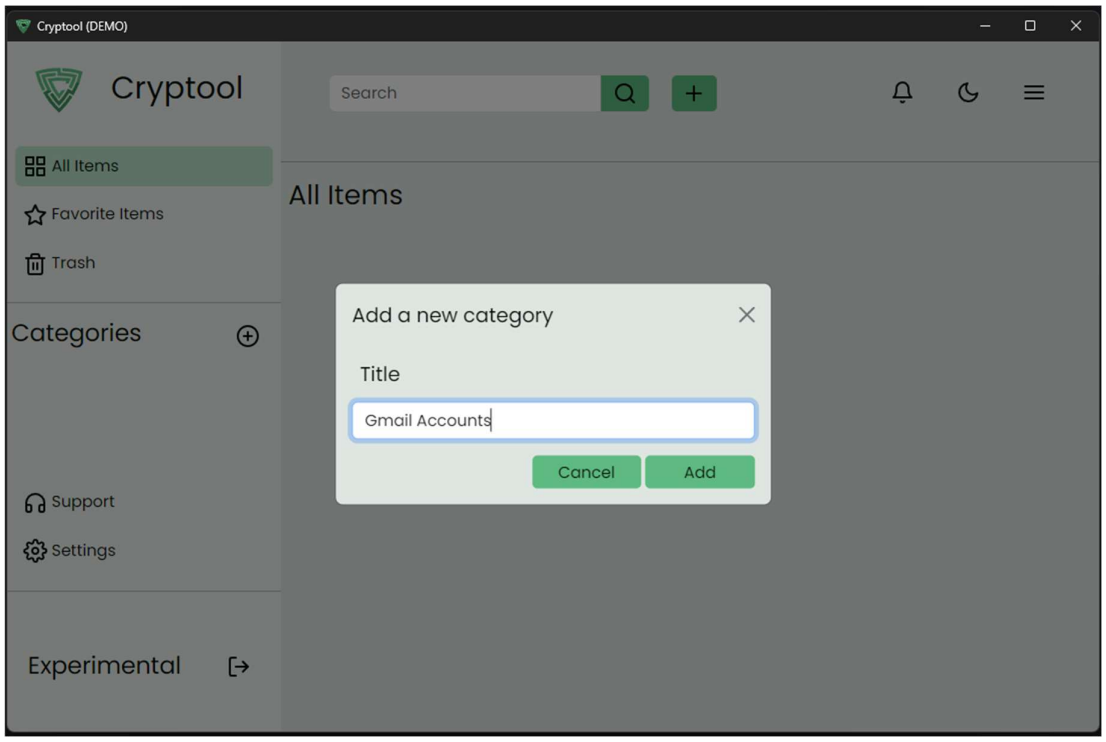

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**12:** Cryptool user guide (Add category).

Settings: The user is able to change their account name or reset password.

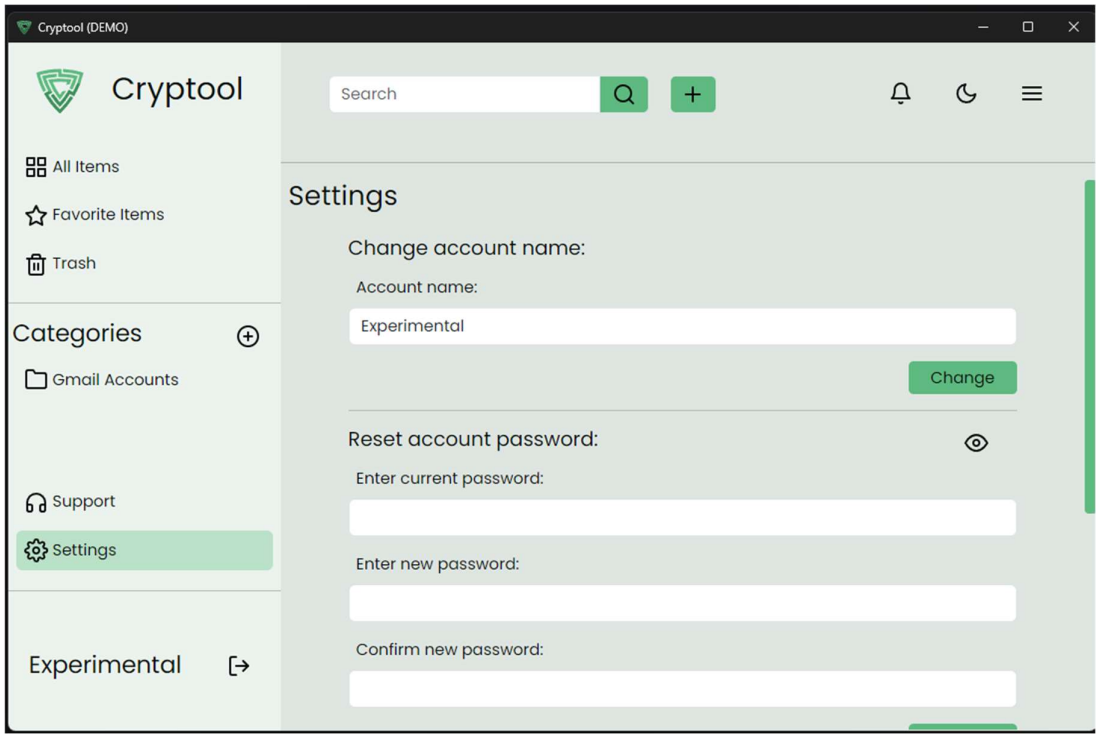

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici.**.13:** Cryptool user guide (Settings).

Export account file:

- The user could export the account by clicking the export file button and setting the directory.

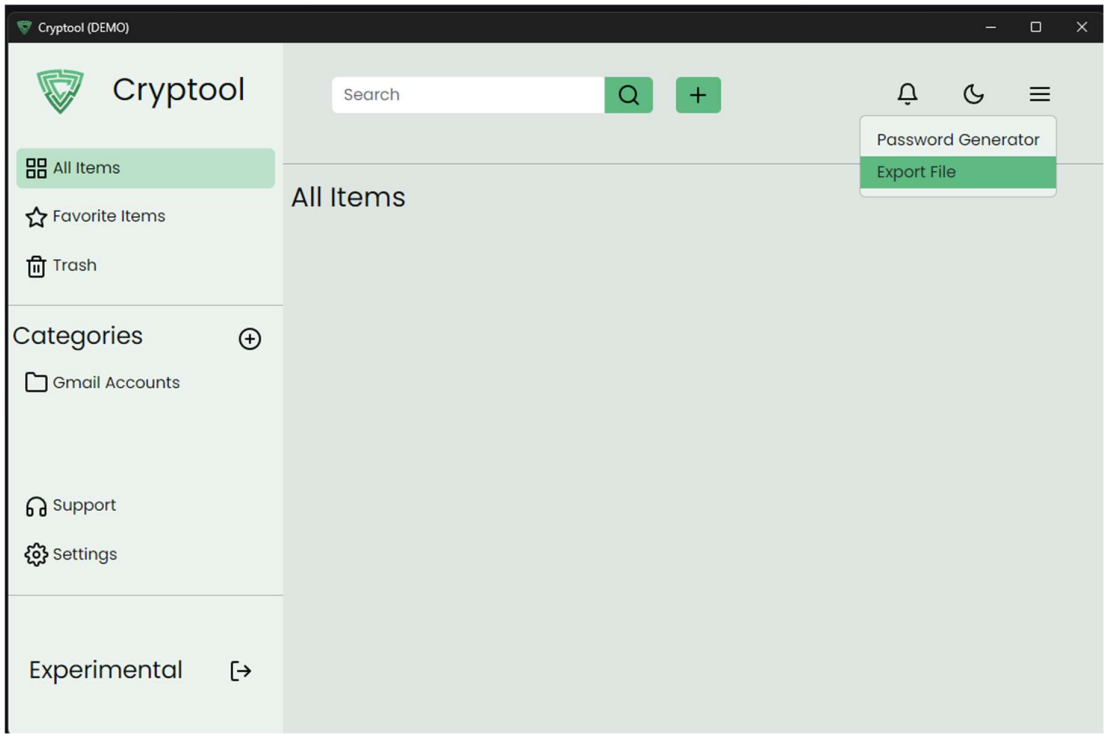

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**14:** Cryptool user guide (Export account).

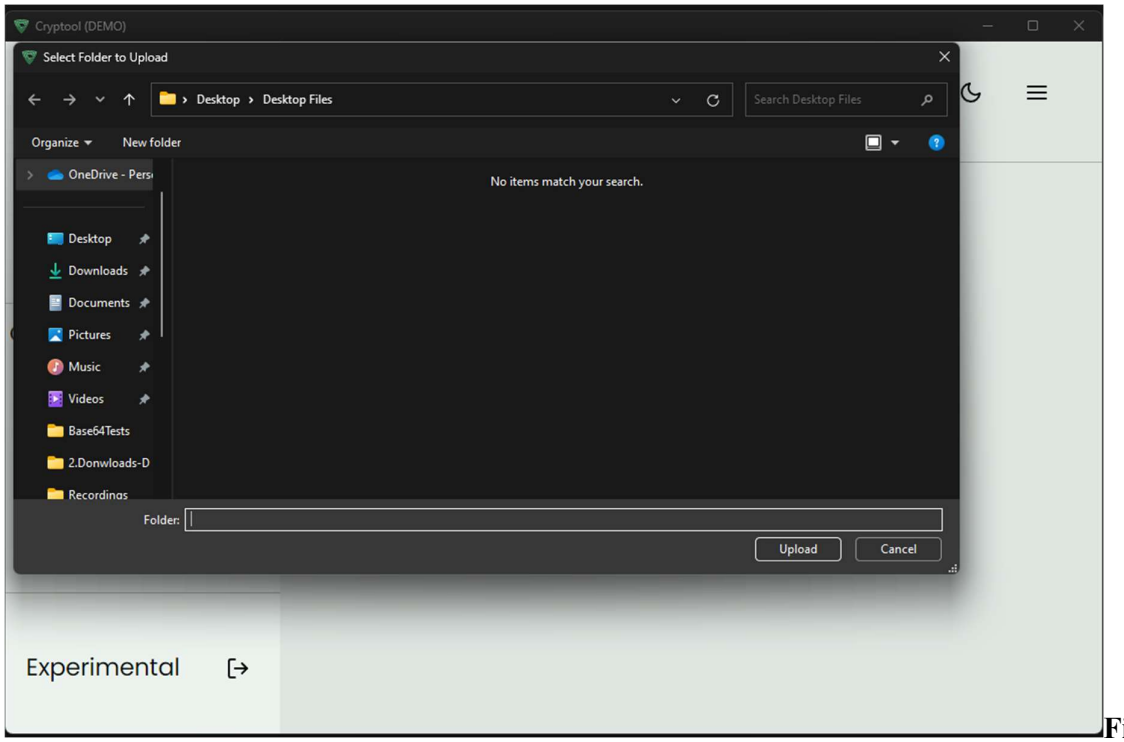

Fig. Erreur ! Utilisez l'onglet Accueil pour appliquer Title au texte que vous souhaitez faire apparaître ici..**15:** Cryptool user guide (Export account).

5.4.2. Disclaimers

Here are some disclaimers regarding the use of Cryptool Account Manager:

1. We are not involved in any shape or form in storing users' data on our servers except data used for authentication purposes like an email.
2. Cryptool Account Manager bears no responsibility for any risks that may arise due to carelessness in maintaining the strength and security of the account password, the user is fully responsible for the security of their account password.
3. At the time of writing, the Cryptool Account Manager is still in beta version, so it cannot ensure full compatibility, and it does not guarantee uninterrupted or error-free operation.
4. Since Cryptool is an offline application, the user must be aware of storage options and possible data loss. We do not impose any cloud storage solutions the user is free to choose.
5. Cryptool Account Manager is provided 'as is' without warranties and is intended for personal use. We do not claim any misuse or particular use-case issues.

5.4.3. End-User License Agreement (EULA)

This Software License Agreement is entered into between EZMSoft ("Licensor"), and the end user ("Licensee") of the software.

Permitted Use: The Licensee may use the Software solely for its intended purpose as described in the accompanying documentation. The Licensee may not modify, reverse engineer, decompile, disassemble, or create derivative works based on the software.

Data Storage and Backup: The software stores the user' data locally on the Licensee's computer or device. The licensee is solely responsible for backing up its data and understands the risk of data loss if the device experiences hardware failure or other issues.

Ownership and Intellectual Property: The Licensor retains all rights, title, and interest in and to the software, including all intellectual property rights. This agreement does not grant the Licensee any rights to patents, copyrights, trade secrets, trademarks, or other intellectual property rights.

Limitation of Liability: In no event shall the Licensor be liable for any indirect, incidental, special, or consequential damages arising out of or in connection with the use or inability to use the software, even if the Licensor has been advised of the possibility of such damages.

Support and Updates: The Licensor may provide updates, patches, or support services for the software at its discretion. Licensee agrees to comply with any applicable updates to ensure the proper functioning of the software.

Termination: This agreement is effective until terminated. The Licensor may terminate this agreement immediately upon notice if The Licensee breaches any provision. Upon termination, the Licensee must cease all use of the software and destroy all copies.

Miscellaneous: This Agreement constitutes the entire agreement between the parties regarding the subject matter hereof and supersedes all prior or contemporaneous agreements, understandings, or communications. By installing, copying, or otherwise using the software, the Licensee agrees to be bound by the terms and conditions of this agreement. If the Licensee does not agree with these terms, they should not use the Software.

5.5. Future enhancements and research directories

5.5.1. Feature enhancements

Feature enhancements and improvements are a crucial part of the development and growth of a software product, we aim to adapt the Cryptool Account Manager to achieve the latest technologies and features by introducing new features like adding business accounts management, security notifications, favorite cloud fast upload and more.

5.5.2. Security enhancements

Security is our foundation, we aim to enhance different aspects of security from encryption and authentication by optimizing the encryption algorithm and adding other authentication methods, to expanding the security by monitoring the dark web for data breaches, notifying users, and adding AI to help auto detect malicious actions.

5.5.3. Usability improvements

We aim to improve the ease of use of the Cryptool Account Manager by adding shortcuts and suggestions changing the design of the user interface and enhancing the user experience (UX) to make it more intuitive, efficient, and enjoyable for users to navigate and use.

5.5.4. Integration options

We try to make the Cryptool Account Manager more compatible with different platforms and operating systems by integrating it with other platforms like mobile and smart watches and adding browser extensions to enhance the convenience of using Cryptool to log in to online accounts.

5.5.5. Research directories

We look forward to doing further cybersecurity research, especially on the topic of data encryption and secure access, by having opportunities for industrial collaborations, explorations of different security measures, and academic research on the topic.

5.5.6. Long-term vision

With the evolving Web 3.0 technologies with decentralization as its foundation, decentralization is essential for ensuring that the internet stays a public resource that is healthy and accessible to all of us, rather than being dominated by a few firms and governments throughout the world. [30] Since user data privacy is our number one concern, we aspire to take advantage of such technologies for what it offers for data privacy and achieve a full-fledged solution for data management and security that is decentralized augmenting the user security on the web.

5.6. Conclusion

In this chapter, we presented a discussion on Cryptool Account Manager. We started with a comparison between our proposed solution and the existing solutions which helped in highlighting strengths and limitations and providing guidelines and precautions for users. Then, we suggested the future evolution and enhancements that can be done, reaffirming our vision to improve different aspects of the Cryptool Account Manager.

General conclusion

With the fast-evolving internet technologies and services, cloud computing provides unlimited infrastructure to store users' data online with less maintenance and high scalability. It has become a target of different attacks that affect users' privacy. In addition, with the fast race in Big data and data mining, and big tech companies and firms trying to collect their users' important information, the users find themselves between two risks of losing their data. With that in mind, an account manager solution that guarantees data privacy and security was proposed. The tool uses a custom encryption function coupled with AES to ensure the security and privacy of private data at the same time while keeping it simple and convenient for all users.

In the end, by developing a Cryptool Account Manager we have added a block to the field of data management and privacy. We recognize that there are still aspects left unexplored and improvements to be made. As such, we encourage future research and innovation in this domain, with future goals focusing on decentralizing user's data to prevent corporative dominance on the internet.

Bibliography

[1] "World Password Day - Global Survey 2024," 2024. [Online]. Available: https://bitwarden.com/resources/world-password-day/. [Accessed 03 May 2024].

[2] "Cost of a data breach 2023," 2023. [Online]. Available: www.ibm.com/reports/data-breach. [Accessed 03 May 2024].

[3] "Global Data Breaches and Cyber Attacks in 2024," 02 05 2024. [Online]. Available: https://www.itgovernance.co.uk/blog/global-data-breaches-and-cyber-attacks-in-2024. [Accessed 03 May 2024].

[4] "What is a phishing attack," [Online]. Available: https://www.cloudflare.com/learning/access-management/phishing-attack/. [Accessed 03 May 2024].

[5] "Browser Market Share Worldwide," April 2024. [Online]. Available: https://gs.statcounter.com/browser-market-share. [Accessed 04 May 2024].

[6] "Expediting changes to Google+," 10 12 2018. [Online]. Available: https://blog.google/technology/safety-security/expediting-changes-google-plus/. [Accessed 04 May 2024].

[7] "Data Protection API - Wikipedia," 29 Apr 2024. [Online]. Available: https://en.wikipedia.org/wiki/Data_Protection_API. [Accessed 04 May 2024].

[8] "Complaint – #1 in Brown v. Google LLC (N.D. Cal., 4:20-cv-03664)," 02 Jun 2020. [Online]. Available: https://www.courtlistener.com/docket/17216783/1/brown-v-google-llc/. [Accessed 04 May 2024].

[9] "Security Incident Update and Recommended Actions," 01 Mar 2023. [Online]. Available: https://blog.lastpass.com/posts/2023/03/security-incident-update-recommended-actions. [Accessed 04 May 2024].

[10] M. &. G. D. Singh, "Choosing best hashing strategies and hash functions," in *2009 IEEE International Advance Computing Conference (pp. 50-55). IEEE*, Patiala, 2009.

[11] J. &. P. S. Bonneau, "The Password Thicket: Technical and Market Failures in Human Authentication on the Web.," in *WEIS*, Jun 2010.

[12] "8.3 million plaintext passwords exposed in DailyQuiz data breach," 21 May 2021. [Online]. Available: https://therecord.media/8-3-million-plaintext-passwords-exposed-in-dailyquiz-data-breach. [Accessed 05 May 2024].

[13] M. Dworkin, "Hash functions CSRC," 16 Jun 2023. [Online]. Available:

https://csrc.nist.gov/projects/hash-functions#approved-algorithms. [Accessed 05 May 2024].

[14] "Public cloud computing market size 2024," Nov 2024. [Online]. Available: https://www.statista.com/statistics/273818/global-revenue-generated-with-cloud-computing-since-2009/. [Accessed 05 May 2024].

[15] B. E. H. H. Hamouda, "Comparative Study of Different Cryptographic Algorithms," *Journal of Information Security, 11(3), 138-148,* 2020.

[16] T. A. F.-B. A. B. K. A. J. G. Alabi Orobosade, "Cloud Application Security using Hybrid Encryption," *Communications on Applied Electronics, 7(33), 25-31.,* May 2020.

[17] M. J. Dworkin, *Advanced Encryption Standard (AES),* Gaithersburg, MD: Federal Inf. Process. Stds. (NIST FIPS), 2023.

[18] W. Stallings, Cryptography and Network Security: Principles and Practice, Pearson, 2022.

[19] D. M. a. J. R. a. M. P. a. S. Machani, "TOTP: Time-Based One-Time Password Algorithm," RFC Editor, 2011.

[20] "About the Unified Modeling Language Specification Version 2.5.1," Dec 2017. [Online]. Available: https://www.omg.org/spec/UML/. [Accessed 11 May 2024].

[21] "Green: Color Psychology, Symbolism and Meaning," 04 Mar 2024. [Online]. Available: https://www.colorpsychology.org/green/#:~:text=The%20Psychology%20of%20Color%20Green&text=It%20is%20regarded%20as%20the,optimism%2C%20hopefulness%2C%20and%20balance.. [Accessed 12 May 2024].

[22] "Documentation for Visual Studio Code," [Online]. Available: https://code.visualstudio.com/docs. [Accessed 12 May 2024].

[23] "Postman API Platform," [Online]. Available: https://www.postman.com/. [Accessed 12 May 2024].

[24] "Goodbye Postman Chrome app," 01 Nov 2017. [Online]. Available: https://blog.postman.com/goodbye-postman-chrome-app/#:~:text=While%20Postman%20started%20out%20as,the%20Chrome%20app%20in%20performance.. [Accessed 12 May 2024].

[25] "Introduction to MongoDB - MongoDB Manual v7.0," [Online]. Available: https://www.mongodb.com/docs/manual/introduction/. [Accessed 12 May 2024].

[26] "About GitHub and Git - GitHub Docs," [Online]. Available: https://docs.github.com/en/get-started/start-your-journey/about-github-and-git. [Accessed

12 May 2024].

[27] "Introduction to Node.js," [Online]. Available: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs. [Accessed 12 May 2024].

[28] "What is a REST API," [Online]. Available: https://www.ibm.com/topics/rest-apis#:~:text=A%20REST%20API%20(also%20called,transfer%20(REST)%20architectural%20style.. [Accessed 12 May 2024].

[29] "Best password manager of 2024," 10 May 2024. [Online]. Available: https://www.techradar.com/best/password-manager. [Accessed 13 May 2024].

[30] A. a. B. R. a. A. K. Goel, "Web 3.0 and Decentralized Applications," in *The 2nd International Conference on Innovative Research in Renewable Energy Technologies (IRRET 2022)*, 2022.

الملخص:

في هذه الأطروحة، تم تطوير برنامج مدير الحسابات لتوفير طريقة آمنة ومستقلة لإدارة بيانات إعتماد حسابات مستخدمي الانترنت . يعالج هذا البرنامج بعض الثغرات و العقبات الموجودة في البرامج الأخرى من خلال التأكيد على ملكية البيانات وآليات الأمان القوية. كان هدفنا هو تعزيز أمان المستخدم وراحته دون الاعتماد على خدمات أطراف أخرى لتخزين البيانات. توضح النتائج تحسين الأمان وتحكم المستخدم في بيانات اعتماد المستخدم. بينما تُظهر اختبارات الأداء الحد الأدنى من استخدام الموارد أثناء حالات الخمول والاستخدام المعتدل تحت الضغط، مما يضمن الكفاءة. بينما يضيف عملنا إلى خصوصية البيانات وأمانها، لا تزال هناك جوانب تتطلب المزيد من التحسين والبحث.

الكلمات المفتاحية: ملكية البيانات، أمن بيانات الإعتماد، إدارة البيانات.

Abstract:

In this thesis, an Account Manager software is implemented to provide internet users with a secure and independent method for managing their online account credentials. This software addresses common vulnerabilities and inconveniences found in other solutions by emphasizing data ownership and robust security mechanisms. Our objective was to enhance user security and convenience without relying on third-party services for data storage. The results demonstrate improved security and user control over their credentials. Performance tests show minimal resource usage during idle states and moderate usage under load, ensuring efficiency. Even though our work contributes to data privacy and security, some areas still require further enhancement and research.

Keywords: Data ownership, credential security, data management.