Strathmore University

Faculty of Information Technology

BITREST

Password Management System that employs the use of Emoticons (emoji) Characters

A documentation submitted partial fulfillment of the requirements for the Bachelor's in Business Information Technology at Strathmore University

Date of Submission: January 2021

Nairobi, Kenya

STUDENT NO: 100231

GROUP A

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other university. To the best of my knowledge and belief, this documentation contains no material previously published or written by another person except where due reference is made.

Admission Number: 100231 Date: 27th January 2021.

Signature: TKD.

Approval:

The documentation of 100231 was submitted for review and approval by the following Supervisor:

Supervisor Name: Nelson Ochieng.

Abstract

Through password managers, users can securely store their valuable information and sensitive information, from online banking passwords and login credentials to even passport and national identity numbers. But many password managers use centralized systems for data storage and security to protect users' data which can be hacked or breached by hackers in various ways without them even knowing about it. For example, in a scenario where an employee is terminated from employment and he/she had access to the system's codebase and was feeling vengeful about his/her termination, the employee could easily gain access to the company's serves and redirect user’s data and information to their private server to sell or conduct illegal activities with the data. This problem hence invoked the need for a system where a user would be the sole controlling entity over their data and where no man or system could never again breach or extract their information without their authorization. The Object-Oriented approach for design and analysis was used in the development of this solution with the application of use cases, data scheme diagrams, and entity-relationship models. In the development there was the use of programming language C# for both the front-end and back-end of this project. This system was mainly aimed at improving older methods to help the average internet user acquire the best possible data and internet security they deserve.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IT Information Technology

AES Advanced Encryption Standard

TOTP Time-based One Time Password

UI User Interface

URL Uniform Resource Locator

USB Universal Serial Bus

CDN Content Delivery Network

IDE Integrated Development Environment

MySQL My Structured Query Language

DBMS Database Management System

RAM Random-Access Memory

P2P Peer-to-Peer

CCV Credit Card Verification Numbers

Chapter 1: Introduction

1.1 Background

Nowadays information from the internet has become an increasingly important subject or asset in our daily lives and the ability to safeguard it and protect its integrity and stature for the continuation of organizational development and growth has become the top driving subject for many businesses today. Regrettably, there has not yet been a complete formula to ensure total and thorough information security up to this day (These Are the Top Cybersecurity Challenges of 2021, n.d.). According to the world bank, about 49.723% of the world individuals have access to the internet (Clement, 2018).. Around 90.2% of that number are using the internet to text message, 74.4% use it for online social networks and 68.5% use the internet for shopping, making reservations or other consumer-based service purchases (Clement, 2018). All these users including text and instant messaging, are required to have online accounts to keep unauthorized users from retrieving their sensitive data. There required to fill in either their phone numbers or email addresses for their user names and fill in password fields that must be to some character long, rememberable, and be uniquely different password in character complexity. An estimated 300 billion passwords will be in the hands of internet users in 2020 meaning roughly 40 passwords for each individual on the internet (Zane, 2018). On a personal level, this ordeal will prove to be a quite tiresome and frustrating task when the time comes for one to remember and retrieve their forgotten or lost passwords and, as Information Technology continues to expand exponentially, the scope for password use will also proliferate meaning, the need to protect and ensure our privacy will increase as well.

One method of ensuring credential safety is using a different unique password on each website accounts you create. Unfortunately, most people use the same passwords they've used before on their different internet accounts. From a survey carried out by Cyclonis (a cloud-enabled software development company based in Ireland), of the 275 subjects studied, 83.15% of the respondents who were recorded, were found to be using the same passwords for most of their internet account (Zane, 2018). Changing passwords, a lot can be very and difficult leaving those who can't remember they recently set passwords stranded in a sequence of hefty procedures to retrieve or renew their passwords.

To solve such a worrying and problematic issue for users, password managers were developed. They help eliminate the problem we have by generating unique passwords and storing the

passwords for the user, but criticism has arisen about the technology used and whether it will truly protected user passwords as efficiently (Fowler, 2019), leaving users occasionally to deal with the impression that one day their data may be hacked. Companies like Google or Kaspersky which are both very large companies in the technology industry are tasked with storing users' valuable information and from past incidents like Facebook's recent data breach or Amazon's continuous occurring security breaches (The Guardian, 2018), it is now very likely that companies like Google, who provide their chrome password manager, aren't completely secure from intruders and penetrators as well. Thus, my proposal for a more decentralized and localized password manager.

In decentralized systems, there is no single entity in the system that is the dominant authority. Systems are interconnected to each other through nodes, where each node makes its own decision. Each owner in the system stores a copy of the resources the user can access hence more convivence and when one or more of the central servers fails, the other owners can continue to provide data access for users. Hence, the user controls where his/her data is hosted.

1.2 Problem Statement

With the countless number of on-growing websites requiring us to create online accounts, it has become increasingly harder each day to record and store our login and privacy information details. A tremendous amount of work goes on to remember our passwords, which sometimes can lead to infuriating moments especially when there is a forgotten password, long and complex to remember. Password managers help us eliminated this concern however, most companies today like Kaspersky, use centralized data storage methods to store their user's information providing enough room for user data infiltration (The Difference Between Centralized and Decentralized Networks | SolarWinds MSP, 2018). If systems are breached, the hacker can simultaneously gain access to all user information stored on the system. It can also be very expensive to purchase a security firm's software to store our information for example, the Kaspersky's password manager that charges its users a monthly subscription $14. Many of the password managers as well do not offer command-line tools for developers who constantly use terminal or windows shell to access file utilities that require passwords. The provision of the automatic auto-fill features in the system is a problem as well because it creates new vectors for attackers to use when trying to infiltrate the systems i.e., a sweep attack (Hoffman, 2020).

1.3 Aim

To develop a password management system that will aim at running locally as a desktop application on users' devices. Users input and login information will be directly hosted on their computers, accessed with the use of their master key during login. The master key will be encrypted and become a necessity after each complete login session period. Due to the continuous use of characters and alphanumeric symbols used today, the system will implement the use of long and complex emoticon (Emojis) characters for password generation to increase the number of permutations previously available and make it harder for an attacker to infiltrate the systems.

1.4 Objectives

The main objective and purpose of this study is to develop a desktop password management system, making it as locally available as possible and increase user password safekeeping by increasing previously available permutation or combinations.

1.4.1 Specific Objectives

i. To determine the needs required to build and attain efficient security for a user's data without compromising the integrity of a device.
ii. To analyze, review, and critique the previously created password management systems and their supporting technologies.
iii. To develop the highlighted system in (1.3) helping users eradicate the restlessness of trusting centralized application vendors with their data.
iv. To conduct the testing and validation of the system among potential users of the service.

1.5 Justification

By using of heavy data encryption algorithms and emoticons in a simple but sophisticated UI, this project will ensure that users' information is comprehensively and securely stored in the user device predominantly. Due to the current expansive availability and use of centralized systems, data breaches have become a very high costly endeavor in terms of damages, halting the growth of large companies. Centralized system data breaches are very difficult to recover from as well, which is why access control to our own information and data is very important.

In 2019 it was reported that Equifax, the database company that stores most Americans' social security numbers was exposed in a hack (Solon, 2017). The hack exposed more than 143 million Americans' social security numbers which could be used to exploit millions of people data involved even to this day. This goes to show that the adoption of a more decentralized approach could have the prospect of aiding us reduce the harms involved in the foreseeable future because users could now make the choice of where to store or host their data.

1.6 Scope and limatations

The scope in this project will seek to eliminate centralized password management systems and primarily interact with the host's device alone. It will revolve around high-security functions and bit algorithm functions for increased data security and data loss prevention. Password generation will be implemented into the system, helping users generate random alphanumeric and emoticon (emojis) combinations, reducing user's chances of data theft.

A limitation to this project is that it will not cover other domains of execution like mobile based platforms and will only cover desktop devices. Due to the use of the current methodology, I'll be attempting to use prototyping which will be very time consuming and may require extensive user collaboration in order for the project to be successful.

Chapter 2: Literature Review

2.1 Introduction

In this section, the primary focus will be to review and revise current and previously written literature on personal user data security management and storage software. It will also explore the current architectures that have been used to build theft-resistant storage and management software, their current effectiveness, and challenges faced. The section will review the security of password manager database formats used by popular password managers, also defining the two realistic security models designed to represent the capabilities of real-world adversaries.

2.2 Current supporting technologies and tools in data security

Password manager software are very critical pieces of software relied upon by users to securely store valuable and sensitive information, from online banking passwords and login credentials to passport and social security numbers. Because the number of provided services on the net is on rise, the quality of passwords a basic user is going to be required to recollect will increase, to the point where it's now not feasible anymore for people to recollect or remember their contemporary passwords, strong enough to protect every internet account the user holds (Hoffman, 2019).

The foremost common way and solution used for the problem faced by users today is the reusing of passwords. Maybe with a few altering of characters on their alternative accounts. This method, however, only increases the probability of damage to the user if your password is stolen, cracked or if a service that has access is breached or compromised. Another possible approach to the problem commonly used is the use of a password manager. This is often a software that needs users to recollect or remember one strong master password alone, used to decrypt the password manager's database (Gasti & Rasmussen, n.d.). Trying to recollect one password to access all your accounts sounds more feasible than trying to recollect hundreds for varied accounts. The user of a password can still also enjoy the advantages of acquiring high security by employing a different password for every online service used.

Users of storage managers enjoy many benefits, including domain names or URLs stored alongside the corresponding passwords, accustomed to fill login forms automatically. Users who depend upon this software also abate the risk of typo-squatting and phishing attacks.

Whether or not a user is directed to a malicious website that's designed to seem as dead ringer for the web site the user expects, the password manager won't login automatically, providing an additional layer of protection. Password managers protect their content from unauthorized users typically through encryption/ decryption key generated from a master password entered by the user. This protection can although come at the price of allowing users to store the password manager database on untrusted storage. Some producers and developers recommend to their users storing password databases on USB sticks, within the cloud, or perhaps mobile devices to permit convenient methods of accessing stored passwords. These storage options however, also can allow convenient access to stored passwords. These storage options enable potential aggressors to induce hold of the database, even when a password database is stored on an area drive. The attacker could use various means at their disposal.

When the password manager storage and database format is insecure, then all the benefits of a decent password manager can be wasted as well, and users could be less secure and at risk of attacks that will result in leakage and manipulation of user's private data

The current password managers differ in many ways and many of them out there have different aspects to them to securely manage and store users' information but what are their specific unique attributes that make customers acquire the compulsion to purchase them for use.

2.2.1 Google Chrome password manager

It is well known that Google has a distinctive history over its competitors when it comes to well-thought and convenient products and with its chrome password manager features it continues to prove that they are leads in industry of technology. Chrome and Android users, in fact, probably use the smart lock in their everyday lives without having necessarily made the choice diligently or noticing it conscientiously. It serves users by syncing their passwords between all their devices and automatically fills in web forms and signs them into their mobile apps.

A drawback to these is that you're giving your most personal information to Google, who store your data in closed-source software. It also doesn't provide its users with integrated UI to generate a random password when signing up for a new websites, hence users rely heavily on the bad practice of having the same or just the slightly different passwords on all their apps, compromising all their account if one from the various ones the users has gets hacked (Derousseaux, 2017). It can't be used for cryptographic key pairs but on the bright side being a google product it has a strong vendor-locking effect. It's mostly optimized for Google's

Android and chrome and becomes less convenient if a switch to other browsers occurs like IOS or Firefox.

2.2.2 LastPass

With LastPass software, you can sync your information across all your devices, share them with other people of your choosing, auto-fill forms on web pages, use their integrated strong password generator, use strong multi-factor authentication and secure notes that can be used for any plain text secret for a monthly subscription 2$. With this software however, the drawback comes with using being able to trust closed-source software (centralized servers) with our sensitive data. This means that someone with access to LastPass' codebase could tweak their front-end application or browser extensions, in case of a breach, to redirect all the users' information, decrypted after the master password submission, to their servers for their malicious causes and motives. It can occur whenever one of their employees goes berserk or rogue or when someone breaches they servers without them noticing it.

2.2.3 KeeWeb

Keeweb is another free cross-platform open-source compatible with an in-built KeePass vault file format. It has almost the same feature as LastPass except form integrated secret sharing and form auto-fill. It can be however be incorporated with their web version from their GitHub pages through Cloudflare CDN, synced with a Dropbox account (Derousseaux, 2017). But this process includes using a centralized storage system which is not completely a secure method of storing your information. It's also a closed-source software like LastPass. The initial product lacks an auto-fill feature which is good because it can't be exploited as an attack vector for sweep attacks. An Advantage to KeeWeb vaults is that it has a setting called key encryption round that has default of 6,000. It can be increased to prevent brute force attacks if someone gains access to your vault while slowing down your typical encryption/decryption process by just a second.

2.3 Problems, challenges and architectures

Although the products above in (2.2) prove useful to consumers they don't completely secure users from information theft and have loopholes that somewhat need fixing. The reason a password manager is good is that it allows you to generate long, unique, "unguessable" passwords for every site. The downsides to this are that it's not often flexible enough to take the various rigorous complexity and size restrictions that many sites try to implement. Also, it isn't very easy to deal with, when the resource requires you to change your passwords periodically, and that resource keeps track of similarity (R/AskNetsec - Are Password Managers Safe? n.d.). If someone learns or can figures out your password by correlating the results of multiple breaches, you can then lose everything.

Chrome's password manager tools that are meant to save your password expose you more to the risk of attack, regardless of who makes them or who gives them away. Closed-source software systems have proved unreliable with large cloud-based servers being breached to attain credit information for malpractice (Merie, 2018). Users have to live with the fact that most of the data they provide from your personalized accounts, they provide to Google's closed-source software. Other password managers, including Google, Mozilla or Explore, do not generate random passwords for new websites, hence the same old practice of reusing passwords is employed compromising the account in a scenario where one gets hacked (Password Manager vs Remembering Passwords, 2017.).

With today's increase in information technology developers, creating and using different software tools to come up with cutting edge technology, it has become increasingly hard for them as well to keep track of password requests that occur in the tools they frequently use like Terminal, window's shell or Command-Line. Password managers lack command line integration properties to help developers keep track and manage their passwords from terminal or command line.

Another problem that seems to be a reoccurring one as well with password manages is their high-level of pricing for full features that password managers offer. Some password managers contain hefty prices for a normal consumer or user to even consider purchasing e.g. Kaspersky, LastPass, with users required to pay monthly fees for their premium services that might not completely safeguard them against internet attacks. People who are located in remote areas where the internet use might have just started increases and live on expenses of almost less

than $5-3 a day can't enjoy the benefits of password managers without paying for high premium fees each month. They can't enjoy the benefit of avoiding intruders or attackers that might like to exploit them for their own personal information to satisfy the malpractice needs.

Some password managers provide a password autofill feature which can be exploited by infiltrators as a considerable vector option for intrusion or attack into their user's devices. There are largely two ways in which this sort of attacks can happen: through an evil coffee shop attack and a sweep attack.

2.3.1 A Coffee Shop Attack

Attackers here are assumed to be able to enact an active man-in-the-middle network attack to attain access and modify arbitrary network traffic from a user's machine. However, there is no requirement that the user explicitly visits or logged in to any particular site to steal the credentials for that site. A rogue Wi-Fi router in a coffee shop for example is all that is needed to connect to and your personal digital information is gone. These kinds of attacks require attackers to temporarily control a network router and are much easier, thus more likely to happen in practice. The attacker doesn't require to interact with the victim website and is occasionally unaware that password extractions are taking place.

2.3.2 Sweep Attack

They mostly target password managers that support the autofill of password fields. They target users when they connect to the Wi-Fi hotspot controlled by the attacker.
When users launch the browser, the browser is redirected to a standard hotspot landing page asking for user consent to standard terms of use. This is a very common behavior for public hotspots. The user however, has no clue or idea that the landing page contains invisible elements that implement the attack. Once the JavaScript in the attacker's page has the desired password, exfiltration is pretty straightforward. An approach is to load an invisible iFrames and pass the credentials as parameters, another is to modify the action of a login form to submit to an attacker-controlled site.

Well, whether one believes that these scenarios and outcomes are reasonable or likely to, occur or not, is entirely up to the user's choice. They may be completely damaging or manageable by the user

2.4 Gaps, Analysis Architecture and Solutions

Thus, an improved way to managing passwords is through a decentralized system that will keep user's information on their local machines, as well as give them the option to choose where they can store their information hence reducing the loss of a user's private data. A software being decentralized means its mainly open source and isn't tied to any service provider. Developing a decentralized system will provide the user with the ability of importing, encrypting, and storing passwords locally on your device without counting on a cloud-based system (Corbyn, 2018). If one were to choose to use a centralized system, they'd need to store and sync the contents of their encrypted files or folders on safe proof cloud-platforms systems like Dropbox due to their encrypted file sharing. But on a personal note, I would not recommend it.

Another gap that password managers seem to be ignoring or may be unaware of is lack of command or terminal integration tools to configure with users, groups, vaults, and items when standard password managers are not completely adequate for the users. Such integration tools could help users manipulate file management utilities once organized into meaningful folder hierarchies, copied directly from computer to computer, and generally, manipulated using standard command lines. Hence through this project, tools like this will be integrated for its beneficial use by developers and other users who use the command-line or terminal often.

The combinational use of emojis (emoticons) and alphanumeric characters increasing the number of permutations and difficulty for organized attacks within the internet for users is another gap that hasn't been tapped into in password managers. The topology of a password manager is a crucial metric to measure its guess ability. Emojis are visual mediums and the human brain is known for memorizing images well, thus we are more likely to recollect a weird emoji face better than a random string of characters or numerals. A study was conducted by the Intelligent Environments, a British software development company, to check how well people would react to emoji password (Intelligent Environments, 2015).

The subjects who are UK citizens aged 18+ were asked to form a four-digit passcode comprising of 4 emojis selected from a collection of 44 smileys. The confirmed result from the sample of the study subjects was that emoji passwords are way harder to hack compared to PIN codes. If we take Emoji Passcode's set of 44 smileys then almost 3.5 million permutations of

non-repeating emojis are created, whereas the quantity of unique permutations of 4 non-repeating numbers is barely 7,300. additionally, thereto, participating members within the research reported that they'd more fun with emoji passcodes and experienced fewer problems with memorizing their new password which are a few things regular passwords will never be able to replicate (Now You Can Log into Your Bank Using Emoji, 2015).

The database formats currently in use by stand-alone and browser-based password managers is also an opportunity to attempt in password management software. As an example, in Google chrome's password manager, the database format they use store names and passwords in an SQLite database gets into the user profile directory. This database provides neither secrecy nor integrity. they will optionally store all browser preferences, including passwords, on Google's servers to permit synchronization between different devices. This however, could mean that any user with access to the database file can recover all its content and make arbitrary modifications (Encryption - How Secure Is Chrome Storing a Password?, n.d.). As such, users cannot depend on Chrome's password manager for integrity or secrecy of their data. Keepass database is comprised composed of files, divided into two sections: an unencrypted header and an encrypted body where the body stores the encryption of the assorted database entries. it contains a hash where it's computed each time the database is modified and is employed to test integrity. The hash verification doesn't fail. However, verification from other sources indicates that versions like v.0.4.3 are liable to attack (Merie, 2018). Moreover, if the victim makes any change within the modified database, KeePass stores only the entries displayed which might result in silent corruption of the database.

It seems fair to want that a password manager that asks users to authenticate themselves with a password, a minimum of provides secrecy and data authenticity. This may currently be only achieved by one password database format. For general purposes, password managers should be clear about the safety provided by the underlying database format.

2.5 Related Work

With the concept of information and data security storage and management systems are becoming well known and utilized by people everywhere on the planet, but very little scientific literature on the topic.

To resolve the same issue, a report was published by IBM Research written by Blasko in 2005 (Blasko, 2005) proposing a Wristwatch-computer Based Password-Vault. Blasko described the planning and implementation of a wearable computer with wireless connectivity, processing, input, and display capabilities, that are meant to store a user's passwords for various services (Mühle et al., 2018).

One year later, Gaw and Felten published a study of Password Management Strategies for Online Accounts. The authors studied a percentage of passwords 49 undergraduates had, and the way they often reused these passwords. At that point about 38% of the people participating in the study used password managers. Over two-thirds of these used online, web-based password managers. With the inclusion of password managers in popular browsers, that number is presumably significantly higher today (Rosenman, n.d.).

In 2003 Luo and Henry proposed a technique for safeguarding multiple accounts. Their solution requires a user to recollection was just only one password, called a standard password, to access any of a variety of accounts. The authors proposed an internet-based implementation with a password calculator that was written in JavaScript.

In 2009, Englert and Shah published a paper on the planning and implementation of a secure Online Password Vault. This word describes an architecture where encryption and decryption are completed locally on the user's machine, but storage did online.

In Baelenko and Sklyarov analyze the protection of several password manager applications running on iOS and BlackBerry smartphones. Their analysis focuses on a passive adversary, who is in a position to access a password database at rest. The goal of the adversary is to analyze the database master password, therefore, accessing the protected data. The authors show that almost all password PIN, or don't use expensive key derivation functions to compute the database encryption/ decryption key from the master password. This enables an adversary to perform password recovery attacks during a relatively short time for low-entropy passwords (Rosenman, n.d.).

2.6 Conceptual Framework

The diagram below demonstrates some of the functions and features that the information and data storage management will perform. The user of the system will be required to login to the database system using his/her master password from their respective devices and access a list of their user credential to log in to their internet-based accounts with the help of a browser extension. The browser extension will retrieve the user encrypted passwords from a password manager database and ask the user to manually input their information onto the login form then will be submitted for their specific requested access to the website. The password manager will offer the user the option of syncing devices to control their account details and information as well on their various owned devices. A feature to fully encrypt the user's personal hard drive will be included in case the user lost his/her master password but will also be provided with the option of storing and encrypting their information on other drives as well.

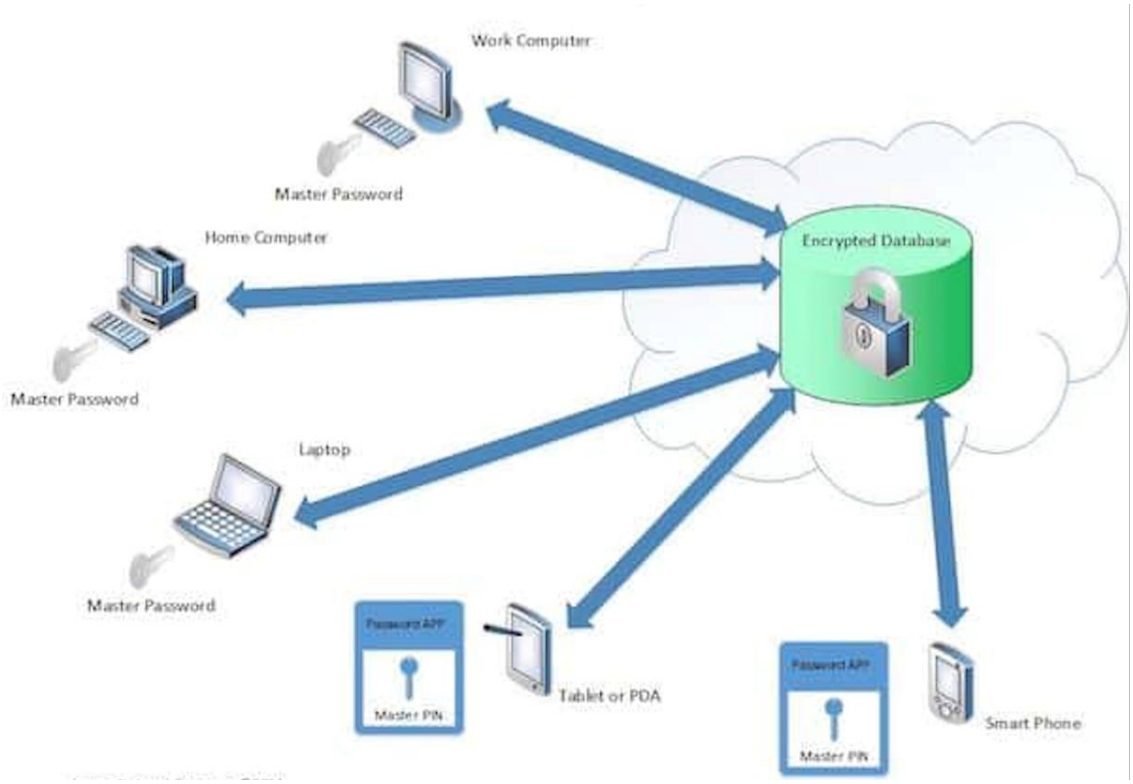


Figure 2. 1: Conceptual Diagram

Chapter 3: Methodology

3.1 Introduction

A methodology illustrates and defines the wide and expansive theoretical and rational underpinning to one research methods, including whether one is using qualitative or quantitative methods, or a mixture of both methods with a clear detailed reason for its use (Writing Your Dissertation: Methodology | Skills you need, 2013.).

The methodology chosen for this project is the Prototype methodology. This methodology allows the developers to create only the prototype of the solution to demonstrate its functionality to the clients and make necessary modifications before developing the actual application. Specification, development, and validation activities are interleaved rather than separate, with rapid feedback across activities.

3.2 Prototype System Development Methodology

The prototype methodology or model requires that the development of a working prototype be built first as preparatory for the development of the actual product/ software. A Prototype that is usually built transpires as a very rudimental and unrefined version of the anticipated software revealing low reliability and scalability, limited functionality, and inefficient performance compared to the actual proposed product. The customers in some instances have only a general view of what is expected to form the software product.

Steps involved in the prototype model include the gathering of initial requirements, design of the prototype model, development of the prototype, customer validation, review, and updating. Only once the customer is satisfied with the prototyped products, is when the development of a more refined actual product begins to take place. The product is then tested and maintained by ensuring the bugs experienced by users are eliminated.

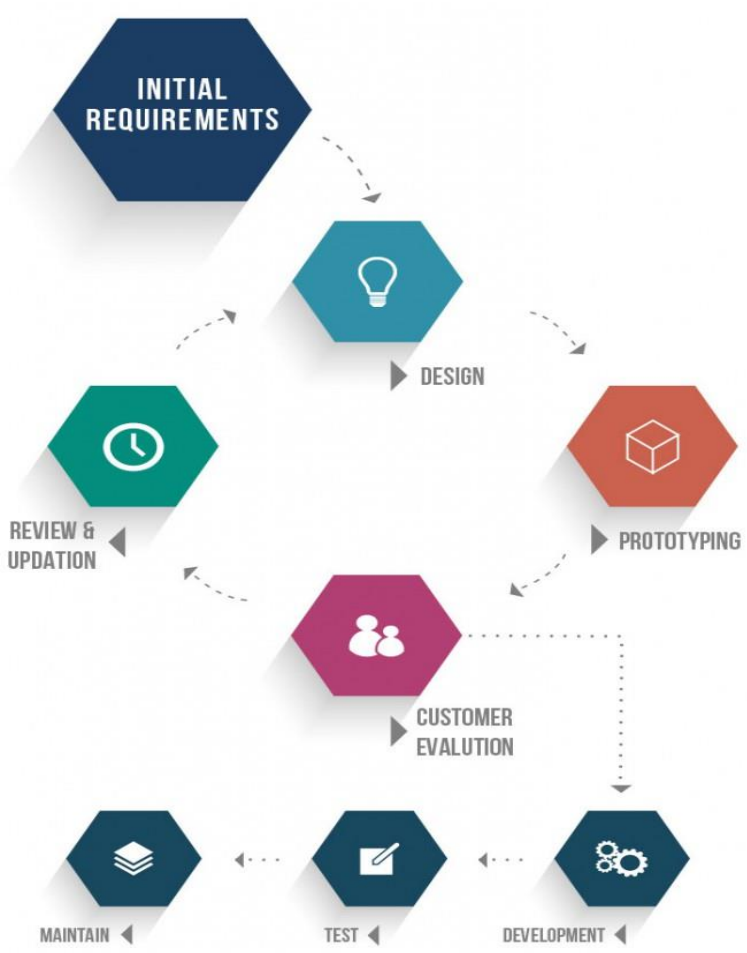


Figure 3. 1: Diagram of the Prototype methodology (Tavasoft, 2015)

3.2.1 Initial Requirements

This first step involves the understanding of the very fundamental products requirements that the product will need to meet user's needs. This comes is particularly emphasized in terms of the user interface provided to users in software development. The more convoluted and unspecific the details and requirements are the more complex it will be for the developer to satisfy the customer's needs.

3.2.2 Design

This step involves brainstorming and designing the concepts that will show how the system will look like and the functions it will perform for its users. The user interface is keenly look at in this stage to meet user's needs, as well as the performance, architecture and security features.

3.2.3 Prototyping

The initial prototype is developed in this stage, where the basic requirement is showcased, and user interfaces are provided. The prototype however may not work as precedented by the user, rather the main idea is to have something, rather than nothing. The prototype is developed to give the user the same look and feel that the actual product/ software would provide to its user, giving the developer ample time to work towards the goal of a complete final product.

3.2.4 Customer Evaluation

Once a simple prototype is developed, then in this stage it is presented to the customer or user and other stakeholders in the project to evaluate if needs are met. The results from the evaluation are then collected for review and further enhancement of the prototype until a finished and working prototype is developed.

3.2.5 Reviewing and Updating

The feedback and results collected from the previous step go through review in this stage. Comments are discussed, and negotiations take place with the customer. The negotiations here are based on the time factors required to be met to meet the prototype and actual product's deadlines, the project's budget, and feasibility in terms of the technicality of the actual implementation. The amendments are then made in a new prototype developed and the process repeats until the user or customer is satisfied.

3.3 Analysis

Software Requirements analysis is the tasks, activities, and procedures that software engineers or developers go through in order to determine the conditions and needs to be met for the development of a new or existing product that needs modification and alteration in order to meet user's needs. The process considers all the needs of the various stakeholders including beneficiaries of the project (McClintock, n.d.)

In this project, the object-oriented analysis and design will be used. This is because, not only does the approach improve the quality and productivity of system analysis and design making by making the components more reusable as compared to the structured approach but it also performs well in a situation where systems are undergoing continuous design, adaption, and maintenance. It classifies the problem domain in terms of data and behavior hence the problem domain's objects are identified.

3.3.1 Functional Requirements

Functional requirements are the requirements a customer or user fundamentally requires from software or system, to provide them with the ability to facilitate their needs (Berg, 2012). They are represented in the form of input to be given to the system, the operations performed, and the output expected. The information password management system will:

Specifically focus a targeted environment, local enough to the user to resist data theft. Even if the host's device is breached, prompts to alarm the user will be placed and user will also be advised to keep the master password safe, away from the rest of their other confidential information.

Users can find or experience password problems especially with login. Users hence would be provided with a self-service system, in order for them to resolve their own personal experienced problems with login without having the need to call the IT help desk for password resets.

Users will be asked to meet the login requirement of having a long, unique and complex master password when creating their accounts to ensure optimum and complete security.

Authorized users who fail login tryouts due to incorrect inputs will be blocked from accessing the completely until the necessary procedures and areas in the self-service section are complete.

3.3.2 Non-Functional Requirements

Non-functional requirements are requirements that specify the constraints that can be used to make a judgment of the operations of a system, rather than specific behaviors. The basic non-functional requirements in this project will deal with issues concerning:

The Deployment of the system: The system will primarily require a client-side device in order to be installed by the user. It will mainly be a self- contained system for swift access.

Optimum security efficiency: Due to the sensitive fragments of the I.T. infrastructure required, the management system will be able to securely manage and help the users control their password information that will allow them to access to confidential user profile data on their internet accounts.

Administration costs involved: The ongoing cost and effort required in contacting help desks to retrieve the password would be abated as well through the self-service system.

Usability of the system will be made as efficient and as comprehensive as possible.

3.4 Design

System design is the stage in development where the gap between the problem domain and the existing system is connected in a more coordinated way. Here the models, data, and interfaces are used to design a product that will satisfy the user's needs.

The Object-Oriented design approach will be employed to collect the object requirements for the system. The project will adopt a use case and flow diagram to exhibit the interaction made between the user and the system. Class relation diagrams will be used to show the interaction of objects defined in the system. A design class diagram will be used to exhibit the combination, splitting, and elimination of classes. Entity Relationship Diagrams will be drawn to give and illustrate the relationship between people, objects, places, concepts, and events that occur in the system.

3.5 System Development Tools and Techniques

The C# language will be used to build the desktop app in this project. This is because C# is mainly used to develop desktop applications much easier for Windows Operating Systems environment. It also has faster run times than most languages like python.

In this project I will be using the Visual Studio Community IDE since it offers fully-featured tools and kits for students and individual developers. It also offers its developers with a live share feature for sharing developer code. It's also the best environment for developers to build programs using C#.

The .Net framework will be the framework used for developing this project. This is because it offers memory management and caching systems that are robust and easy to use. It is based on an object-oriented programming principle where the idea is split down the software into smaller bits that are much easier to manage and combine.

The Database Management system (DBMS) software that I will seek to employ on my project will be the MySQL database. This is because it is open-source software that uses the freemium model and very easy to use as well.

3.6 Testing Method for the developed system

For the software testing method, I will be applying Black Box Testing. I will be using this method to make analyses from users' feedback on bugs they found concerning the system and improve more of the user's experience while using the app. The main objective of the testing will be to check the functionality of the modules in the system under test. It is also less exhaustive to conduct a black box test than a white box test.

The external or end-user perspective on the software is key to the success of this system and by conducting black-box testing by implementing and systems test and acceptance test, then the overall goal of allowing users to feel comfortable with the password manager will be achieved.

3.7 Domain of Execution

The domain of execution for this project will be based on desktop and web based. The software is based on the idea of giving users control over their data by offering users a decentralized database where the database is split into parts and distributed to different nodes for storage and use. Hence the creation of a desktop app for the system is necessary due to the hardware interaction specifications required by the system. The system will require a non-internet environment to completely secure the user's data. Also, because the solution requires a significant amount of RAM and other hardware resources. I'm more familiar as well with desktop applications more than other platforms due to the language I'll be using during development.

The development of a web-based domain of execution will be created in the form of a browser extension. The browser extension will be used to connect data and features from the desktop application where all data will be stored from the user.

3.8 Proposed modules and system architecture

The system will use decentralization database system architecture to store user's information on their local devices where the data is distributed on the database to different nodes for storage. Incase users feel the need to change their master password on the system due to the high complexity of the password, a password change tool will be placed to help them do so.

The use of search to find their information on their login details will be utilized. The search will also be a tool to optimize users' experience by lessening the time they use to search for their data.

A two-factor authenticator will be added to help users add an extra layer of security to their login information. This will be done through the use of a time-based one-time password tool that will add an extra layer of security for its users.

The use of a password generator will be introduced into the system, but the integration of an Emoticon generator for increased password security for the users.

Chapter 4: System Analysis and Design Description

4.1 Introduction

This chapter illustrates the preparation process in designing and modeling the proposed system, by illustrating the functionalities of the system and how the various components of the system will be merged. It defines the data collected from research and shows how the data will be used to shape the design and architecture of the system. It will cover the direct and indirect relations that will occur inside the system and how these interactions shape user experience.

4.2 Requirement Gathering

This section involved identifying and documenting the necessary requirements that users or any other stakeholder would have like to see for the system developed for their use. It involved being able to seek requirements that would favorably satisfy a non-functional or functional requirement. This essential process paramount to the understanding and fulfilment of users' needs involved requirement gathering techniques such as interviews, document analysis, and observation.

Through the use of interviews, different personal stakeholder perspectives easily understood for the study, a major and crucial technique that resulted in supplying essential data to the project. The process usually began with open ended questions for example like: 'How do you feel about the way you store and manage your passwords today?'. Most respondents who answered this question gave answers that mostly implied that passwords were routinely forgotten or misplaced by the person. The process later drilled down to detailed questions or higher-level questions that required more specifics like if they would use password managers that implemented the use emoticons in passwords as opposed to generalized questions.

Reviewing the documentation of existing systems helped create the necessary components for the system and finding the necessary gaps that I might have missed during the interview process. In an ideal world, reviewing the requirements helps drive the creation of the existing system – a starting point for documenting current requirements. In this solution, a review of past system managers assisted in providing the essentialities that users would feel had substantial value before to them as opposed to overlooking those necessary requirements essential to the final product. Nuggets of information are often buried in existing documents that help us ask questions as part of validating requirement completeness(Tutorials point, n.d.).

During the observation of users, process flow could be identified, steps, pain points and opportunities for improvement of previous solutions. In this study observations were both passive and active intermitting between asking questions while making observations among users. Passive observation was much better for getting feedback on a prototype to refine requirements, where active observation was more effective at getting an understanding of an existing business process. Either approach however was essential to the requirement gathering of the system.

4.3 System Requirement

This section defines the requirements at the system level that describe the functions which the system as a whole should fulfill to satisfy stakeholder needs and requirements and are expressed as functional and non-functional requirements.

4.3.1 Functional Requirements

ID	Descriptions
FR 1	The system should allow users to create accounts
FR 2	The system should allow users to login
FR 3	It should users to generate emoticon passwords
FR 4	Should allow users to add a new website account
FR 5	Should allow users to view their login information
FR 6	It should rename a site name
FR 7	The system should allow users to edit passwords
FR 8	The system should allow users to save/edit card info
FR 9	The system should allow users to reset their pass info
FR 10	It should allow download reports.

Table 4. 1: Functional Requirements

4.3.2 Non-Functional Requirements

ID	Descriptions
NFR 1	The optimum security to its users by grating access only to registered users.
NFRQ 2	The system should provide optimum performance
NFRQ 3	It should have easy design for easy comprehension
NFRQ 4	It should allow users of windows OS to efficiently use
NFRQ 5	Should grant edit privileges only to the owner

Table 4. 2: Non-functional Requirements

4.4 System Architecture

The system was designed based on the P2P architecture that works best when there are lots of active peers in an active network, so new peers joining the network can easily find other peers to connect to. The system's prototype however does not yet require such networks for the time being however the concept of optimum security will need the implementation of this architecture in order to be aptly achieved. If a large number of peers drop out of the systems network, there will be still enough remaining peers to pick up the momentum and maintain security efficiently. If there are only a few peers, there are less resources available in optimum performance and security. For example, in a P2P file-sharing application, the more popular a file is, which means that lots of peers are sharing the file, the faster it can be downloaded. P2P works best if the workload is split into small chunks that can be reassembled later (Peer2Peer, n.d.). This way, a large number of peers can work simultaneously on one task and each peer has less work to do.

In using P2P networks for the system, no central server would be needed to maintain and to pay for (disregarding tracking servers), providing economic value to users (Tiwana, 2014). That also means there is no need for a network operating system, thus lowering cost even further. Another advantage would be there would be no single point of failure, unless in the very unlikely case that the network is very small. P2P networks are very resilient to the change in peers; if one peer leaves, there is minimal impact on the overall network. If a large group of

peers join the network at once, the network can handle the increased load easily. Due to its decentralized nature, the P2P networks will survive attacks fairly well since there is no centralized server.

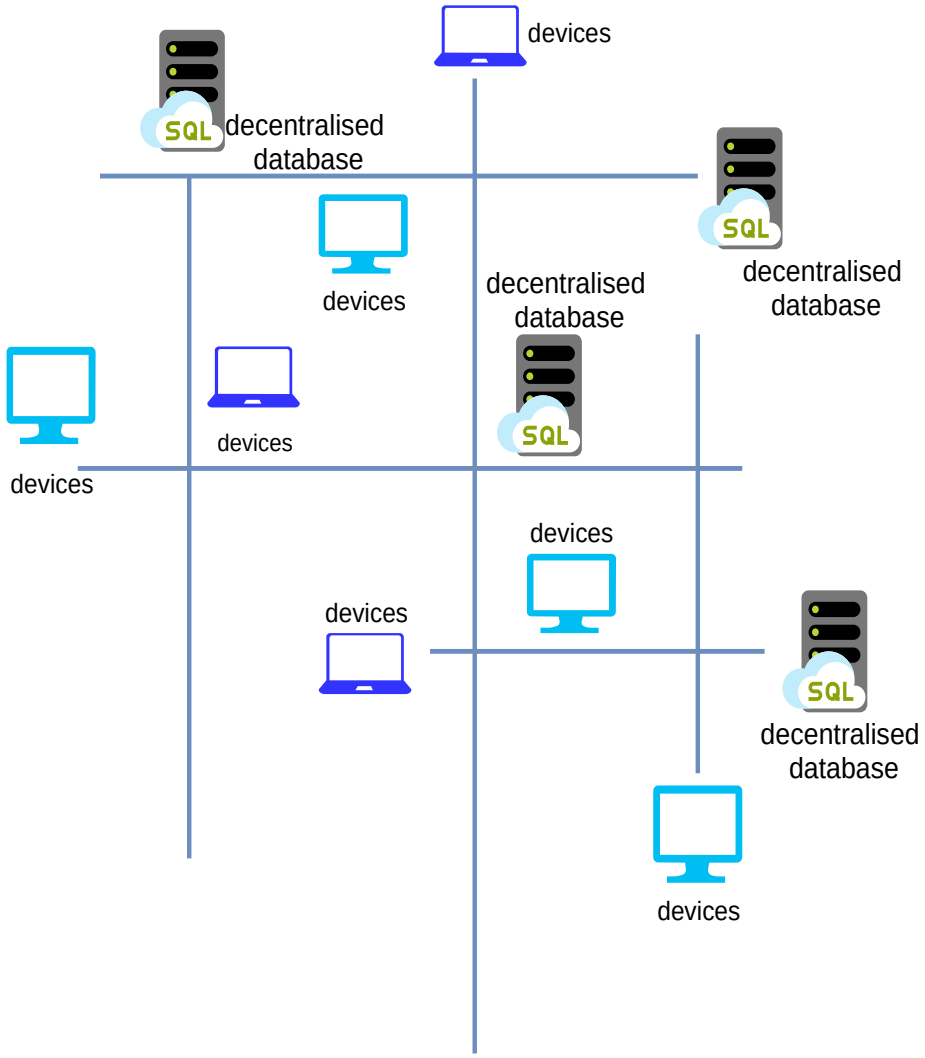


Figure 4. 1: System Architecture

4.5 System Analysis

System analysis represents the dependence of hazardous events on lesser, more basic events of the system (Rushton, 1998). The illustration below demonstrates the aim to aid communication of analysis between a system analyst and decision maker, supporting the discussion of how the representation how a hazard can take form as analysis ensues.

4.5.1 Use Case Diagram

The use case diagram illustrates the specific interactions that take place between the user and the password management system. The user who is the account holder can manipulate different components and levels in the system like access user login credentials or a generate passwords.

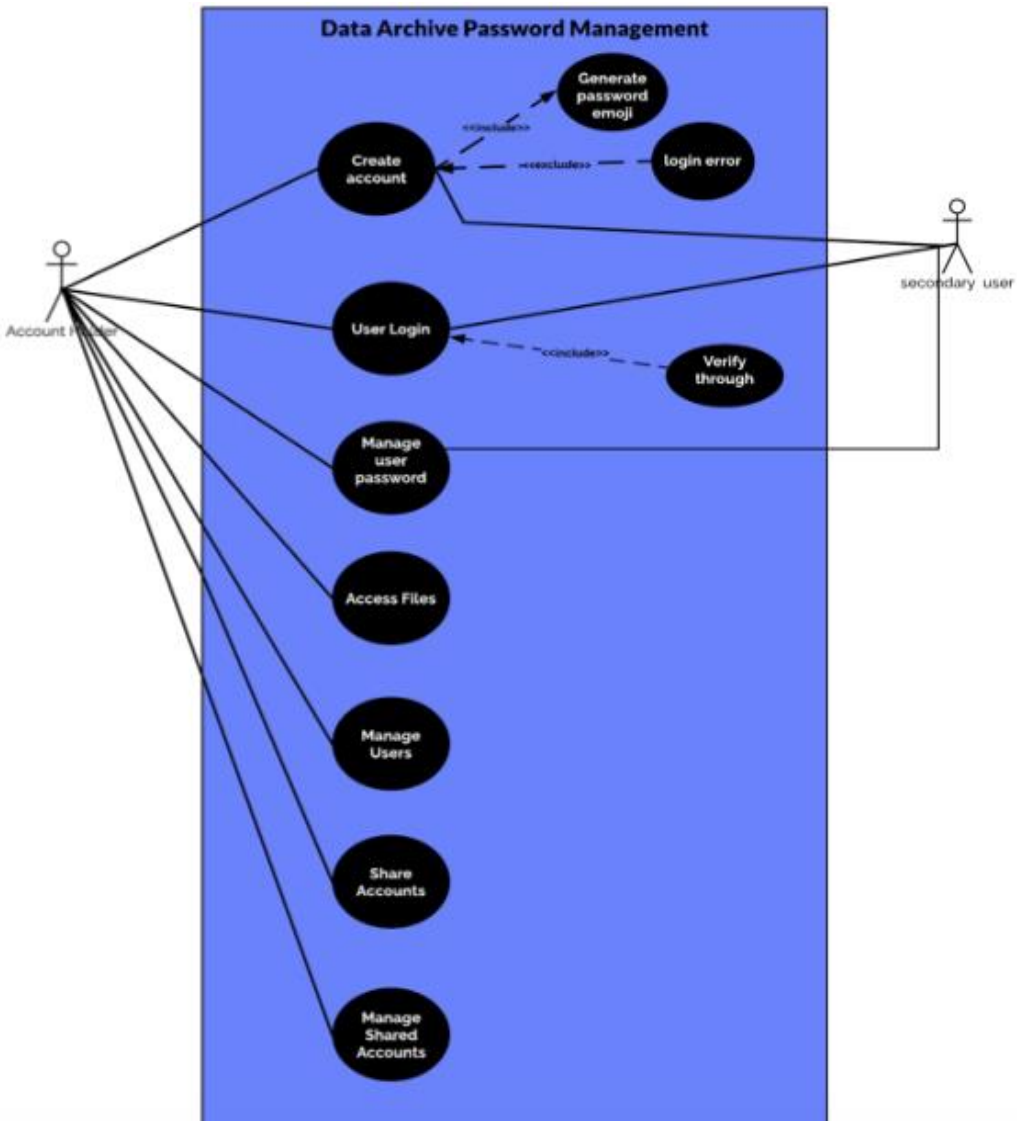


Figure 4. 2: The Use Case Diagram of the System

4.6 System Design

System design involves modelling and demonstrating components to assist in the visualization of the solution. The system diagrams illustrate the optical models of a system's components and their interactions. Below demonstrates the visual models of the password management system and the interactions occurring its environment i.e., with the actor.

4.6.1 Sequence Diagram

The sequence diagrams displays the interactions involved and the operations carried out in the password manager. It illustrates how and in what linear order the system will interact by using its vertical axis to represent where what messages are sent and where.

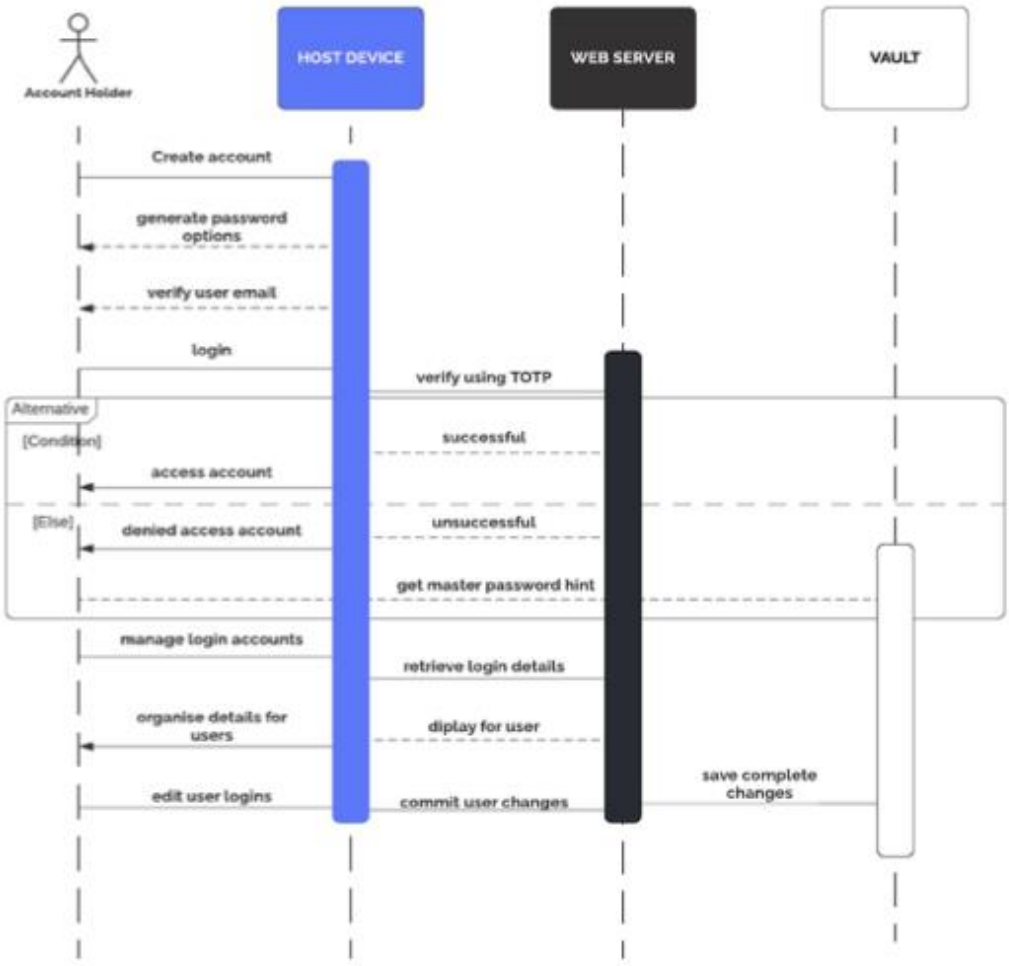


Figure 4. 3: The Sequence Diagram of the System

4.6.2 Class Diagram

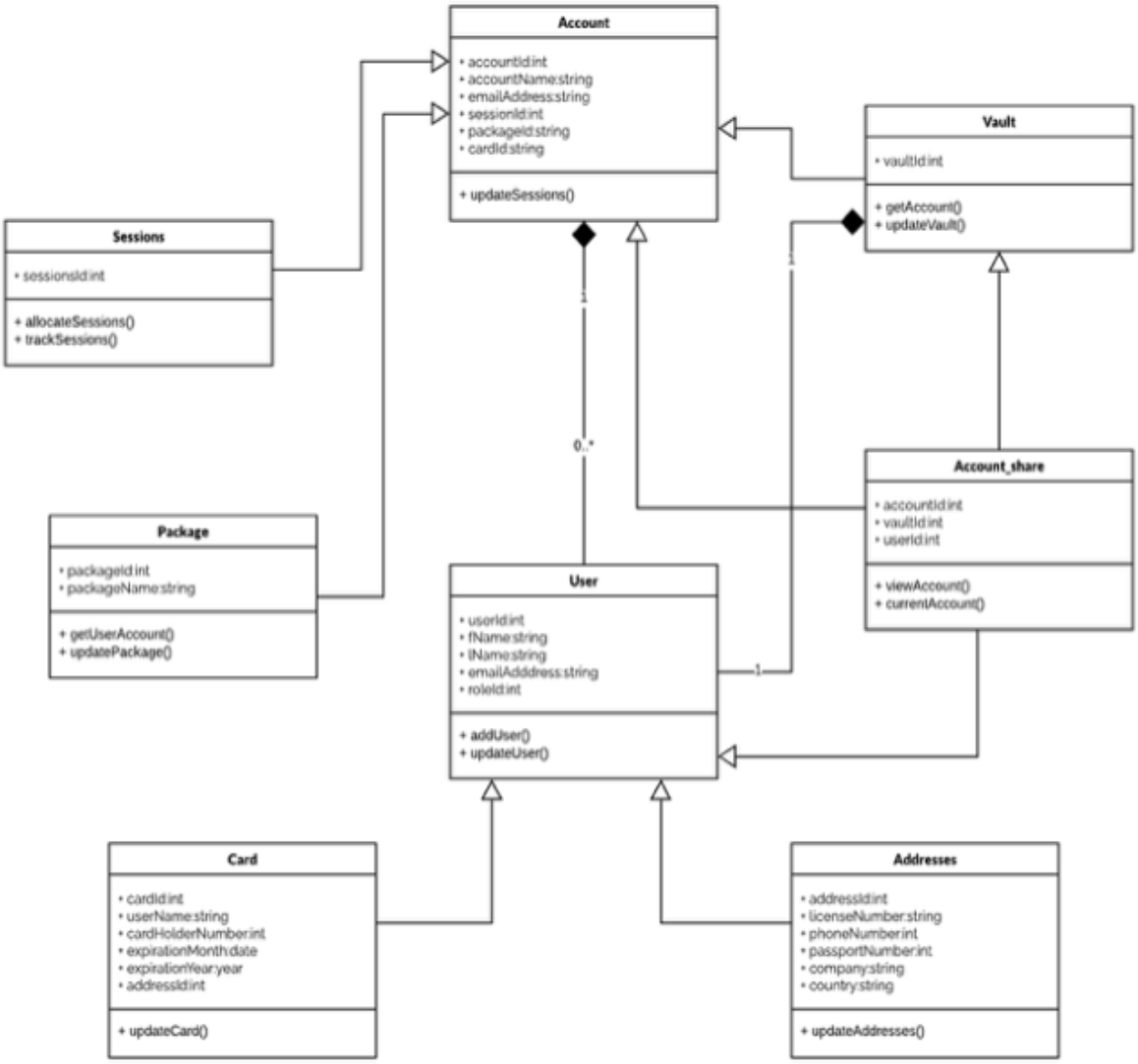


Figure 4. 4: The Class Diagram of the System

The class diagram above depicts how different object present in the system, attributes that users interact and the operations performed on this attributes, and finally the relationships that the attributes and operations from other classes have with each other.

4.6.3 Entity Relationship Model

The figure 4.5 below illustrates the conceptual data model of the relationship between different database entities in the system, depicting the logical nature and construction of the database, commonly interpreting to the actual database's structure.

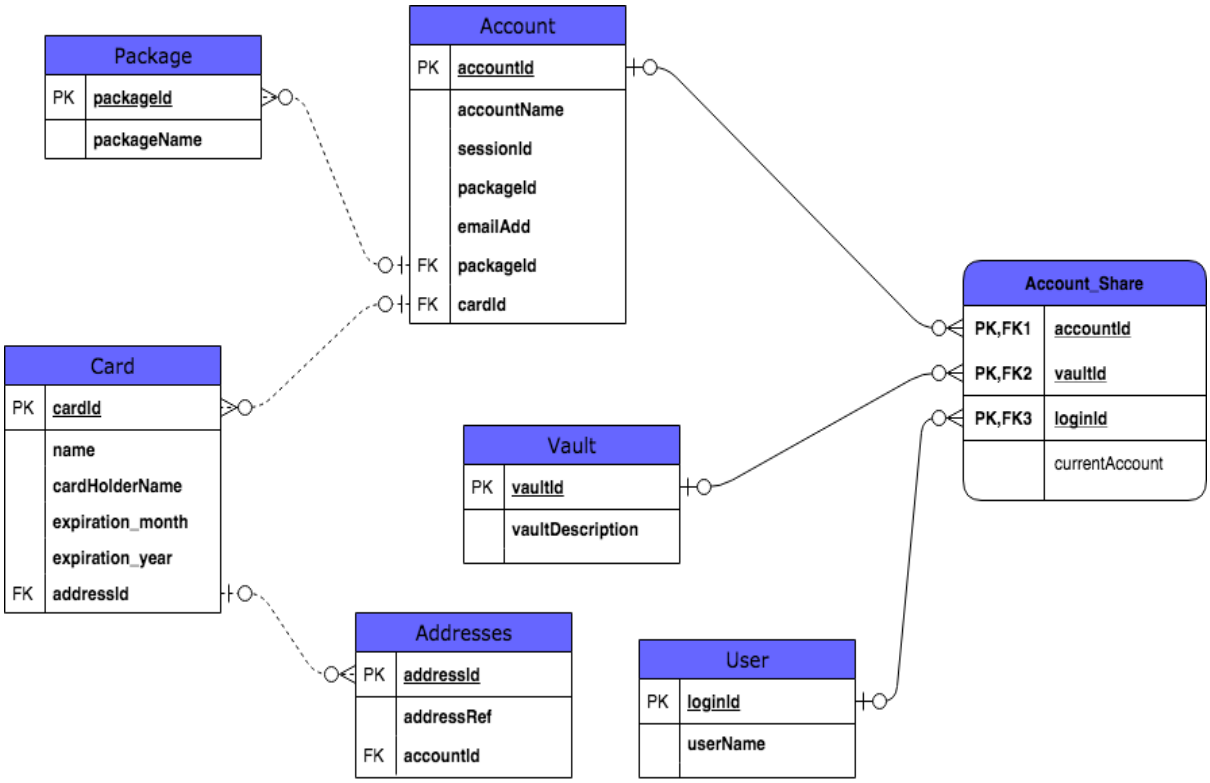


Figure 4. 5: Entity Relationship of the System

4.6.4 Database Schema

The figure 4.6 below represents the logical configuration of a relational database, indicating how the password management system's entities relate to one another. It not only displays the interrelationship between entities but also depicts the constraints of data managed and stored inside the database.

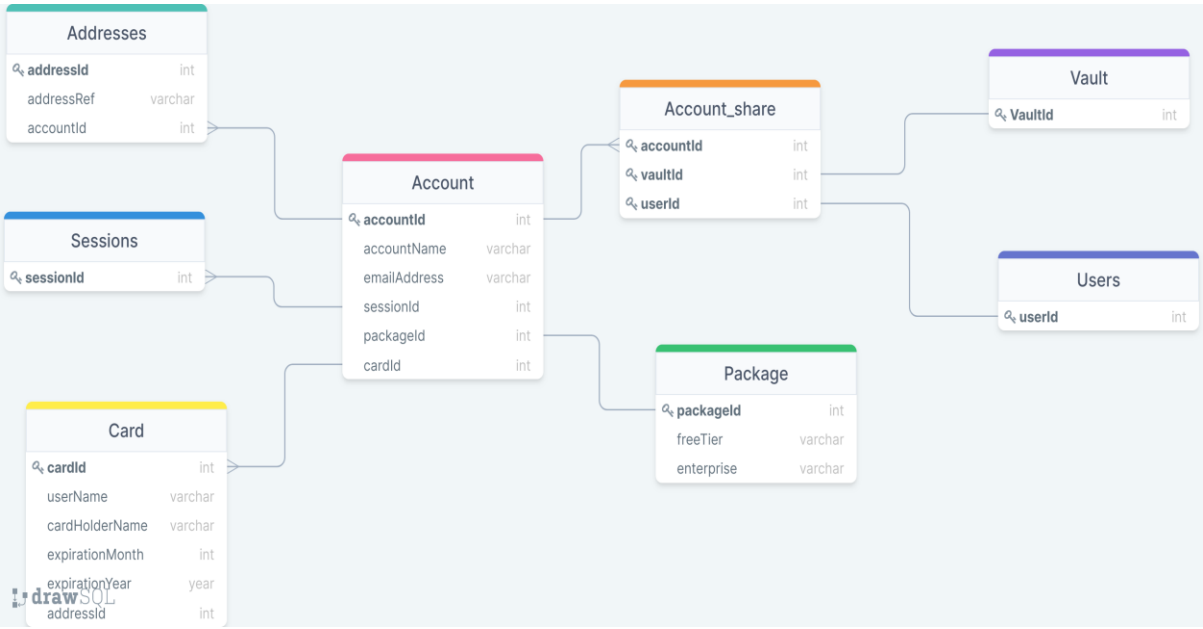


Figure 4. 6: The Database Schema of the System

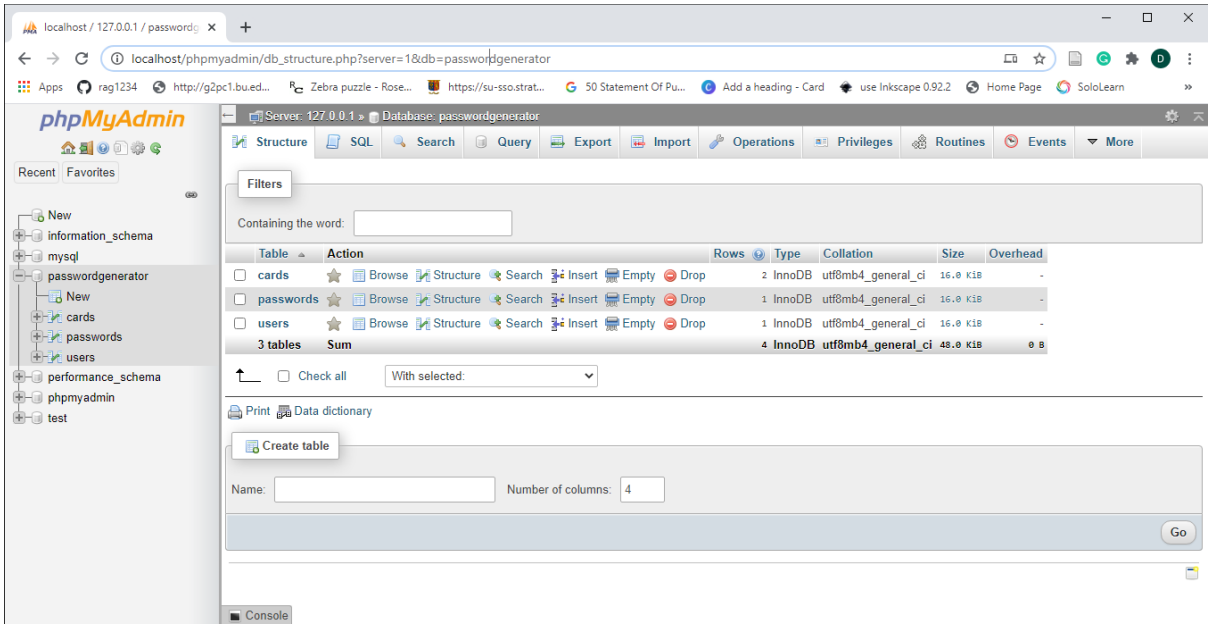


Figure 4. 7: The Entire System's Database

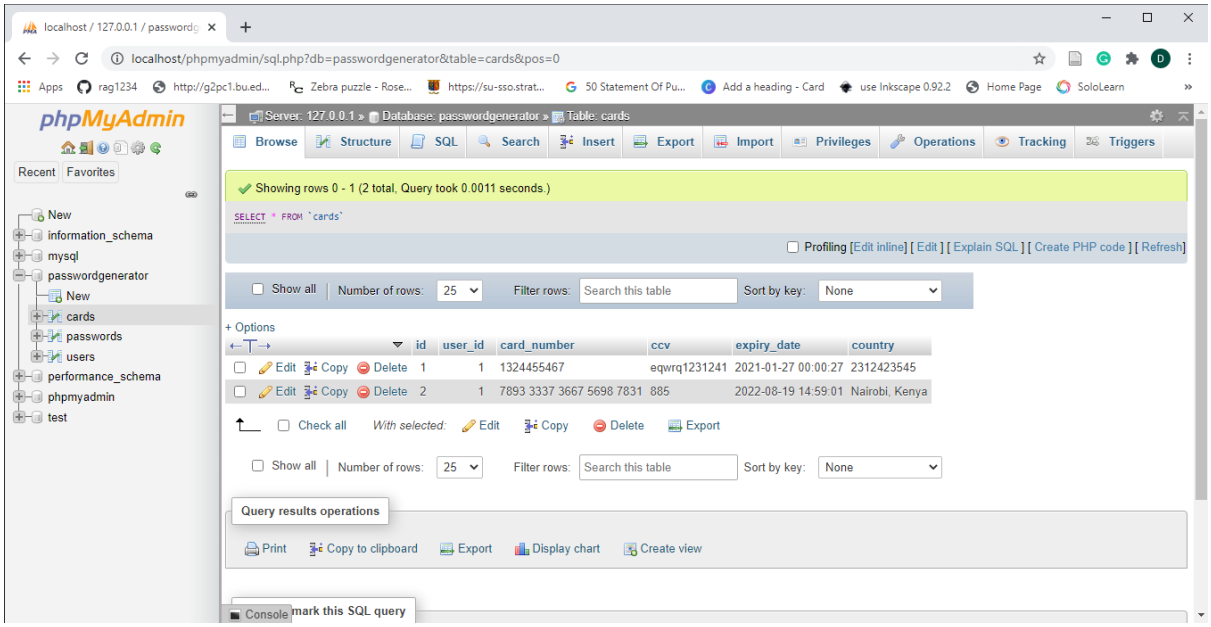


Figure 4. 8: Card Information Table

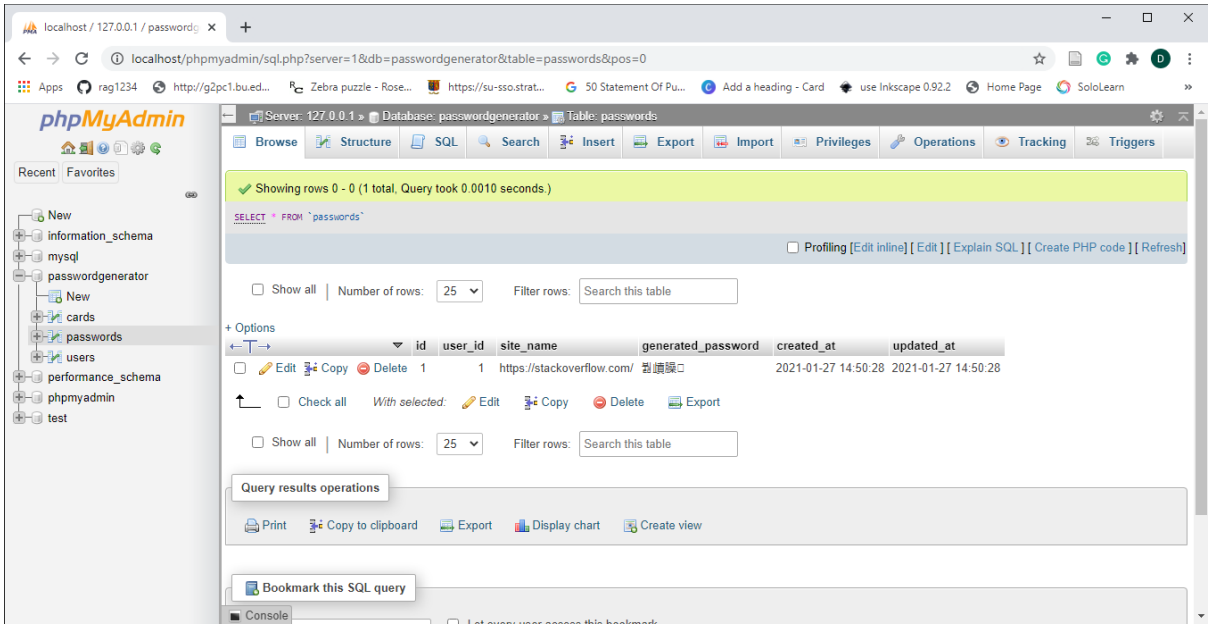


Figure 4. 9: Password Information Table

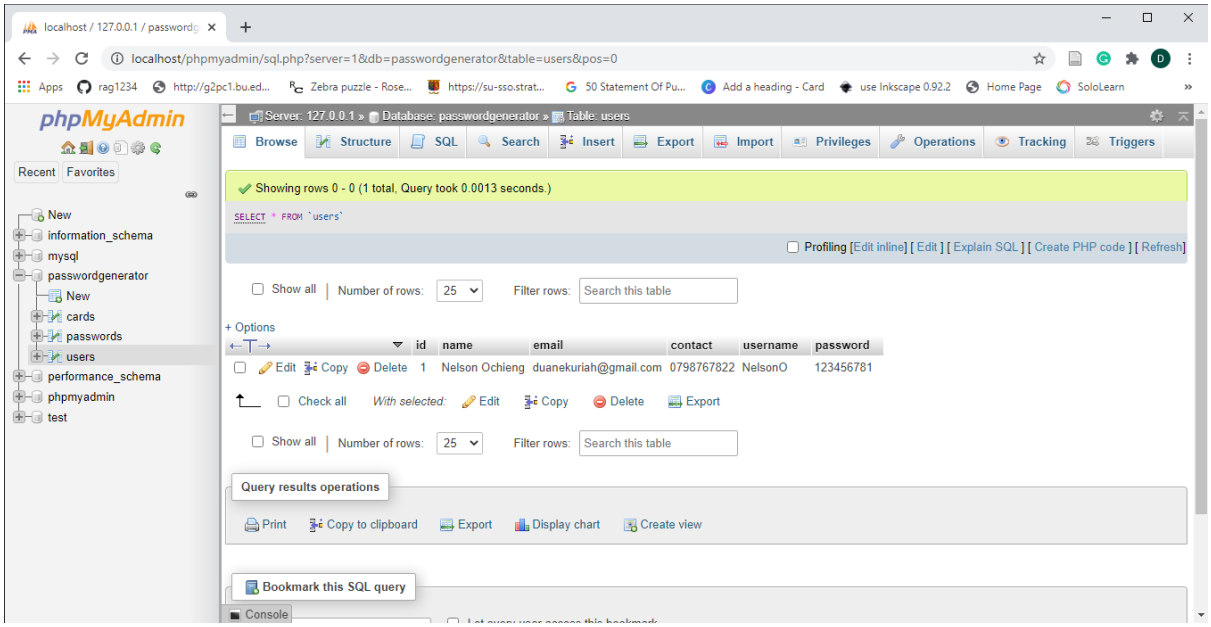


Figure 4. 10: User Information Table

4.6.5 Mockup Design

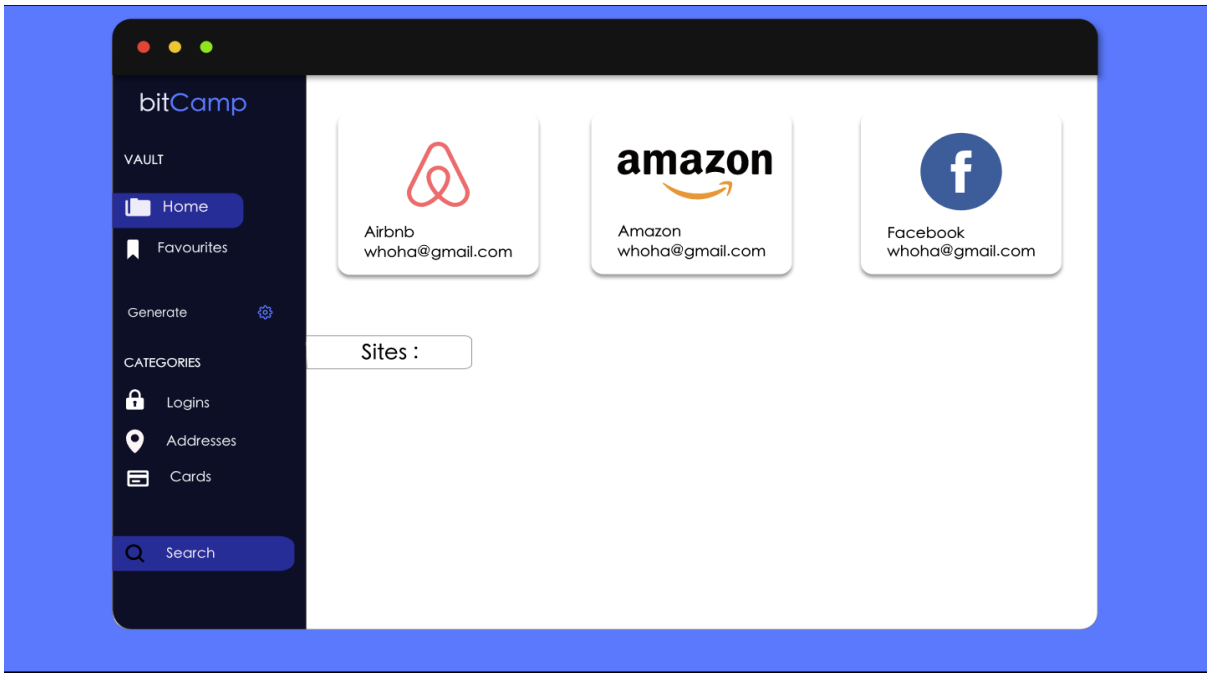


Figure 4. 11: The Mockup Design of the System

The above figure illustrates the mockup of the desktop application, indicating the various tabs or buttons available for selection and for storage of data. It is currently displaying the vault section of the system and is currently on home page. The home page is displaying the websites that the user may have saved, each with its own selection of data stored inside. The user can also navigate to their favorite websites most frequently visited, displayed on the mockup. They could click on the generate text block to generate a random array of characters and emoticons for use. Other buttons include the logins, addresses and card information button, each with its agenda of safely storing user data on the database.

Chapter 5: System Implementation Testing

5.1 Introduction

The chapter will cover the implementation testing stage of the project. It will cover the procedures conducted in the installation of key components used to develop the password management system. It will present the different factors that were used to developed and facilitate user functionalities in the system. It will broadly describe how the system was developed and the testing processes that took place to verify that the system satisfies user needs and all other necessary functional requirements.

5.2 System Implementation

The system was built using the C# language on visual Studio 19, where the .NET framework primarily considered a key Windows Operating System Component, was the main component used to the run the application (Adegeo, 2020). Windows Forms allows users to run applications not as web browser but as a web application locally. This approach, during the system's development was effective for data security and easy accessibility (Design, n.d.) as compared to the conventional online application because data decentralization is one of the core objectives the project seeks to accomplish and Windows form.

With Windows Forms, apps that are graphically rich and easy to deploy, update and work on while offline can be developed while still connecting the user to the internet. Windows Forms apps will allow users to access data safely and effectively from their local hardware and file system systems of the computer as the app is running.

5.2.1 System Specifications

In order to realize this project, the following software and hardware components are used:

i. The processor should be an Intel ® core™ i7-2670QM
ii. The processor should run at a speed at least 2.20 GHz
iii. The installed memory is 4 GB RAM
iv. The system type should be a 64-bit operating system with a 64-based processor.

The following components are used to set the android application:

i. The operating system version is; Window 8, 10
ii. The processor should be; Quad-Core Processor
iii. The internal memory should be; 8GB
iv. The RAM module should be; 2GB

5.2.2 Network Specifications

i. The bandwidth should be greater than 50Gbps.

ii. The device must be internet enabled.

5.2.3 System Components

5.2.3.1 Registration and Login

The system uses the database to authenticate users and does not use or require an administrator system to authenticate users. The system will the database to register and save users data for the login process. Once the users have been saved, system will notify the user of their successful registration and allow users to login using the information they registered with.

Users will be required to use a length and easily identifiable password, easy to remember during the registration process. This password will be recognized as the master password, similar to a master key, and will be used primary to access the user's confidential data, stored within the system's database.

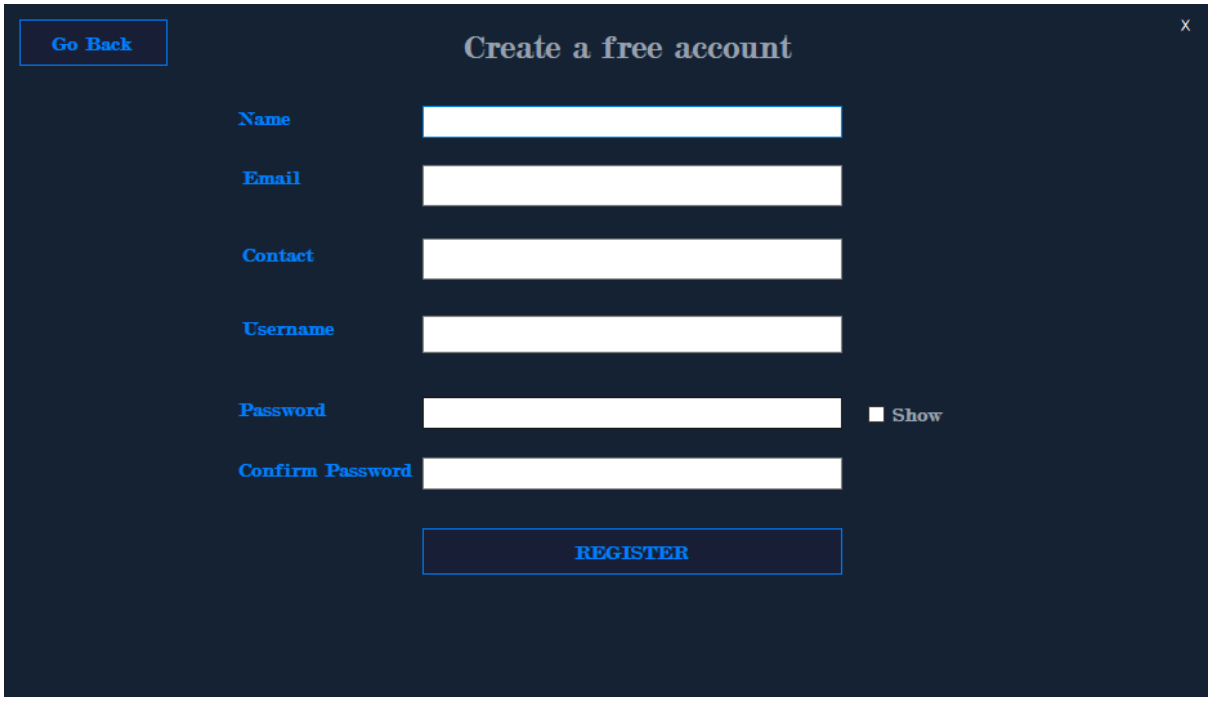


Figure 5. 1: Registration Page

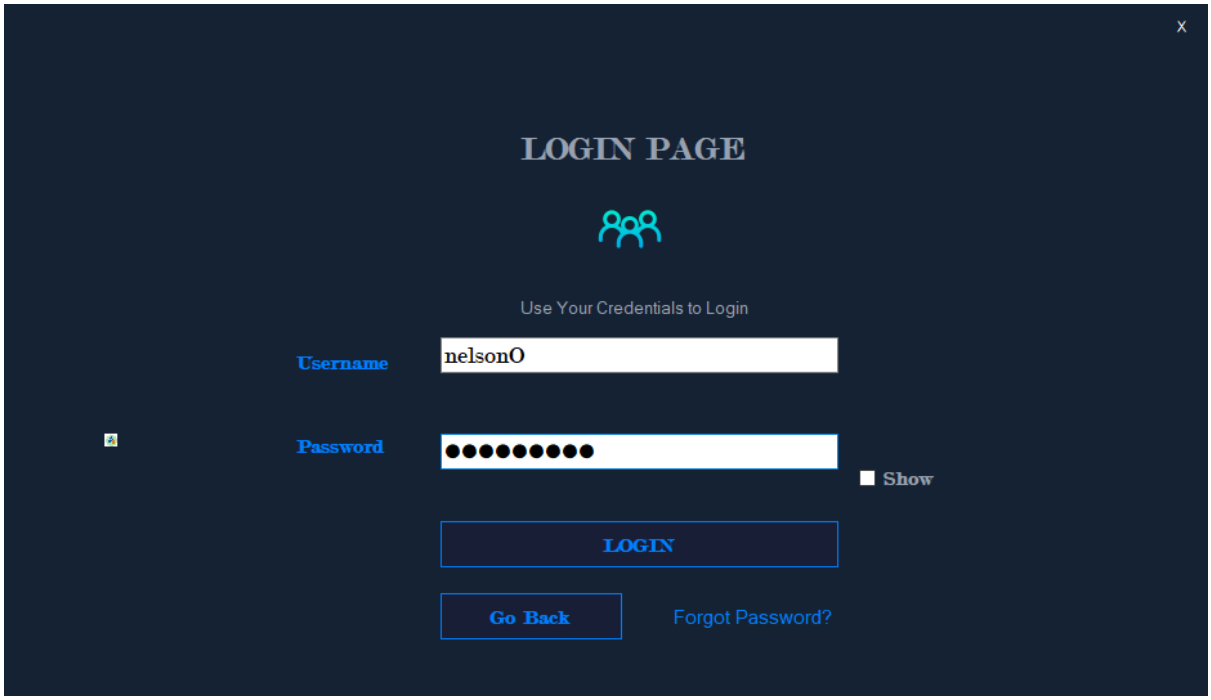


Figure 5. 2: Login Page

5.2.3.1 The Main Dashboard

The dashboard of the system displays the necessary components necessary to the user to store their information. Most of the information displayed is stored on the database. The different modules present are Add new site module used to stored or input new login information data, the favorite module used to edit and view all the logins the user stored and the card module for card information and data the user may see relevant in storing in the system.

Figure 5. 3: The System's Main Dashboard

5.2.4.1 Add New site

When the clicks on the add new site, users will be ushered to a page where they can easily store their login credentials form a specific website and can randomly generate a password that is a blend of emoticon symbols, characters and numbers. The passwords have been randomized to be effective and adequate for security measure and protocols on the internet.

Once these details have been added on the website, users can save and edit their passwords and site names on a different page to prevent confusion and make the website easy to use.

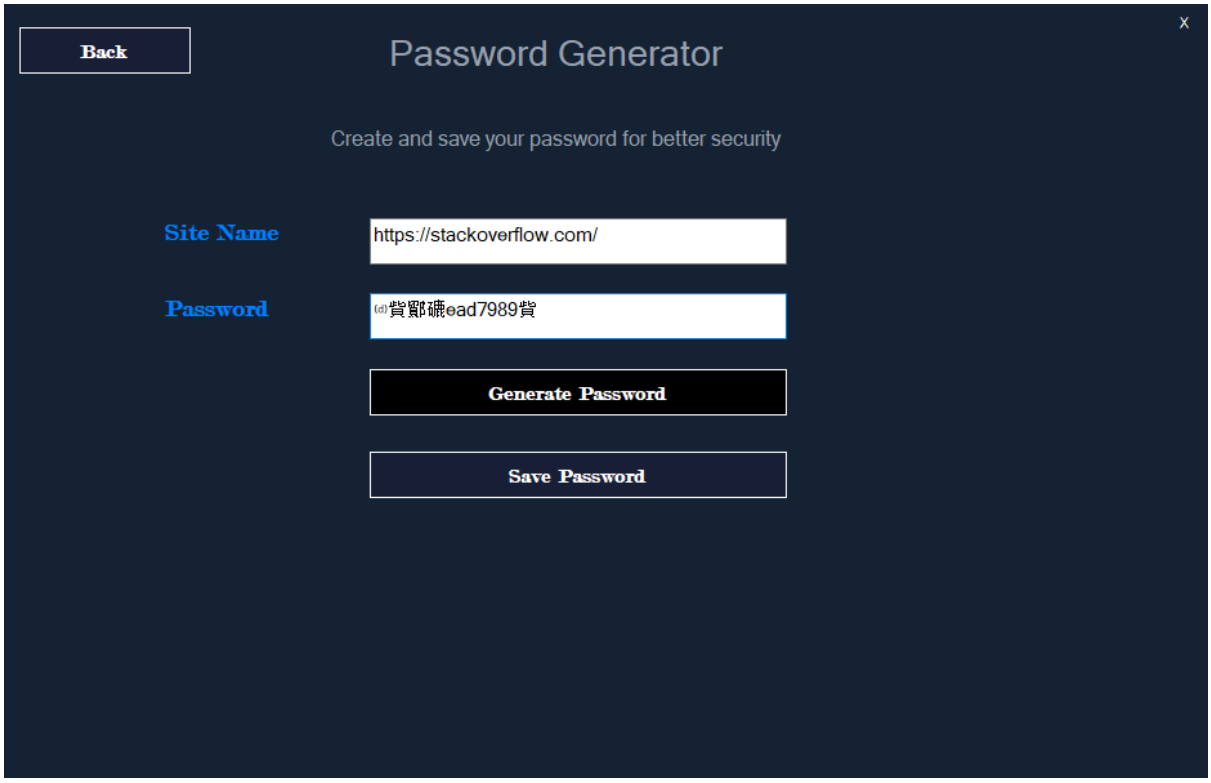


Figure 5. 4: The user interface module for adding new website information

5.2.5.1 Manage website and password information.

The users can view all the websites data and information and password after clicking on the favorites button as displayed below. The user will have their site name displayed, password information, the date the website was created and time and when it was last updated.

This information can be edited by the user if need be and deleted from the database by clearing the section the needs the deleting.

A search bar also exists at the top of the website in case the user requires searching for new his previously saved passwords.

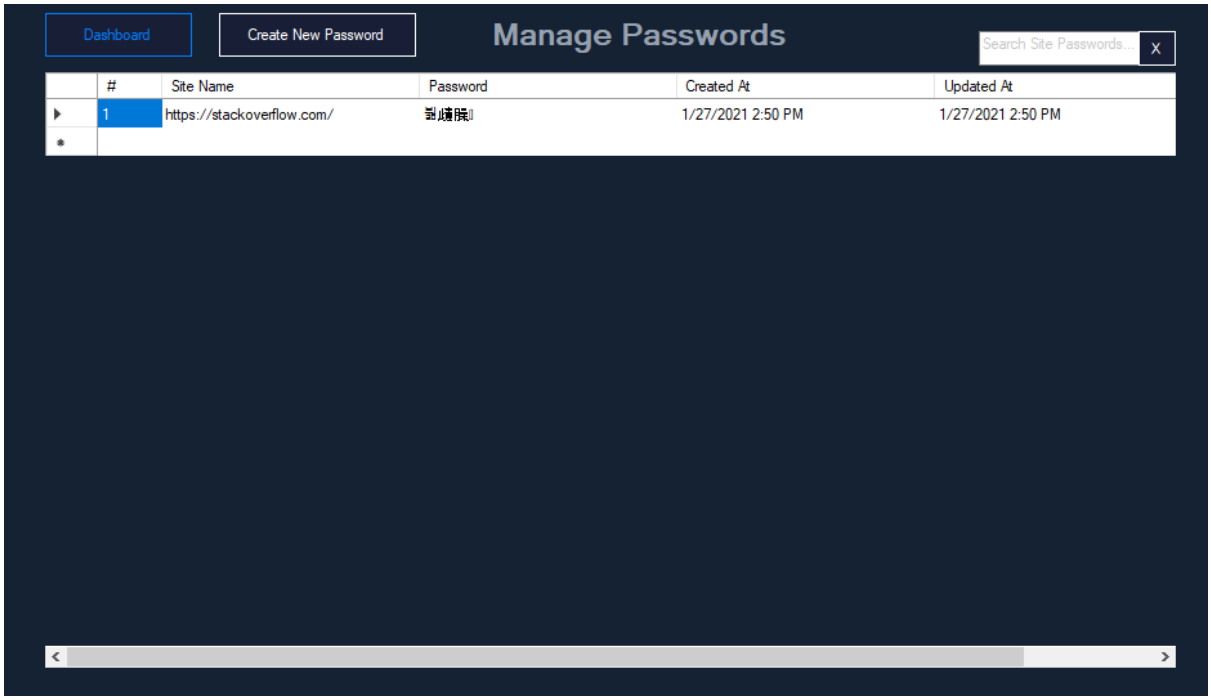


Figure 5. 5: A Display of websites saved by the users

5.2.6.1 Add Card Information

An additional module has been developed in the system to allow users to add any card information they may and that they consider safely keeping for easy retrieval and access. The add card interface displays a text box for users to input their Card Number, their Credit Card Verification numbers (CCV), Card Expiry numbers and information on the country they received their cards from.

When the Add Card button is clicked by the user, their card information will be saved on the Card module of the system or users can use the Back button to view their can information.

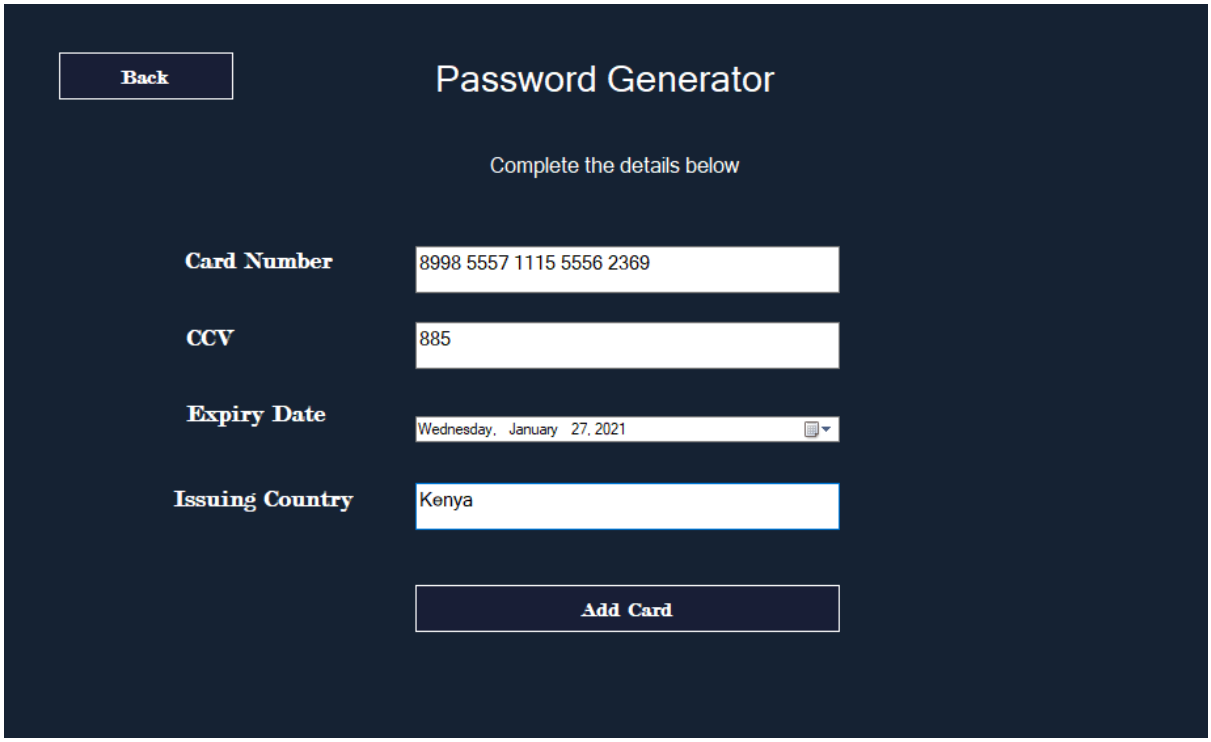


Figure 5. 6: Add User Card Information

5.2.7.1 Manage Card Information

The manage card interface displayed when users click on the Card button of the system, users will be able to view their card information directly project from the system database. In this module users are automatically assigned an id number and provided with their details concerning the card numbers they saved, their CVV numbers, expiry dates and the country of issue they previously stored.

Users can delete or edit their information in this section reflecting the results of the recent updates onto the system's database.

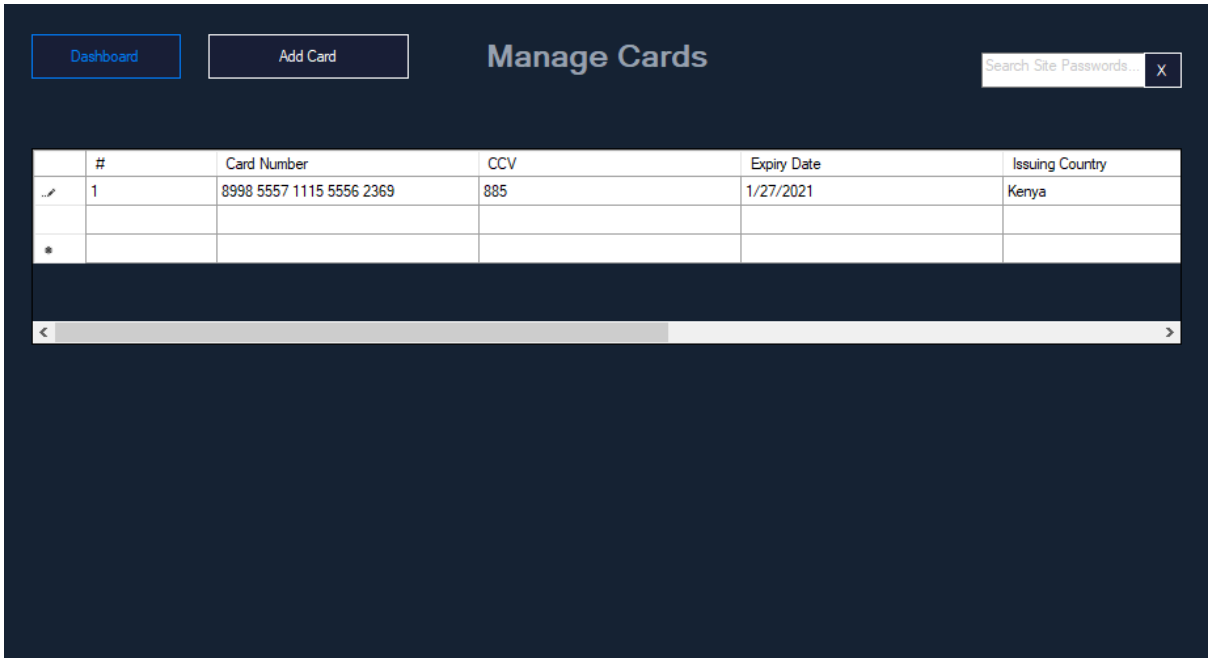


Figure 5. 7: A Display of Card Information Saved by the Users

5.3 System Testing

After the integration of modules, the whole system was tested. The system was tested using a local host and provided auspicious results. Sample details were input into the system and it were stored in the database successfully.

5.3.1 Unit testing

Each unit of the system were individually tested independently to ensure that each unit or component was working as initially intended. The system's code and infrastructure were tested and to ensure that IDE components were effectively working, and no bugs or glitches were

detected. Each module was tested to see how they would work alongside the system's database for example to observe how the login and the sign-up modules integrate with the database or if users could view the details they saved.

5.3.2 Functionality Testing

To ensure the specified requirements were working white box testing was carried out on the system. The system was tested at every development stage to ensure the working prototype was effectively storing and securing user's passwords and their details. The application also received collaborative support in testing from other designers and developer who would understand and provide relevant feedback to improve the functionality of the system and effectiveness of the system.

5.3.3 Usability Testing

The system underwent testing to determine feasibility and ease of use to improve user experience. This test was based on how easy and efficient users will find the application to be. Some of the identified criteria that were used to test usability include:

The portability of the system. Because the system did not offer users the option of using the application on the phones, users found this to be hindering and a bit inconvenient due to the obstructing deployment method used.

Ease of use of the system. This was based on how easy it was for people to use the application with minimal training. Sampled users found the application easy to use and remained consistent in finding the different modules friendly. The application has been implemented using an approach the minimalized on the different user interface components and an easy page transition approach.

Distractibility of the system. This was performed to determine the hinderances and distractions users would find in the system and take note of the observances to make the password management system as straight-forward in design as possible.

5.4 Test Cases

Test ID	Related requirement	Inspection check	Pre-condition	Test data	Priority level

M1	The system should begin with a dashboard page containing various modules	Does the system display the available modules?	There should be internet connection	If the link loads the login page or bring a connection error	High
M2	The system should validate User registration details	Does the system validate user details?	The details of the user should be available in the database	If details uploaded are posted to the database	Medium
M3	The system should be able to allow the user to input area details and location favorable to supply the products	Does the system allow and accept the details of the user?	There should be input to the system	Input posted to the system.	High
M4	The system should display on the reports of the system and allow them to be printed or downloaded	Is the system able to display the reports and download them if need be	There should be feedback of login entry report and card data report	If details can be validated	Medium
M5	The system should be able to show an output of card details and new website information	Does the system display the desired output?	There should be output from the system.	Output posted by the system	High

Table 5. 1: Test Cases

5.5 Test Results

Test ID	Expected result	Actual result	Status	Remarks
M1	The various modules should be displayed	They are displayed	Success	In case of a fail, the internet connection should be checked, otherwise it should function efficiently.
M2	There should a successful access to the profile depending on the validity of the details entered in the sign up	There is a successful login to the module of choice	Success	In case of a fail in the registration, check on the validity of the details submitted, otherwise it should successfully log in
M3	There should be a successful attempt allow the user to input information on the web sites in which they possess accounts	The system is able to allow and accept the details of the user	Success	In case of a fail of not accepting input posted to the system the system should be checked to verify otherwise it should function well
M4	The system should display reports and be able to successful download them	The system was able to display the system reports and download.	Success	In case of failure system code is validated otherwise it should display and downloaded
M5	There should be an output of card and website information added by the user.	There is a successful display of the desired output?	Success	Output is posted by the system

Table 5. 2: Test Results

Chapter 6: Conclusions, Recommendations and Testing

6.1 Introduction

This chapter aims to discussion and summarize the objectives mentioned in chapter 1 above and provide relevant analysis on the objectives achieved. It seeks to discuss the technical traits that aided the system to effectively work as was intended. Finally, it will seek to provide areas for further improvement what can be done in the future to improve its ability to solve user's needs.

6.2 Conclusion

The development of this system was primary aimed to solve the issue of users safely storing their data and personal information safely and for users to own their private data. It was aimed at improving older techniques of storing data on a centralized database or web server and locally store their own data safely effectively encrypted in the local devices. This was to remove the single point of entry hackers find when our data is stored in one place with other's data, each increasing vulnerability to safety of our own data.

In the course of achieving this task, a user could safely have their data stored and in their local database, encrypted from others. Users were also able to successfully view their saved data, edit or delete their account data. They were also able to successfully store their card information on the database and view their data at a separate location in the database. Users could easily navigate and transverse through the account and card information through the easy-to-use interface implemented.

The convenience of the local database allowed users to quickly and flexibly access their data from the local database effectively. Its existence also proves through its implementation, that it will be able to economically save user data reducing users' need for paying large sums subscription-based fees for password managers.

This project is set to offer the opportunity to internet user the ability to safely add, edit and store their confidential data and information for use on their web accounts that seem to be progressively increasing day by day. It will generate random length and complex passwords for the users, not only in alphanumeric characters but also in emoticon character, increase the

number of permutations or combinations available from 72 to 2100 character making it harder for intruders who seek to steal our information.

6.3 Recommendations

Internet users seeking to utilize the availability of password managers should consider using a long, complex and uniquely rememberable combination of characters, in order to effectively prevent intrusion into the system. The master password, used only during the login process of the application, should confer with the normal complexities required during the opening of website accounts.

Users should also try to remember their master password more mentally. When users tend to write their passwords down on a piece of paper, the piece of paper sometimes may end up in wrong hands and may provide a path of entry to intruders. Trying to maximize the remembering of passwords mentally, eliminates this potential path and maximizes the apps user security.

6.4 Future Works

Even though research was made, aimed to improve previous system techniques, more functionality is required to make the system as effective as possible. The system could be improved through use of a one-time password (OTP). This particular feature will allow users to easily their identity and prevent intrusion by hackers or infiltrators of data. Full disk encryption could also be implemented to provide a contingency measure in case the system is infiltrated. Full disk encryption completely encrypts and blocks unauthorized users from accessing the user's hard drive without the necessary login details required for gaining entry.

The password is only set once and cannot be changed until access to the hard drive is achieved. If the password or login details are lost the user cannot access the files inside the hard drive until the password is retrieved.

Another feature that would be useful and was not yet achieved in the project, is the use of a command-line tool, implemented to allow users to effectively retrieve their data using command-line. Password managers do not offer command-line tools for developers who constantly in use of terminal or windows shell to access file utilities that require passwords.

References

Adegeo. (2020, October 26). What is Windows Forms—Windows Forms .NET. Microsoft. https://docs.microsoft.com/en-us/dotnet/desktop/winforms/overview/

Derousseaux, T. (2017, November 21). Comparing 3 Password Managers: Google Smart Lock, LastPass and KeeWeb. Medium. https://medium.com/@tibdex/comparing-3-password-managers-google-smart-lock-lastpass-and-keeweb-f43cfefa8d4a

Design, G. (n.d.). Desktop Application vs. Web Application Design| GoProtoz Design Studio. Desktop Application vs. Web Application Design| GoProtoz Design Studio. Retrieved January 27, 2021, from https://www.goprotoz.com/insights/desktop-applications-vs-web-apps-design.html

encryption—How secure is Chrome storing a password? (n.d.). Information Security Stack Exchange. Retrieved May 20, 2020, from https://security.stackexchange.com/questions/170481/how-secure-is-chrome-storing-a-password

Fowler, G. (2019, February 19). Are password managers safe? A new report finds flaws in five. - The Washington Post. The Washington Post. https://www.washingtonpost.com/technology/2019/02/19/password-managers-have-security-flaw-you-should-still-use-one/

Hoffman, C. (2020, July 10). You Should Turn Off Autofill in Your Password Manager. https://www.howtogeek.com/338209/you-should-turn-off-autofill-in-your-password-manager/

McClintock, M. (n.d.). Systems Analysis & Design - Assignment: System Design Report. Study.Com. Retrieved January 22, 2021, from https://study.com/academy/lesson/systems-analysis-design-assignment-1-process-design.html

Now you can log into your bank using emoji. (2015, June 15). IeDigital. https://www.iedigital.com/news/now-you-can-log-into-your-bank-using-emoji/

Password manager vs remembering passwords. (n.d.). Information Security Stack Exchange. Retrieved May 20, 2020, from https://security.stackexchange.com/questions/3458/password-manager-vs-remembering-passwords

Peer2Peer.pdf. (n.d.). Retrieved January 24, 2021, from https://student.cs.uwaterloo.ca/~cs446/1171/Arch_Design_Activity/Peer2Peer.pdf

Requirement Gathering Techniques—Tutorialspoint. (n.d.). [Article]. Tutorials Point. Retrieved January 23, 2021, from https://www.tutorialspoint.com/business_analysis/business_analysis_requirement_gathering_techniques.htm

Rushton.G, A. (1998). System Analysis—An overview | ScienceDirect Topics. ScienceDirect. https://www.sciencedirect.com/topics/engineering/system-analysis

The Difference Between Centralized and Decentralized Networks | SolarWinds MSP. (2018, November 30). https://www.solarwindsmsp.com/blog/centralized-vs-decentralized-network

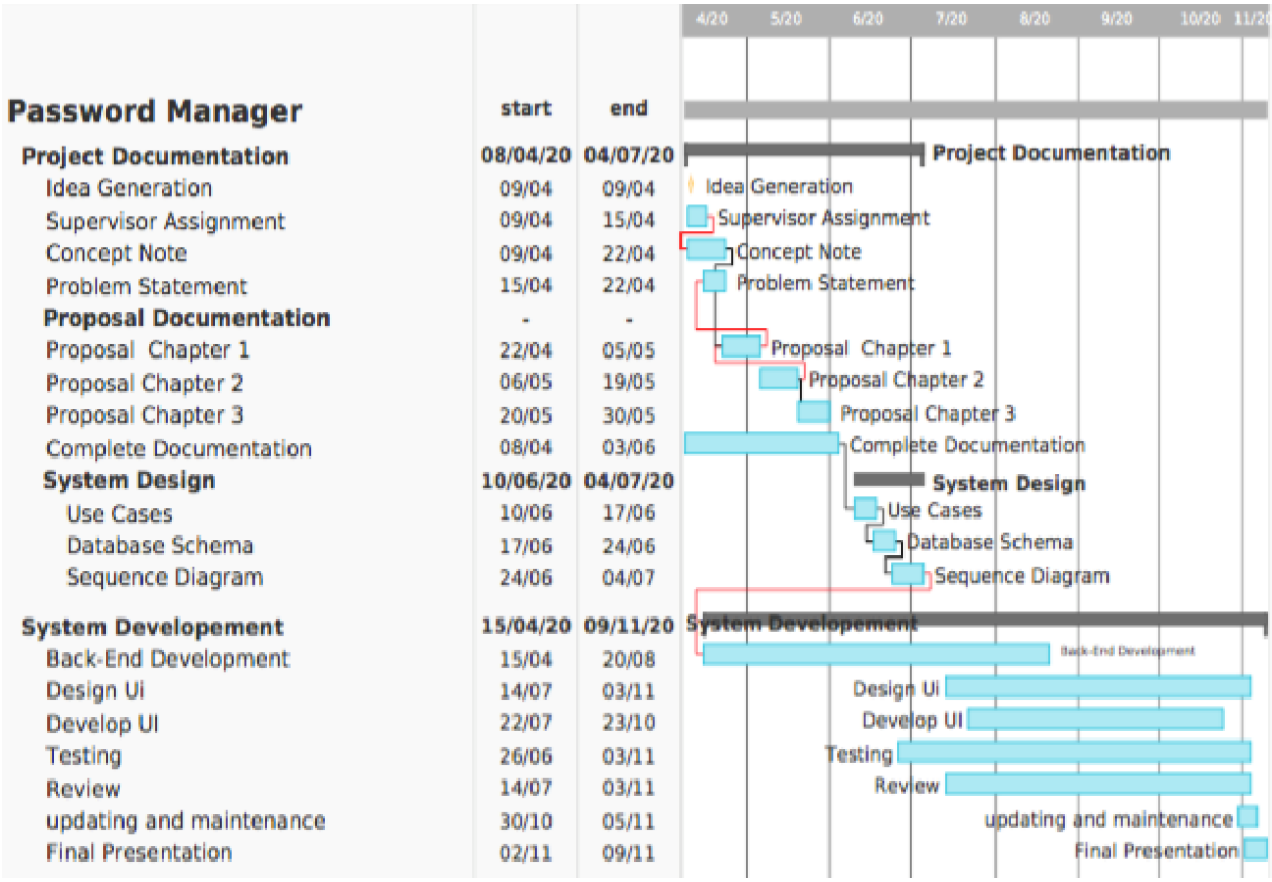
These are the top cybersecurity challenges of 2021. (n.d.). World Economic Forum. Retrieved January 22, 2021, from https://www.weforum.org/agenda/2021/01/top-cybersecurity-challenges-of-2021/

Tiwana, A. (2014). Peer-to-Peer Architectures—An overview | ScienceDirect Topics. https://www.sciencedirect.com/topics/computer-science/peer-to-peer-architectures

Zane. (2018, July 13). Password Security Report: 83% of Users Surveyed Use the Same Password for Multiple Sites. Cyclonis. https://www.cyclonis.com/report-83-percent-users-surveyed-use-same-password-multiple-sites/

Blasko, G., Narayanaswami, C., Raghunath, M.: A Wristwatch-Computer Based Password-Vault. Technical report, IBM Research Division (2005)

Appendix A: Time Schedule



Appendix B: Interesting Code