

PORTIO PASS: A PASSWORD SHARING APPLICATION

by
Sarah Oloumi

A project submitted to
the School of Computer
Science in partial
fulfillment of the
requirements for the
degree of

BACHELOR OF COMPUTER SCIENCE

Computer and Internet Security Stream B.C.S. Honours

At

CARLETON UNIVERSITY

Ottawa, Ontario
December, 2020

Abstract

The combination of fragmentation in the video streaming market alongside constant price increases for these services has resulted in many video streaming service subscribers to share their subscriptions in order to keep their costs low. Many normal users already have poor coping strategies when it comes to password management, and password sharing only jeopardizes their password security even more. The purpose of this project is to design and develop a secure iOS password sharing application as a means of helping users to diminish the risk of having their accounts compromised. Portio Pass is inspired by Apple's WiFi sharing system which allows Apple devices to share Wifi credentials without exposing the credentials to the recipient. The objective of this tool is to address the issue of unsafe password sharing by exploring the possibility of sharing encrypted credentials with a set of permissions controlled by the owner to the recipient.

Acknowledgements

Throughout this project, I have received a great deal of support and assistance. I would first like to express my appreciation and sincere gratitude to my supervisor, Prof. Elizabeth Stobert for her patience, continuous support, enthusiasm, feedback and her guidance with regard to my project. Her knowledge and expertise in usable security systems was invaluable in formulating the design and development methodologies for this project. Thank you for everything.

I would also like to thank Prof. Robert Biddle whose Human-Computer Interaction (HCI) course and lectures helped me understand the fundamental design principles and the development and evaluation practices of HCI in addition to its applications in usable security; resources which I continuously referenced during this project.

Lastly, I would like to thank Sean Sullivan for all his love, support and motivation throughout this project.

Table of Contents

Abstract	1
Acknowledgements	2
Chapter 1 Introduction	4
1.1 Motivation	4
1.2 The Research Problem	6
1.3 Contribution	6
1.4 Project Outline	7
Chapter 2 Background	7
2.1 Coping Methods for Managing Passwords	7
2.1.1 Differences in Password Management between Different Generations	11
2.2 Types of Sharing	13
2.2.3 Device Sharing	13
2.2.3 Proximate versus Distributed Sharing	14
2.3 Factors that Influence Sharing	15
2.4 Password Sharing Strategies Used Today	16
2.4.1 Borrowing	16
2.4.4 Helping	19
2.4.4 Broadcasting	19
2.4.5 Accidental Sharing	20
2.5 Implications for Security Design on Social Practice	20
Chapter 3 Methodology	23
3.1 Requirements Analysis	23
3.1.1 Overview of Project Idea, Personas, and Scenarios	23
3.1.2 Narratives of Potential Scenarios	29
3.2 Affinity Diagram and Key Requirements	31
3.2.1 Research Question	31
3.2.3 Descriptions of Functional Requirements	34
3.3 Initial Design Alternatives	37
3.3.1 Low-Fidelity Prototype of the System	38
3.3.2 Upgraded Low-Fidelity Prototype of the System	40
3.3.2 Low-Fidelity Prototype of the App's User Interface	41
3.4 Upgrade Low-Fidelity Prototype of the App's User Interface	42
3.4.1 Tools and Components Used For this Prototype	42
3.4.2 Storyboard and App User Interface	43
3.4.3 A Peek into Cloud Firestore	50
3.5 Technical Details of Portio Pass	52

3.5.1 Firestore Authentication	52
Firestore's authentication library allows us to authenticate Portio Pass users with Email and Password based authentication. Using the library, Portio Pass has assigned every user a unique UID and a token which is collected by Firestore Authentication upon user login[15]. Within the firestore also, Portio Pass has the ability to set its own set of rules for authentication if it wishes to later on. This is definitely a feature worth looking into for the future for making Portio Pass more secure.	52
3.5.2 Firestore Server-Side Encryption	52
Chapter 4 Conclusion	53
4.1 Challenges Faced	53
4.1.1 XCode Challenges with Portio Pass	53
4.1.2 Encryption Challenges with Portio Pass	53
4.1.3 Cloud Firestore Challenges with Portio Pass	54
4.2 Portio Pass Recommendations for Future Work	54
4.3 Summary	55

Chapter 1 Introduction

1.1 Motivation

In the past decade, media streaming services such as Netflix, HBO, and YouTube have dominated the entertainment industry and significantly changed the way users view media content. While 15-20 years ago most users would have had a cable or satellite TV provider, or simply viewed channels available over the air, many users today have cut the cord, and have opted in to subscription services. A recent survey illustrated that around 46% [1] of Canadians are subscribed to some sort of online streaming service. Not only has this been beneficial in that it allows users to access more content, but it has also created a better user experience and greater flexibility in that users can view any content on demand.

While initially, the price of watching content was low when compared to cable, there is now a series of tech-driven services that are trying to compete with each other, fragmenting the availability of content. In fact, there are now over 200 [2] streaming services for users to choose from. While the price of an individual service may be affordable, service prices have been constantly increasing [3]. Users' viewing habits are also changing. Users are no longer interested in simply watching a single show or movie, they want it all -- this is especially true during the pandemic as people are required to stay home as much as possible. Moreover, due to exclusive licensing of streaming content, individual streaming platforms are only able to provide a subset of the content a user would want to watch. As a result, most users have opted to subscribe to more than one service to satisfy their entertainment needs. It is no surprise then, that TV entertainment expenses for families have gone up; a recent report from Convergence Research reported that Canadians spent \$1.53 billion CAD on streaming services in 2019 [4]

(this is an increase of 37% from the year before) and is expecting the spendings to reach \$2.07 billion by the end of 2020 [4].

To lower the cost of streaming while still having access to a significant amount of content, most users have turned to sharing their passwords and credentials with their family and friends. A recent unofficial survey [5] by Jake Moore shows that over 60% of users have shared their account. Although this solves their problem in terms of costs, it has created an even bigger issue; users are putting their accounts' security in jeopardy everytime they share their credentials. This is especially alarming as many users today are still reusing the same password for multiple accounts. Therefore, this project aims to create a tool that enables users to easily and securely share their credentials, whilst providing additional control over how the recipient can use the credentials.

1.2 The Research Problem

Currently, the primary issue with password sharing is that there is little to no safe way of doing so. As of now, there are a small number of dedicated password manager applications and tools such as LastPass [6] and 1Password [7]. Regardless of the availability of these tools, Stobert et al.[8] found that people still resort to writing down the majority of their passwords, or simply reuse the same password for all of their accounts. This paper went into further detail and described that the subjects who were using dedicated password manager apps stopped using them after some time, reporting that these apps were inconvenient for them -- it was inconvenient for them to copy passwords out of the application.

Presently, most user-centered security design is aimed at users in an organizational setting. Therefore, the principal issue that needs to be addressed is how we can design and develop a secure but simple tool for individual users pivoted on usability. As of now, many

users sharing their account and subscription credentials are left vulnerable. Apart from changing their passwords manually, users have no control with regards to how and for how long their password is being used and managed by the recipient.

1.3 Contribution

Portio Pass aims to contribute to this issue by focusing on enabling users to securely share their passwords and credentials. This application's main focus is on usability and ease of use. It's intention is to provide a minimalistic application which allows for a safer method of password sharing. Portio Pass aims to empower the password owner by granting them a set of controls with which they can modify the parameters of password sharing. These parameters allow the owner to control the duration of time for which the password is available to the recipient, as well as how the recipient is able to manage the password once they've received it.

1.4 Project Outline

This project consists of four chapters. The second chapter goes into detail about the background research done which was on user behaviour and coping mechanisms for password policies, password management and password sharing. The third chapter goes into detail about the methodology used for the project -- it describes what tools were used and why they were used, what features Portio Pass is going to provide the users with, and why those specific features were picked for the application. Finally, in chapter four, a discussion is presented that will depict the challenges and limitations faced when working on the design and implementation phase of the project.

Chapter 2 Background

It is important to note that I was unable to conduct interviews and gather data due to Covid-19. As a result, I tried to do extensive research by reviewing prior work done within the context of this issue. This includes current struggles and coping methods for managing passwords, what is being shared the most at the moment, factors that influence how and why users share, current password sharing strategies, and finally, the implications sharing has on security design. This chapter consists of detailed notes and findings from the background research. I believe this is absolutely necessary for chapter 3 which describes the methodologies for this project.

2.1 Coping Methods for Managing Passwords

The number of passwords a user requires is continuously increasing as life becomes more digital. It's typically extremely difficult for a user to determine what a “good” and “secure” password is and not repeatedly use the same password for multiple accounts. Furthermore, users are also expected to update their passwords on regular intervals. As a result, users are facing a lot of issues when it comes to password management and have adapted unsafe coping strategies. Stobert et al. [8] explores users' behaviours in terms of creating secure passwords, and delves into the users' password coping methods when it comes to managing and storing them. Stobert et al. [8] indicates that password management applications address issues regarding a less costly password management system that addresses privacy and theft issues, and also removes the burden of remembering the password. However, after interviewing with the participants, Stobert et al. [8] discovered that password managing applications also have a number of usability problems. For the users, it was too inconvenient to copy passwords out of the application.

Though there has been an increase in various authentication methods (ex: two-factor authentication, email, and biometrics), passwords are still the most frequently used method of authentication today.

Apart from dedicated password manager applications, Stobert et al. [8] described that built-in system (ex: Apple Keychain), and built-in browser password managers and cookies were also used. Single sign-ons (i.e., a single password authenticates multiple websites) such as those provided by Facebook or Google were not unpopular either. This was sometimes paired with two-factor authentication or multi-factor authentication to create more security for the participants' passwords.

Stobert et al. [8] also reported that some of its participants were also leaning more towards password recycling as a way of managing their passwords. It was described that when password recycling, users would categorize and use the same password across several different devices and accounts in order to ensure they don't forget their password. This creates a significant risk where if a reused password is discovered by an attacker, it will expose multiple other credentials for other sites and accounts. This can prove fatal if the user has high risk accounts such as bank accounts or work-related accounts with the same password. Furthermore, password-composition policies heavily affect how users reuse previous passwords and create new ones. Through previous studies cited in their paper Stobert et al. [8], it seems evident that although more complex password-composition policies frustrate users, they also make them feel more safe and secure. Additionally, it was realized that some users still use fragments of their previous passwords in their new passwords when password-composition policies are changed to facilitate remembering the new password, resulting in long-term reuse.

When looking at how users managed their passwords, Stobert et al. [8] made sure to have participants from different ages, educational backgrounds, and responsibilities. This was done to create variety, and to better understand how different users cope differently. It was discovered that on average, participants had five passwords that they would reuse for all of their accounts and either Apple Keychain, a built-in browser password manager, or cookies for managing them. Interestingly, although no user was actually using a dedicated password manager application, a good number of participants were still writing down at least some of their passwords.

Stobert et al. [8] also explored the psychology and behaviour among the participants when it came to passwords and security. When interviewed, participants described anxiety towards having difficulty remembering and managing their passwords. Stobert et al. [8] reported that the main concern and worry for individual users is about “doing the wrong thing” when it comes to password creation privacy. As a result, they changed their behaviour and have tried to adapt to this new digital world. Most participants reported they no longer saved passwords in browsers as they perceived it to be dangerous and unsafe, while some reported to no longer be using any password managers; they perceived them to be not very time efficient. Furthermore, some participants reported keeping an eye out for security breaches and quickly changing their passwords from the fear of their account being compromised. Stobert et al. [8] discovered that participants were of the younger population who interact with technology on a day-to-day basis, they had a severe lack of understanding with regards to threat models. During the interviews, it was noted that participants could not differentiate between personal attacks, large-scale password hacks, and the loss of private data. It is essential that users are better informed about this. Having a better understanding will help participants better categorize their accounts and passwords for

reuse. It is also important to ensure users have a better understanding of the severity of threats and their frequency.

Stobert et al. [8] derived four coping strategies for users to allow them to have better support, and a better understanding of what they should be doing when it comes to security. The first is to keep a secure record of all passwords. This includes writing down and keeping the passwords in a secure place (either physical or digital such as Dropbox). Consequently, one can use a dedicated password storage application that uses encryption and decryption algorithms to store passwords securely or writing passwords on a paper and storing it securely.

The second method is to use password cues. During the study, it was noticed that some subjects had difficulty with regards to matching passwords to usernames or websites. Stobert et al. [8] suggested subjects use image cues in order to aid their memory -- this is similar to image-based anti-phishing mechanisms. Moreover, Stobert et al. [8] also suggested the use of password management applications, but with a twist. There were only a small number of participants who reported using such applications, all of whom indicated that they had stopped using them because of how inconvenient they were to use. Thus, the paper suggested developers and designers rethink their software to improve user experience in order to promote users to use them more frequently.

Last but not least, the fourth method by Stobert et al. [8] was to use single sign-on. When interviewed, participants in this study were not properly informed about single-sign on and the technology behind it. It seems that they believed their information would be used elsewhere, or their social media would become cluttered. To fix this, in addition to educating people about what single sign-on is and how it works, the study suggests that companies providing such services become independent entities and only verify authentication attempts.

It is clear that currently users don't have enough support when it comes to their credentials. Users must be better informed of the privacy and security concerns around passwords. New and better opportunities must be created to better support users when it comes to password management.

2.1.1 Differences in Password Management between Different Generations

Currently, there is a huge gap between how users in the 80s and 90s generation manage their passwords and how those in the older generations manage their passwords. Merdenyan et al. [9] examines how these generations behave differently when it comes to password management. They pointed out that although there is already existing research about how users manage their passwords, social desirability -- the likelihood of some participants giving answers that they think are appropriate, not what is actually true -- was never taken into consideration. To address this issue, in conjunction with a questionnaire about password management, participants were also asked to complete a much shorter questionnaire which was a reflection of Marlowe-Crowne Social Desirability (SD) Scale. The Social Desirability Scale is essentially a self-report in the form of a questionnaire which assesses whether or not participants are concerned with social approval. The outcome illustrates that half of the older participants were in the higher part of the social desirability scale, whereas the younger participants were mostly in the Low SD group. Thus, the potential for social desirability bias into consideration about every question accordingly.

After the interviews, Merdenyan et al. [9] concluded that only two of the fourteen questions received answers with high SD scores. The first question was in regard to sharing

passwords with others, and the second question was in regard to logging in from a shared computer. When observing the SD scores from the older participants, it was clear that half of them were on the high end of the SD scale for both of these questions. Consequently, although less people from the older group reported not sharing passwords or logging into their accounts from other devices, the results might actually not be 100% accurate. Moreover, it was observed that older participants did report storing their passwords more frequently than younger people. A reason for this was seen when they also responded to forgetting their passwords more frequently than younger people. As a result, it is essential now to inform older participants on proper password management and sharing. Additionally, on the other hand, younger people also continue to have risky password behaviors by continuously reusing their passwords without much variation. In consequence, regardless of age, people continue to undertake risky password management behaviors.

One thing that can be concluded from this is that when it comes to privacy and security, it's important to ensure users are comfortable with the system. The system must be convenient people to use regardless of age. A system that develops strong passwords, and assigns it to the different accounts a user has automatically as a means of password maintenance, would be very beneficial to users. This way users don't have to know when to change their password. It will change the password for them on a schedule, autofill passwords for websites and applications, and keep a record of the passwords for the user if they ever want to see their passwords.

2.2 Types of Sharing

Currently, a huge portion of devices presume that they will only have a single user. This creates a ripple of issues where all authentication solutions, password managers, OS level authentication, and other auto-backup cloud accounts are shared between users. An example of this could be that a password management system saves the credentials of both the guest account and the owner account. Consequently, an authentication system that enables users to stay logged on across different sessions can enable the guest to have unanticipated access to the owner's account and vice versa.

2.2.1 Account Sharing

Typically accounts are shared between family members and close family friends. This is because, both family members and close friends tend to mostly share their computer accounts, email, and entertainment subscriptions. A present issue with this is that users who use the same profile but not use them as intended will frequently observe personalization issues.

2.2.3 Device Sharing

Of all the devices, handset devices, tablets and computers are the most common devices to be shared. Handset devices are often shared to perform ad hoc activities with family members and friends, tablets are shared among the family members for activities such as playing games and reading books, and computers are shared primarily between family members for entertainment purposes or monitoring activities (ex: parents logging in to monitor children's online activity).

2.2.3 Proximate versus Distributed Sharing

Proximate sharing is described as several close relations using a certain device or account. An example of this is when all family members use one computer. Distributed sharing on the other hand, is described as when an additional network (a person or device for instance) is used to transfer information to the target user. An example of this can be seen when person A wants to contact another person B who does not have the resources to communicate. Consequently, person A will call another person C who is with person B in order to be able to communicate with them. It is important to note that although distributed sharing mostly occurs in developing countries or remote areas where there is a lack of resources, it can also occur in situations where one person does not have the knowledge or resources to communicate with someone directly (Ex: A person wanting to talk an older person who does not own or use phones may require to contact the person's caregiver initially in order to talk to them).

It seems that owners will typically attempt to regulate a recipient's access to their data as a way to provide their accounts and devices with some security. This is done through several methods including deleting or hiding sensitive data, utilization of device level passcodes, monitoring what is being shared by physically being present, refusing to share all together, profile switching and authentication methods. Profile sharing allows the owner to lower their privacy and security risks, but the process of profile switching is time consuming as the owner will have to set up a new profile and switch to it. Authentication methods on the other hand allow owners to have more control in ad hoc sharing scenarios. It allows them to set up custom access to whatever they desire before sharing.

2.3 Factors that Influence Sharing

Mathews et al. identified three factors which influence sharing: trust, culture, and utility. In terms of trust, they've identified that there is a direct relationship between the amount of trust between two people and their willingness to share devices or accounts with one another. It is important to note however, that this relationship did not always extend to parent-child relationships.

Culture also has a significant impact on sharing. For instance, this can be observed in Indian vs American culture. In India, family, friends, and even neighbours are very involved in each other's lives, thus it is normal for one to reach for another's device. The same can not be said for American culture where everyone is entitled to having privacy. Furthermore, spouses are also an integral part of some cultural expectations. In a country like Saudi Arabia for instance, bank and phone credentials are shared not just as a sign of trust but also due to cultural expectations. Consequently, the sharer might not always be comfortable with sharing but is obligated to do so anyways due to culture beliefs. Last but not least, there is a direct relationship with how much utility the sharer gets from sharing their device or account to a sharee whom they trust and their willingness to share.

The Socio-economic status of the sharer is also essential to note. An example of this would be if a family might not be able to afford having multiple computers for their children. Thus the children will have to share one computer to get their work done. This raises issues regarding personalization which will be spoken about in further sections.

Last but not least, utility hugely influences sharing habits between people. In some areas, sharing is essential due to lack of resources. For instance, a person may be in an inaccessible and remote area where they have no access to the internet, so they will share their account with someone whom they trust in order to get their errands done

2.4 Password Sharing Strategies Used Today

Mathews et al. established five different sharing methods that are common today. The next five subsections will go into detail to explain these methods.

2.4.1 Borrowing

The first and most common type is borrowing (Ex: a friend is sharing another friend's handset device and its pin when they're out to check their email and occasionally use other apps). In terms of what is most frequently being borrowed, Mathews et al concluded that handset devices are the most frequently shared devices, followed by computers and then tablets.

Borrowing occurs for mostly three reasons, convenience, content, and capabilities. It seems that people usually use someone else's device if it was located at a more convenient spot than their own. For instance, a user might borrow another user's phone because their phone was left at home to charge. Furthermore, some devices have specific content that is only available on that specific platform (Ex: A child borrowing their parent's handset/tablet to play a game that is not available on any other platform). Last but not least, operating system level and hardware level capabilities of a device also affect if or when a device is borrowed. A friend using their friend's android phone because Google Assistant gives more relevant results than Siri or a friend using another friend's phone to take a picture because their camera is better are perfect examples of this.

When it came to coping methods, Mathews et al. reported that users use a mix of security approaches when they shared their device with the recipient. It seems like most owners typically allow for full and unsupervised access but this only occurs if the owner fully trusts the recipient. At the same time, there are some owners who will only allow limited and supervised sharing accesses regardless of their trust in the recipient.

2.4.2 Mutual Use

The second most common type of sharing reported by Mathews et al. is categorized as “mutual use”. Mutual use does not involve an owner sharing their account or device with a recipient, but both people having equal ownership over a single device or account. Between the two, devices are observed to be the most frequently shared item.

Mathews et al. came up with three reasons why mutual use occurs: necessity, limitations on resources, and convenience. For instance, between spouses there is usually a mutual account as there is a need to share a bank account especially if they mix their finances. Moreover, lack of financial resources can also limit people to only a limited number of devices or accounts. To put this case in point, currently there are many subscription-based accounts and when their costs accumulate, it can be a very heavy expense. Mathews et al. observed that such account credentials are typically shared between the participants’ families and friends in attempt to reduce such expenses. Last but not least, it seems that the participants (between their families and friends) would typically opt and reach for whatever device was closer to them. Consequently, there is a direct relationship with how frequently a device is shared between two people and how convenient it is for them.

Additionally, among the participants, Mathews et al. observed that those who would mutually share a device or account would know all of the credentials for it. Sometimes however, users will set up OS-level profiles alongside with different browser profiles on mutually shared devices. This is something that happens a lot not just at home, but also in the workplace. At this time, every user only knows their own credentials and although it seems more private, it may actually result in accidental sharing at times as owners would forget to log out of their accounts.

This is very dangerous if someone with malicious intent also has access to the device and those accounts.

2.4.3 Setup

After mutual use, the third most common sharing is “setup”. Mathews et al. describes this as an event which often involves someone other than the owner, executing initialization, maintenance and configuration activities on an owner’s device or account, which can be done either in person or remotely. Mathews et al. reported that most of the time, users share devices rather than accounts. When accounts were shared, it was mostly for sensitive information such as credential management systems and bank accounts.

This type of sharing occurs when the owner does not have enough information and knowledge with regards to properly setting up their accounts. This type of sharing is typically rare and usually happens only when someone is setting up an owner’s account/device. In some cases this sharing can occur more than once as the owner might want the person to frequently maintain the account/device for them. In this case, the person would typically have full access to the owner’s credentials. It is important to realize the risks of this and thus most owners will typically have “setup” sharing with either someone they trust (ex: an elderly asking their child or grandchildren to help them) or with someone from a position of authority. An example of this would be when an ISP technician takes over the control of your router remotely to reconfigure it.

2.4.4 Helping

Mathews et al. described another method of sharing to be “helping”, which is when an owner shares their device or account to help them get their tasks done and make things more convenient for them. An example of this would be a friend entering an address on the owner’s phone or

making a call for them while they drive. Furthermore, participants reported that they would mostly just share their devices-- handsets are the most common Mathews et al. --rather than their accounts.

It is good to note here that typically, when an owner shares with a recipient here, the recipient will have full access to the owner's device or account. As mentioned in section 2.3 regarding trust, most people seem to only allow the recipient to have full access to their device and account if they fully trusted the individual. Mathews et al. concluded that because of how exposed owners feel during this sharing, it rarely occurs and is dependent heavily on the activity or tasks that need to get done. Answering calls/texts, executing in-vehicle navigation, helping with finances and credential management are just some examples of when this sharing occurs.

2.4.4 Broadcasting

Broadcasting is when at least two people use or view the same device concurrently Mathews et al. An example of this is when an owner shares a video or picture (some media content) with a recipient on their own device. Mathews et al. concluded that the most common devices shared when broadcasting are handset devices as people always carry their phones with them. Moreover, Mathews et al. also reported that users seem to only perform this type of sharing when they want to share the enjoyment of some content together with a recipient. It seems that this also only happens when the owner trusts the recipient .

2.4.5 Accidental Sharing

Accidental Sharing occurs when the recipient gains access to the owner's account or device without the owner knowing or intending for it. Mathews et al. characterized all accidental sharing stories in their research to be disclosed by the users were only about accounts (mostly

social media/communication accounts and several work accounts). An example of accidental sharing is when users forget to logout of the accounts on devices which are being shared mutually at work for instance. It is not surprising then to see how unknowingly and very easily they revealed their credentials. Last but not least, unlike other types of sharing mentioned in previous sections, due to owners being unaware of sharing here, Mathews et al. concluded that the owners must also not have had any trust in the recipients either -- they unintentionally share credentials with an unknown person.

2.5 Implications for Security Design on Social Practice

Singh et al. goes into detail about how because of inconvenience and lack of resources, many people have resorted to sharing their credentials (especially banking credentials such as banking PINS). This is mostly due to the fact that there are very limited systems which are both secure and easy to use for individuals; most user-centered security design is aimed at users in an organizational setting.

There are three defined categories for user-centered security, with the first being the relationship between activity done and the usefulness of that activity. For most users, the usability of an application is much more important to them than the security, thus it is important to keep passwords easy enough to help with usability remember but difficult enough to guess to prevent breaches. Furthermore, it seems that what users currently perceive to be secure is simplicity -- users want an easy to use site with readily available support.

The second category is the relationship between trust and security. It seems that, on its own, usability security is not enough for trust. Human values are essential for establishing a level of trust between two parties and maintaining it. This trust falls further into two categories, hard trust, and soft trust.

Hard trust is authenticity, encryption, and security transactions. Designers and engineers are in charge of developing hard trust -- they are in charge of making decisions to what type of encryption algorithm is used, what type of authentication is used (Ex: 2FA vs SSO), and what security protocols are used in order to maintain a secure application.

Soft trust is more concerned with human psychology (i.e., the developed trust between two parties), user friendliness (i.e., how easy to use do users perceive an application or system to be), and brand loyalty (i.e., users developing trust towards a certain company due to their transparency, their history, and usability through providing tools for users to customize their own privacy).

Last but not least, the third category for user-centered security revolves around privacy and control over personal information and credentials. Privacy is different from security in that privacy is the ability of users having control over their personal information and how they wish to share it. On the other hand, security relates to how users' personal information is protected. For an application that truly has user-centered security, it is important to take personalization and user control into consideration. In order for designers and developers to better improve their applications to focus on user-centered security, they must thoroughly understand how security and privacy are different.

In terms of the implications for security design on social practice, Singh et al. reports that sharing is done between couples due to trust and convenience, it's done within remote aboriginal communities in order to survive, and sharing is done among people with disabilities due to inaccessibility. Consequently, privacy and security systems should be regulated and redesigned to become more accessible and usable. Furthermore, apart from designing more

usable systems, designers should also focus on designing systems where personalization and accessibility is possible and easy regardless of whether or not the account or device is shared.

Four proposed improvements for security systems for individual use are as follow Sing et al.:

1. Security systems should have more flexibility especially when it comes to delegations. This ensures that more than one person can access the account when everyone agrees to it.
2. Personalization should be kept in mind when designing security systems as it will help customers have more privacy since they will have more control over what information they share and don't share.
3. When designing security systems, designers must take into consideration the need for access in remote areas. This will ensure other remote communities (Ex. Australian indigenous communities continuously put their security and privacy at risk due to lack of technological resources Sing et al.) will be able to easily use the services without jeopardizing their privacy.
4. When designing new security systems, designers must keep accessibility in mind to ensure that those with disabilities can also have a good user experience.


As Sing et al. reported, many users complain about design and usability for individual use applications. This is especially true when it comes to password policies and how most tend to share because of convenience. An application whose main focus is on simplicity, ease of use, and follows user-centered security, will be one that attracts many users. If done with the right tools, and with people one trusts, the benefits of credential sharing outweigh the negatives. My goal for my application is to focus on usability and clarity. Providing a good user experience for users, and allowing them to securely and easily share their credentials with people they trust, will mitigate current risks that many users take. It will encourage them to disengage in risky behaviour that puts their accounts' passwords in jeopardy and establish more control over their credentials. The next chapter will go through a simple prototype of an application built specifically for individual users to address their needs for password sharing.


Chapter 3 Methodology

3.1 Requirements Analysis


3.1.1 Overview of Project Idea, Personas, and Scenarios


The various types of users shown below represent possible candidates from an average demographic. That is, there are both people who are knowledgeable in security and technology in addition to those who are not. The goal is to include different variations of users with different backgrounds and interests. There are many factors which influence sharing; here are a few personas that span the demographic.


Jeff Burns -- The Dad		
<p><u>Background</u> Age: 38 Work: Sales engineer Family: Married with three kids Location: Ottawa, ON</p>	<p><u>Bio</u> Jeff is a sales engineer at a medium-sized company who travels frequently through the year for work. He lives with his wife and three kids in Ottawa. Jeff and his wife both have two separate bank accounts--each pay for certain bills around the house--and a single joint account for their savings. Because he travels a few times during the year, Jeff shares his banking credentials with his wife so she can pay for the bills in case he doesn't have the resources to access his accounts. For his credentials, Jeff has organized them all in an encrypted excel document. He shares the password for this document with his wife. Furthermore, both Jeff and his wife's parents live in a city fifteen hours away, thus usually, their only method of communication is online. Also, as neither of their parents know much about technology, Jeff had to install some social applications (i.e. Skype and Whatsapp) for them in addition to maintaining their accounts. Last but not least, because they have three younger kids, they are subscribed to several entertainment and gaming services.</p>	
<p><u>Personality</u></p> <ul style="list-style-type: none"> • Ambivert • Empathetic • Effective communicator • Active 		<p><u>Goals</u></p> <ul style="list-style-type: none"> • Save up for the kids' tuition. • Work promotion for a higher salary. • Go on one vacation a year with his family. • Care for his elderly parents.
<p><u>Device and Internet Usage (on a scale of 1 to 10, 1 being least preferable)</u></p> <ul style="list-style-type: none"> • Desktop Devices: 7 • Handset/Tablet Devices: 8 • Social Media & Networking: 7 • Technical Know-how: 8 		
<p><u>Motivations</u></p> <ul style="list-style-type: none"> • His motivation is to create a good life for his family. • To facilitate his daily activities as much as possible. • Find ways to spend less on non-essential services and items and save more money for his family. 		<p><u>Frustrations</u></p> <ul style="list-style-type: none"> • Lack of communication and transparency in work life. • Having to spend too much money on unnecessary services. • Not having a lot of time to improve his own skills between spending time with his family and his parents.

Andrea Moreno -- The Mom		
<u>Background</u> Age: 36 Work: Registered Nurse Family: Married with three kids Location: Ottawa, ON	<u>Bio</u> Andrea is a registered nurse and works full-time at her local hospital. She is married to Jeff, and lives with him and their three kids. When she is not at work, she is doing chores around the house and taking care of her kids. Her and husband both have separate bank accounts for paying the bills in addition to their shared savings. She's taken responsibility for buying groceries, clothes, and hydro/electricity bills while her husband pays for the mortgage and internet/mobile services. Moreover, she loves using her iPhone for such things. She has her banking credentials saved on her phone's keychain so for her, making the payments is just one click away. If her other applications and services allow her, she typically saves her passwords in the keychain. As a way to make some more money, Andrea and her husband recently bought a new house and have been renting it through Airbnb. It is a bit more work for the both of them, but it contributes more money to the kids' tuition savings and increases funds for their vacation budget.	
<u>Personality</u> <ul style="list-style-type: none"> • Extrovert • Empathetic • Detail-oriented • patient 		<u>Goals</u> <ul style="list-style-type: none"> • Save up for the kids' tuition. • Ensure her kids grow up to be healthy by instilling healthy habits in her kids. • Spending quality time with her family every week.
<u>Device and Internet Usage (on a scale of 1 to 10, 1 being least preferable)</u> <ul style="list-style-type: none"> • Desktop Devices: 2 • Handset/Tablet Devices: 9 • Social Media & Networking: 3 • Technical Know-how: 4 		<u>Frustrations</u> <ul style="list-style-type: none"> • People at work are not empathetic towards one another. • Not getting paid enough for all the work they do. • Forgetting passwords to her account subscriptions and having to spend time going through keychain to find out what they are.
<ul style="list-style-type: none"> • 		

Gale Burns -- The Grandmother

<p><u>Background</u> Age: 70 Work: Retired Family: Lives with her husband Location: Thunderbay, ON</p>	<p><u>Bio</u> Gale is retired and living with her husband in Thunderbay. As neither her nor her husband are very tech-savvy, her son sets most of her accounts (ex: Email, telecommunications apps, Kijiji) and devices up. In the past her son, Jeff, has tried to store her passwords on an encrypted excel sheet, but Gale always forgets the master password. Consequently, she now has all of her passwords stored on a physical notebook in her book closet. Lastly, since Gale and her husband both no longer work, they've started some new hobbies and activities. Alongside gardening, going on walks, and joining a few social clubs, they have also turned to entertainment -- especially Gale. It's the easiest and most convenient way for her to watch her food shows and get ideas for her own cooking and baking.</p>	
<p><u>Personality</u></p> <ul style="list-style-type: none"> • Extrovert • Generous • Present • Humble 		<p><u>Goals</u></p> <ul style="list-style-type: none"> • To be able to maintain her relationship with her grandchildren. • To enjoy her retired life with her husband. • To make delicious pastry and food for her new social clubs.
<p><u>Device and Internet Usage (on a scale of 1 to 10, 1 being least preferable)</u></p> <ul style="list-style-type: none"> • Desktop Devices: 4 • Handset/Tablet Devices: 2 • Social Media & Networking: 1 • Technical Know-how: 2 		<p><u>Frustrations</u></p> <ul style="list-style-type: none"> • Forgetting a password and then not being able to find her password book. • Not being able to learn how to easily navigate and reach through to her favourite applications. • Not being able to spend a lot of time with her grandchildren as she lives so far away.
<p><u>Motivations</u></p> <ul style="list-style-type: none"> • Being able to cook delicious food and make people happy. • Being healthy and active to increase longevity so she can be there for her grandchildren longer. 		

<p><u>Background</u> Age: 35 Work: Software Developer Family: Lives on his own Location: Ottawa, ON</p>	<p><u>Bio</u> Briant is a software developer at the same company as Jeff-- he met Jeff a few years ago at a company social event and they've been friends ever since. Because of his job, Briant has a very good understanding of most tech. When he has the time, he will read through new articles relating to the latest advances in security, privacy, machine learning, AI, gaming technology, etc. For his credentials, Briant uses LastPass to securely store his passwords. Like most typical developers, Bryant is not a huge extrovert. Apart from a few occasions where he might go to some social events, Briant really enjoys going home after work to a nice movie or game (he is subscribed to many entertainment services since he has not too many other costs) for some relaxation and then working on his own personal coding. Although Briant is not huge on hanging out with people in-person, he streams his games and sometimes programming once a week on Twitch for fun.</p>	
<p><u>Personality</u></p> <ul style="list-style-type: none"> • Introvert • Intuitive thinker • Kind • Resourceful 		<p><u>Goals</u></p> <ul style="list-style-type: none"> • To continue learning and getting work related certifications for a bonus or promotion. • To continue challenging himself through his personal projects. • Become more of an extrovert -- have an easier time talking to people.
<p><u>Device and Internet Usage (on a scale of 1 to 10, 1 being least preferable)</u></p> <ul style="list-style-type: none"> • Desktop Devices: 8 • Handset/Tablet Devices: 8 • Social Media & Networking: 5 • Technical Know-how: 9 		<p><u>Frustrations</u></p> <ul style="list-style-type: none"> • Not being challenged enough. • People at work not being detail oriented and Briant having to catch their mistakes. • Not being on time.
<p><u>Motivations</u></p> <ul style="list-style-type: none"> • Solving any problem • Learning about the news technology. • Getting things done, regardless of the task. 		

Lisa Shmid -- The Stranger		
<u>Background</u> Age: 25 Work: Student Family: Lives with roommates Location: Edmonton, Alberta	<u>Bio</u> Lisa is a student at University of Alberta and is in her last year of accounting. She doesn't quite know what she will do after graduation but she has applied to a few jobs within the country that were of interest to her. Currently, she's working as a TA for one of her professors -- She is hoping that she can get a good recommendation from them for her future employers. In her free time she enjoys doing active activities such as running, and swimming alongside with some more laid back activities like watching TV. Since she has been trying to save up money to pay her tuition, she does not pay for any entertainment services -- instead, she usually uses her roommate's TV to watch shows on Netflix, Hulu, and other media streaming apps.	
<u>Personality</u> <ul style="list-style-type: none"> • Ambivert • Athletic • Agreeable • Calm 		<u>Goals</u> <ul style="list-style-type: none"> • To graduate school as early as possible. • To save up money for her student loans. • To have work experience and some recommendation letters from her professors for when she starts working full-time
<u>Device and Internet Usage (on a scale of 1 to 10, 1 being least preferable)</u> <ul style="list-style-type: none"> • Desktop Devices: 6 • Handset/Tablet Devices: 8 • Social Media & Networking: 8 • Technical Know-how: 7 		<u>Frustrations</u> <ul style="list-style-type: none"> • Not being able to buy or pay for everything she wants. • Not understanding a concept in her studies.
<u>Motivations</u> <ul style="list-style-type: none"> • Finish her studies and start working. • Paying off her loans. • Having more control over her life. 		

3.1.2 Narratives of Potential Scenarios

All of the scenarios narrated below address the basic concerns regarding password sharing. These stories describe the typical sharing situation between two or more users. There are a few tools that are currently available for sharing passwords, but, as mentioned in Chapter 2, they are either not user friendly, or are designed for enterprise use rather than individual use. The scenarios described below try to address different scenarios where account owners want to share their account credentials with a recipient.

Scenario 1 -- Password Sharing Between Family Members

Narrative 1.1 Jeff manages most of the accounts and services his family is subscribed to. Apart from their essential services such as their financial services, his service provider, and his energy provider, Jeff also has been subscribed to multiple media streaming services to provide entertainment for his family. Typically, Jeff will create lengthy and random passwords that include both upper and lowercase characters, numbers, and special characters. Alongside with improving the security of his account, this makes it harder for him to remember all of these passwords and their corresponding accounts. Consequently, he stores these passwords, their corresponding accounts, alongside with the dates which he generated them on in an encrypted excel file.

Narrative 1.1 -- Issues to address/ Owner's wants:

For Jeff, it's a lengthy process when he has to look through this excel file every time he wants to have access to his credentials to share that service with his family.

- Ideally, he is looking for a simple tool that can securely share this credential with his family.
- If the old password is ever updated, the tool can be set to automatically share the new password with his family.

- The tool should allow Jeff to add each member individually -- create profiles for them-- and add devices per member. For instance, for his wife, he should be able to create a profile for her and as he shares credentials with her, he can add her devices to her profile.

Narrative 1.2 Jeff's parents retired a few years ago. In order to provide his mother, Gale, with some sort of entertainment during their free time, he wants to give her his media streaming credentials. However, he does not want to give her the password directly as he does not trust in her password storing methods (she stores her passwords on a physical notebook. Furthermore, because she lives far away, he can't go to her house and enter the passwords for her.

Narrative 1.2 -- Issues to address:

- Preferably, he wants a way to share his passwords with her remotely and have it be as simple as possible for her to log in to his accounts.
- Jeff should be able to allow the application to update the password on her device anytime he updates the passwords on his end.
- Also, Jeff also expects the password to be filled in automatically on Gale's end so she doesn't have to do anything.

Scenario 2 -- Password Sharing Between Friends

Narrative 2.1 Jeff's friend and coworker, Briant, has asked Jeff if he can share his media streaming application credentials with him if he pays him for part of the costs. Jeff would like to make sure that Briant doesn't share the password with everyone else as it might jeopardize his own security and privacy.

Narrative 2.1 -- Issues to address/ Owner's wants:

- Ideally, Jeff can set permissions for each device the credentials are shared with.

- Jeff can control who Briant shares the password with by either not letting him share the password at all or allowing him to share the password.

Scenario 3 -- Password Sharing Between Strangers

Narrative 3.1 Jeff and his wife Andrea have a rental home on Airbnb together. In order to interest some more people in their place, they offer to temporarily share their media entertainment passwords with them during their stay.

Narrative 3.1 -- Issues to address/ Owner's wants:

- Must be able to share an encrypted password with them without allowing the stranger to actually see the password and credentials the owner has provided them with.
- The application should have a feature where the password is either retrieved from the user or updated so that the user can no longer use the application without asking the owner for permission again.
 - This can be a feature where Jeff or his wife dictate the period of time a password is available to a device. After this time, the password is destroyed, a new password is regenerated, and the new and updated password is sent out to all other devices that were using it.

3.2 Affinity Diagram and Key Requirements

3.2.1 Research Question

The question to be answered derives from the domain and goals of the project. It is formally stated as: How do account owners manage their passwords when it comes to sharing them with other users?

3.2.2 Diagram

The diagram below follows the process outlined in the course notes. The themes were created by grouping related data together; descriptions flowed naturally from narrative issues in section 3.2.1 and observations gathered from various papers and research done briefly mentioned in Chapter 2.

Themes	Usability	Security	Design	Privacy
Descriptions	Effectiveness: Both owner and user must have the ability to easily share and receive passwords	Anytime the owner sends out their credentials to a potential receiver, that user must be identified and authenticated.	Visually, the interface must not be cluttered. It must be clear what actions are where. The user must be able to predict the consequences of all of their actions.	Prevent information leakage through output. It's best to ensure there aren't any logs or detailed error descriptions in case of an attack.
	Efficiency: The user should only be used to click on different elements.	Passwords must be stored encrypted anytime the user is not actually requesting to view them. This can be done by using cryptographic hash algorithms.	Selection controls must be simple. There must be a good balance between freedom of choice and control for the users.	Complete transparency, choice and control over information to foster confidence. Users will know who is collecting data, what information is being collected and used, and what personal information is being shared if any. No silent ('secret')

				updates
	Engagement: The tool must be pleasant to look at. For the users, regardless of their knowledge of computers and security, it should be easy to interact with.	Passwords must be encrypted while the owner is sending (sharing) it with the receiver.	The application must be designed with scale in mind. Adding new features in the future must be easy.	Educate users on how to best manage their privacy information and protect their personal information by showing them simple information about privacy options.
	Error Tolerance: Must ensure all elements and actions are distinct so that it's difficult for the user to perform any incorrect actions. It must be difficult for the owner and receiver to perform any irreversible actions.	The owner can choose to let the receiver see the password if he wishes.	All functionalities must be documented clearly. It must be clear what the input is,, and what the final output is. Must state what the function requires to be performed -- not how it's performed.	Application will never be enabled socially as that will connect the user with other third-party communities which may hold significant privacy implications
	Learnability: Users must be able to quickly learn about the functionality of password sharing. They must be able to predict what everything does correctly. Interface should	The owner must be able to take back a password/credentials if he/she wishes to do so. This can either be done either manually or by setting a timer to take credentials back	Identify things that might change at some point and ensure to make note of it in the system. This can include how the encryption algorithm works.	There will be no mobile advertising via this app. Users typically stray away from targeted advertising as it makes them feel like they have little privacy.

	be consistent. If new functions are added, there should be controls and explanations to help the user learn faster.	automatically for either a single account, or all of them.		
		The password and credentials must auto-fill for the application when the receiver tries to login. They should not have to copy and paste anything.	System must be designed to be testable. Automation and manual testing must be possible to test the system.	If there is ever the need for location data, the application must capture appropriate consent to use location data.

3.2.3 Descriptions of Functional Requirements

From the affinity diagram above, and the list of issues from the potential narratives, it is clear that our initial design alternatives should incorporate the following key requirements.

Functional Requirement No.	Functional Requirements Description
FR 1	All credentials that are stored in the server must be encrypted.
FR 2	Owner must be able to select an account's credential and share it with another user remotely.

FR 3	The date and time an owner shares a password with another user must be recorded and displayed to the owner.
FR 4	The owner must be able to set permissions and allow users with whom they share their credentials with to also share that credential with another user. This only goes up to two levels.
FR 5	The owner will be able to see everyone with whom they've shared the credentials for a specific subscription/account both directly and indirectly.
FR 6	The owner can choose to automatically share their updated credentials for an account with the other user if they want.
FR 7	Have permissions for allowing users to read the password.
FR 8	Credentials must autofill for subscriptions and accounts so users never have to enter or copy paste their credentials.
FR 9	The database of user profiles must contain information regarding user ID, name, and email at minimum.
FR 10	Selection controls must be simple and intuitive to users.

Non-Functional Requirements

Non-Functional Requirement No.	Non-Functional Requirement Description
NFR 1	The system must have an updated backup of all the credentials belonging to a user on their local computers.
NFR 2	Before an owner can share a password with another user, the user must be verified.
NFR 3	
NFR 4	There must be documentations and notes regarding different components that may require change in the future to allow for scalability.
NFR 5	A user must be authenticated before the password is shared with them.
NFR 6	All sharing must happen over TLS. This will accomplish encryption, authentication of users, and integrity of the data that is being transmitted.
NFR 7	Logs regarding errors and warnings must not give identifying detail.
NFR 8	All passwords must be generated for the users to ensure they're both lengthy and strong, but in case a user has two accounts that have the same password, they will not be able to share until one of those passwords are updated.

NFR 9	Users must be able to share passwords regardless of what browser they're using.
NFR 10	There should be no need of a database administrator, either to manage or maintain the contents for user-profiles.

3.3 Initial Design Alternatives

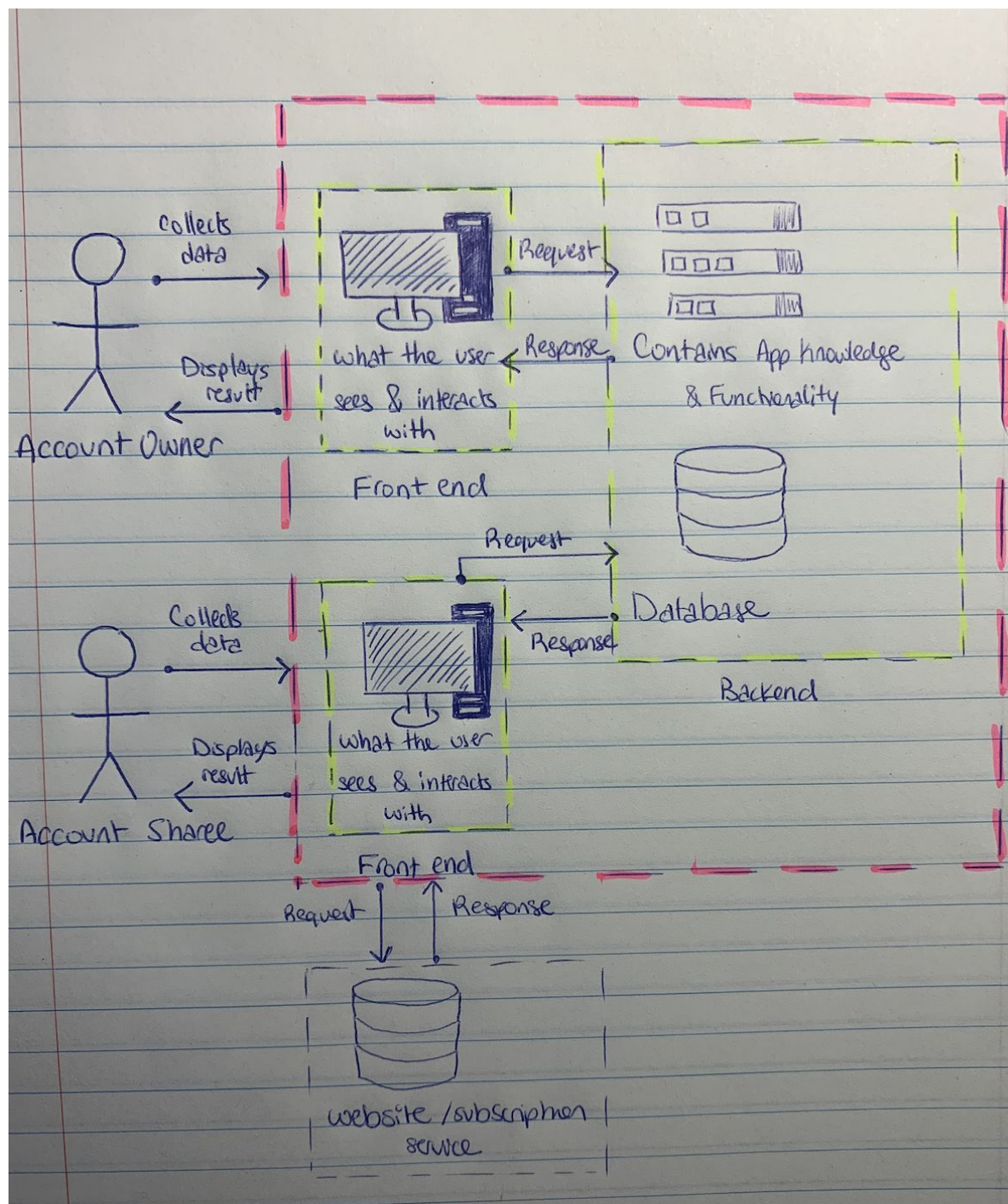
PortioPass, a proof of concept, transforms the FR and NFR requirements from Section 3.2.3 into a conceptual model. The rationale behind this model is based on an in depth analysis of the requirements. The main theme that emerged from the background research done for this project adapted the model composition. Users need a tool that facilitates secure sharing of accounts and provides them with control over what they are sharing. They want to be able to easily grant and revoke access when they desire.

The prototype shown in Section 3.3.1 segregates the system into three sections: the front-end which consists of the devices users interact with, the backend which is the cloud database and contains the functionality of the server, and finally the website subscription service which interacts with the front-end via request/response methods. This low-fidelity prototype was done on paper during the early stages of my design, and is only an abstract of how Portio Pass's system should be at a very high-level. While it may seem like a simple technique, it allowed me to explore different ideas and make adjustments as needed.

The prototype shown in Section 3.3.2 is an update to the one in section 3.3.1. Although this is still a low-fidelity image, it is evident that the system will be working with mobile applications and a cloud server (specifically Google's Cloud Firestore). The connection between

the application and the subscription service has also been removed. This is to make the system a bit more simple and manageable for the proof of concept.

3.3.1 Low-Fidelity Prototype of the System

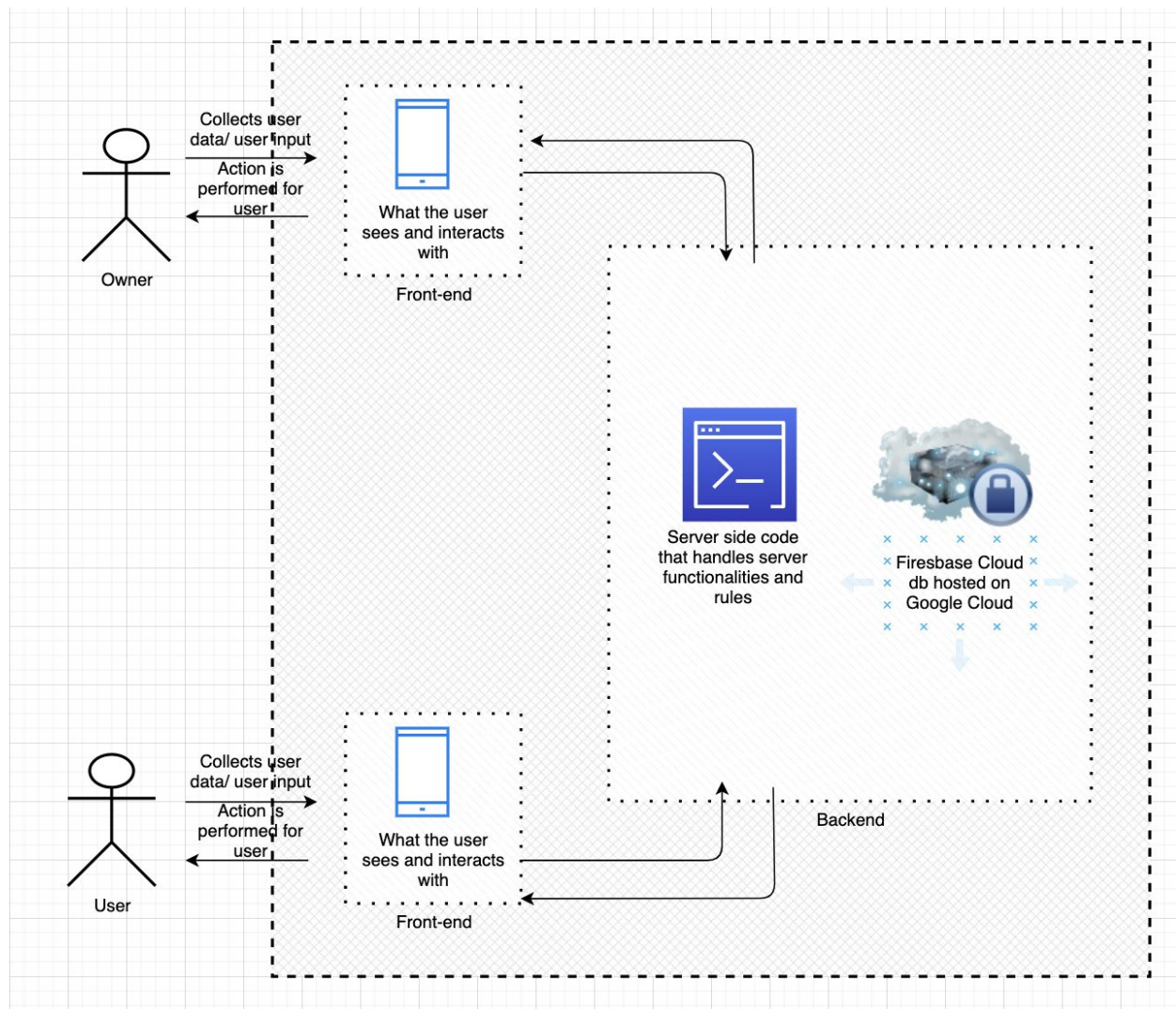


A High Level Example of How the System Works:

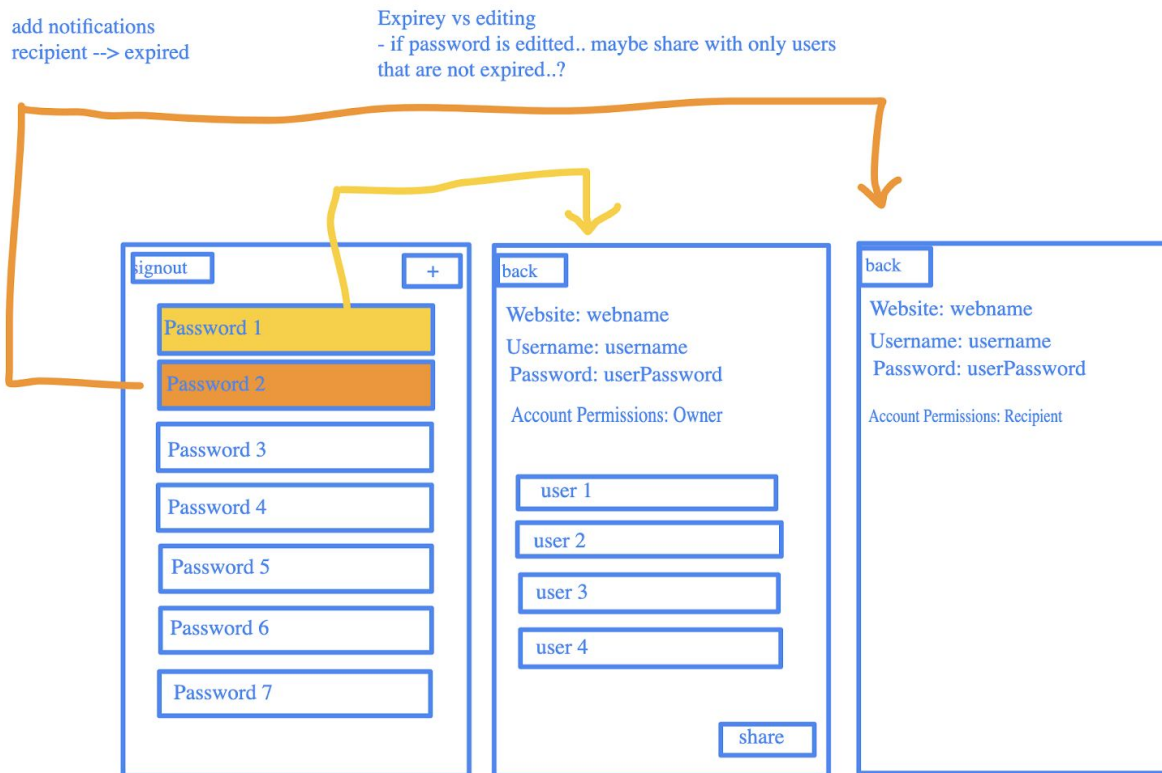
An account owner will interact with the interface and enter data, requesting that another user can have access to one of their subscription services. They will enter who the user is, what permissions they will have (i.e., whether or not they have the ability to share that credentials with another user), and for how long they will have access to those credentials. The server will look through its database to find the user who the owner wanted to share credentials with, authenticate them, and then create a copy of those credentials and add them to the user's own database of credentials. It will then send data back to the owner of the account with information regarding success of the transaction, the date and time, and the permissions the owner had previously set.

On the other hand, it will also notify the account sharee that a password for a new subscription has been shared with them. When this user requests access to the contents of that subscription, the system will look through their table of credentials and check to see if credentials for that subscription exists. If it does, it'll autofill and grant them access.

3.3.2 Upgraded Low-Fidelity Prototype of the System



3.3.2 Low-Fidelity Prototype of the App's User Interface



This is a simple, low-fidelity prototype of how the application will look like. The account owner will select one of their passwords which takes them to a second page (following the yellow arrow will display the page an account owner would see). At this point they have a visible list of all users that they can share the password with. They simply write what privileges they want the other user to have, which is one of: owner, recipient and share, or recipient only, and press share.

Meanwhile, when a recipient gets a new password, a new password button is placed for them in their list of passwords. They can simply click on it to view the data given to them by the owner (following the orange arrow will display the page a recipient would see).

3.4 Upgrade Low-Fidelity Prototype of the App's User Interface

Perception and color scheme for the tool was developed using principles from COMP3008 regarding cognitive processes -- specifically with GOMS in mind. GOMS stands for Goals(i.e., what are the goals of the user?), Operations (i.e., what would be the cognitive and physical process of the user reaching the goal?), Methods(i.e., what procedure is required to accomplish the goal?), and finally, Selection(i.e., If there is more than one method, what is the best method to use?). The layout of the sections followed the CRAP rule for perception. This was done to gain the user's attention and allow them to better distinguish between the elements of Portio Pass. Repetition, alignment and proximity determined how every element in every section was chosen and grouped.

3.4.1 Tools and Components Used For this Prototype

This prototype is an iOS application written with swift 5 and developed with Xcode Version 12.13. Portio Pass is developed as an iOS application because, as it stands, in terms of mobile OS, iOS is generally considered to be one of the more secure operating systems in mobile security. It has an extensive security framework that can be used to make Portio Pass as secure as possible [10].

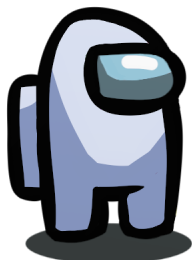
The back-end of this application was done with google cloud's firestore [11]. Firestore is a document-based NoSQL database which comes with integrated authentication and authorization libraries. This tool was picked in conjunction with Google Cloud [12] because it facilitates managing servers, creating a backend, and the issue of scalability. If Portio Pass is to grow in the

future and become accessible on multiple other platforms such as Android or on the Web, we simply have to work on the front-end.

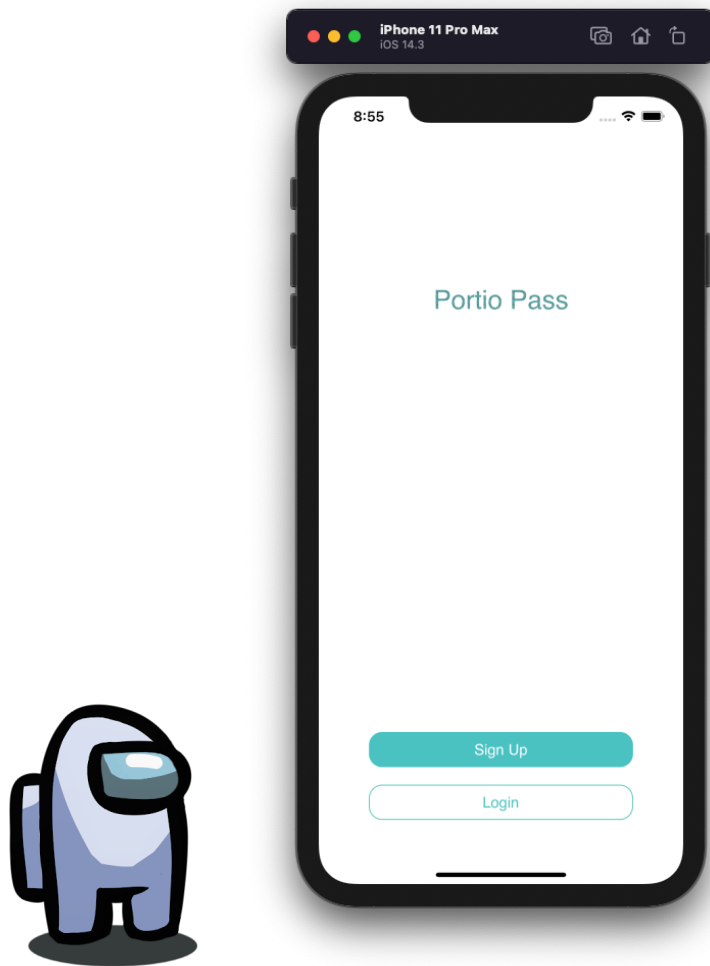
3.4.2 Storyboard and App User Interface



This is the launch screen of Portio Pass.

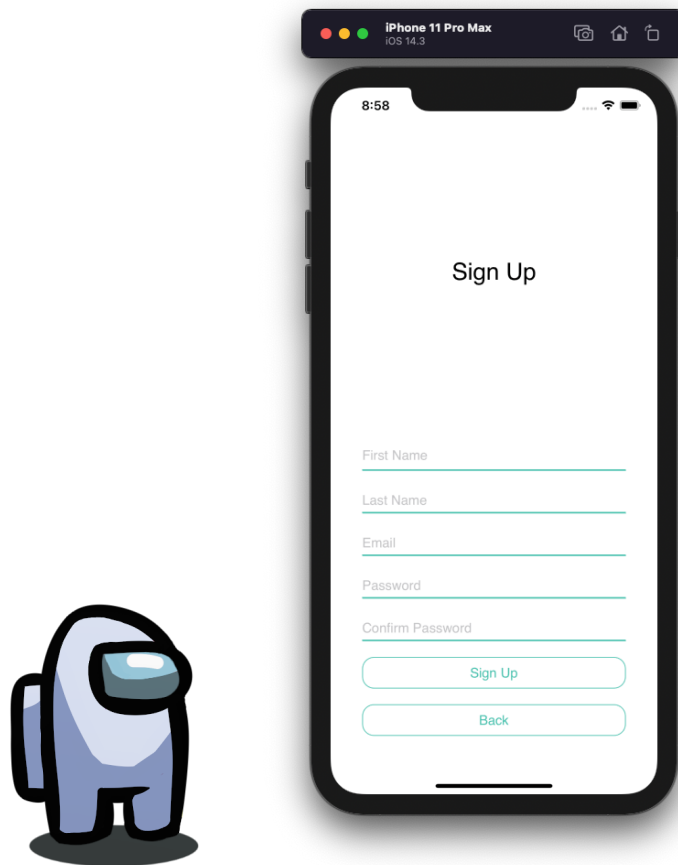


“Hi! I’ll be taking you through a set of tasks that can be done on Portio Pass”



“I want to make a Portio Pass account, guess I can just click on the Sign Up button!”

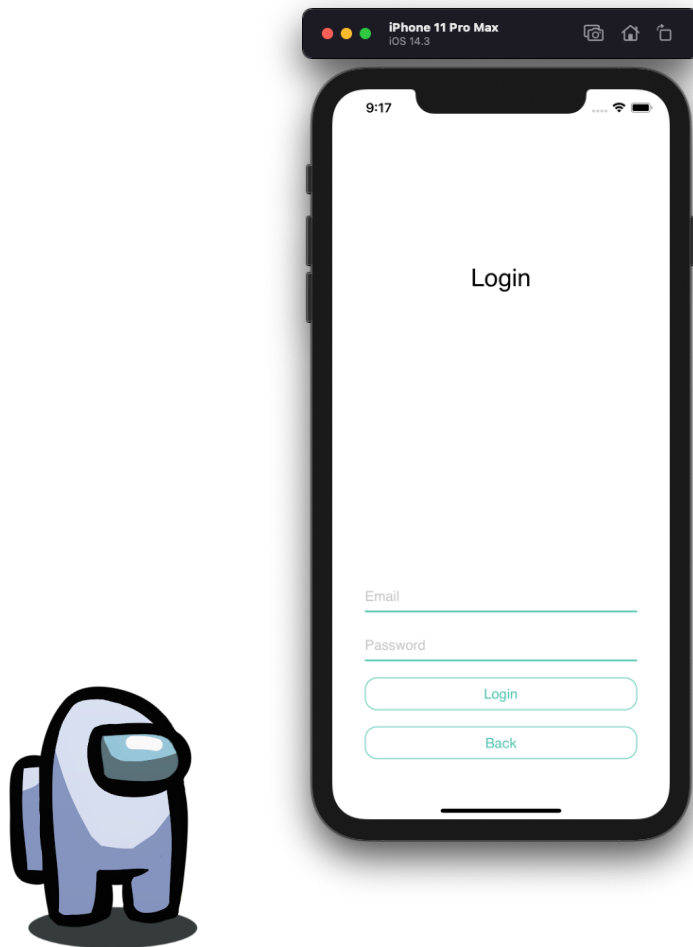
Upon the app opening, the user can Sign Up with another user if they are new or simply select “Login” to login the app.



“Oh nice, everything has a placeholder text for me. This definitely helps me know what I have to put in for each text field.”

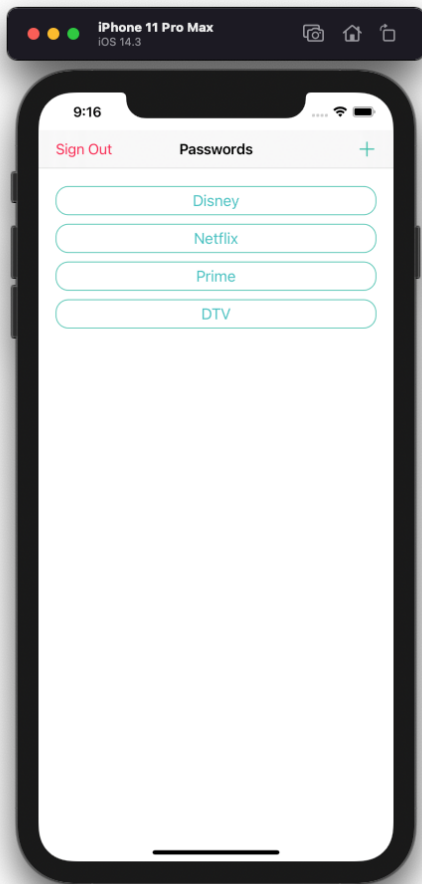
Here, the user can enter their name, last name, email (this will be used to login in the future), a password and a password confirmation. The password is set up to ensure the user can't create a weak password. They may either use apple's auto-generate password feature or create one themselves. The app will display an error message in red if the user doesn't follow the password policy.

Password requirements include: 1) Length must be at least 10 elements 2) There must be an upper-case character 3) There must be a lower-case character 4) There must be a special character.



“Dope, I can login super easily here. The error messages this page gives me are very clear! I know exactly what is wrong if the app fails to log me in. It also helps that those error messages are in red. Really captures my attention!”

This is the login page. To login, the user simply enters their login and password to get authenticated. If there is an issue, the app displays an error message to the user to let them know of the issue. Here, the log is descriptive enough for the user to understand but doesn't give any extra details in case there is malicious intent from someone who might be watching the logs.

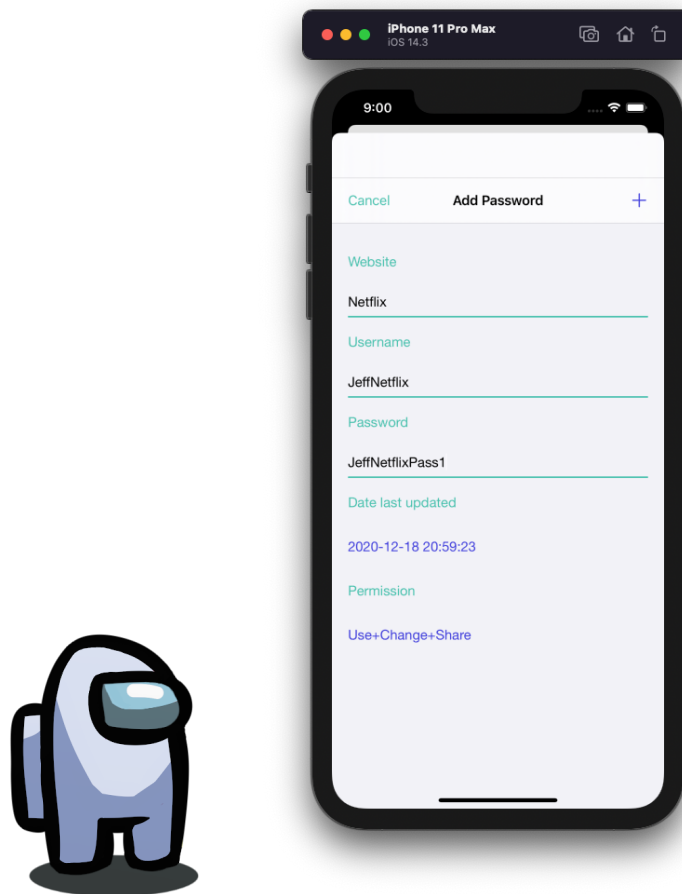


This is the landing page. On the top left, users can Sign Out. The colour picked for this button is red in order to make it more visible to the user. On the navigation bar at the top, the name of the page is written. Any new page the user enters in will have the name at the very top. This is to ensure that the user is never lost in the app. To add a new account entry, the user simply needs to click on the add button at the top “+”. If the user wants to share an account, they will have to tap on one of the accounts in their password list.

Last but not least, the buttons for the password list were made to be bigger/ wider to enable users to not mis-select the wrong button and accidentally share the wrong account with the wrong user. This was done so that the app is usable for people who have difficulty with element selection (ex: if they have larger fingers this might be harder).



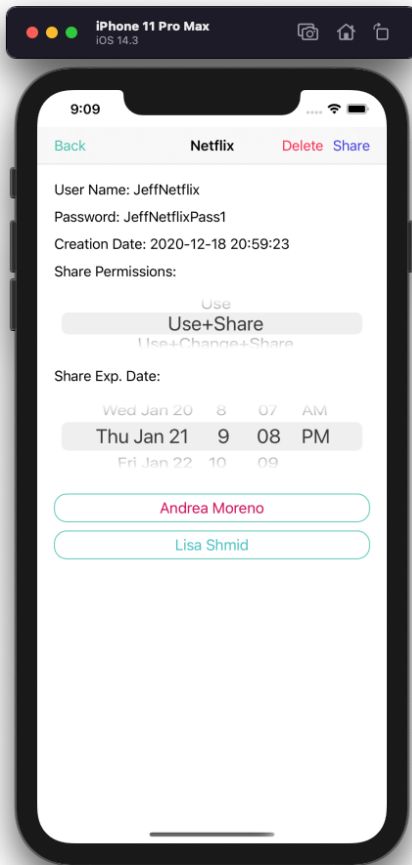
“Let me click on the add button on the top right to add a new account!”



“Oh, so the only thing I have to do here is to enter the name of my application, the username, and password. Looks like it also doesn’t allow me to add the account unless all three fields have text in them. This is good error checking as it prevents me from accidentally adding an account without a username.”

This is the “Add password” page. As mentioned in the previous page, the title is at the nav bar to let the user know exactly where they are in the app. The cancel button is placed on the left side where the SignOut button on most apps is usually placed. This makes it more familiar to the user and makes the app more usable for them.

On the right side, there is the add button. It will be disabled (the colour will be grey) unless all fields are filled. Lastly, the account ownership is set to the highest level for the owner.



Firstly, the title of the nav bar suggests that the current account selected is “Netflix”. The back and delete buttons are to the left and right respectively; again these buttons were placed in their respective positions to be more familiar for the user.

In order to share the account, the user needs to select a permission from one of: use, use+share, and use+change+share. By default, the permission is set to use. This is to make it easier for the user since, most of the time, the user will not be granting full privileges to the recipient.

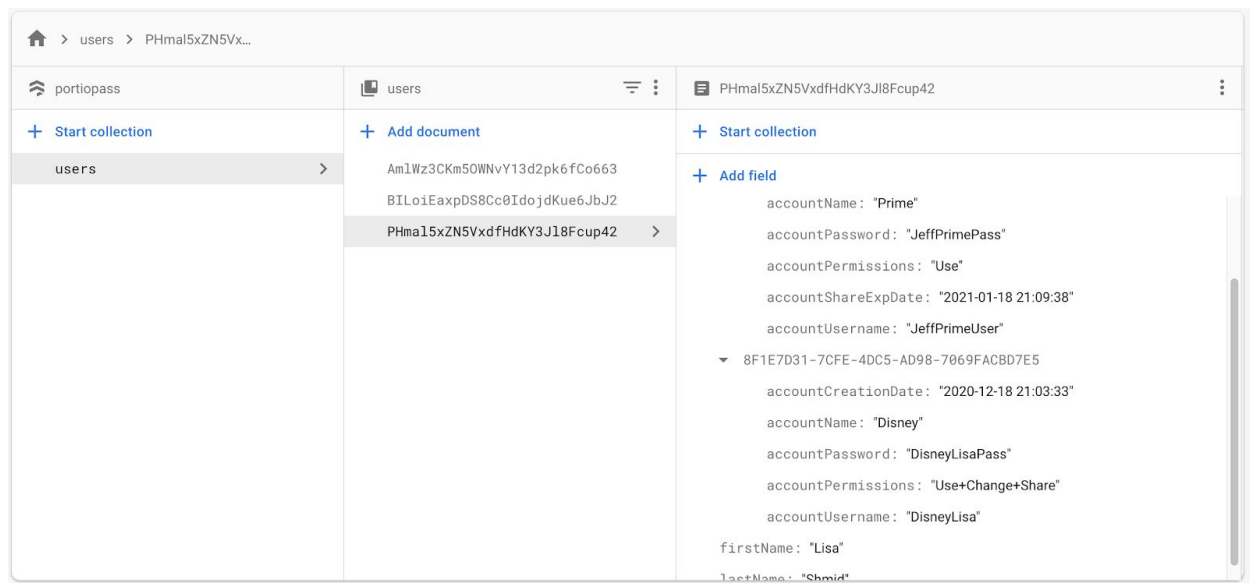
The expiry date here is a date picker to allow the user to simply pick a date and time for expiration. Not allowing for user inputs here will lower the risks of error, code injection, and invalid data.

Last but not least, the user is able to see the selection of authenticated users only.



“I can finally share my password with Andrea. Looks like all I have to do is select the share permissions, the expiry date, and her user button!”

3.4.3 A Peek into Cloud Firestore



The image above is taken from Cloud Firestore and is how the data is represented in the cloud. There is a collection of “users” which includes a set of user documents. Every user is represented by their unique UID. This facilitates identifying and authenticating them. Additionally, every user has their own set of variables. As the image shows, every account the user has in their database is represented as a nested dictionary.

- Collection of users
 - User
 - Dictionary of accounts : [key = Account UID, Value = Account object as a dict]
 - Value
 - [accountCreationDateKey, accountCreationDateValue]
 - [accountNameKey, accountValue]
 - [accountUsernameKey, accountUsernameValue]
 - [accountPasswordKey, accountPasswordValue]
 - [accountPermissionKey, accountPermissionValue]

- [accountExpDateKey, accountExpDateValue]

☰

⋮

PHmal5xZN5VxdfHdKY3Jl8Fcup42

+

Start collection

+

Add field

▼

Accounts

▼

4ED40F2D-85FC-42CC-8E2E-949688E53B6E

accountCreationDate: "2020-12-18 21:01:11"

accountName: "Prime"

accountPassword: "JeffPrimePass"

accountPermissions: "Use"

accountShareExpDate: "2021-01-18 21:09:38"

accountUsername: "JeffPrimeUser"

▼

8F1E7D31-7CFE-4DC5-AD98-7069FACBD7E5

accountCreationDate: "2020-12-18 21:03:33"

accountName: "Disney"

accountPassword: "DisneyLisaPass"

63

J2

42 >

3.5 Technical Details of Portio Pass

3.5.1 Firestore Authentication

Firestore's authentication library allows us to authenticate Portio Pass users with Email and Password based authentication. Using the library, Portio Pass has assigned every user a unique UID and a token which is collected by Firestore Authentication upon user login[15]. Within the firestore also, Portio Pass has the ability to set its own set of rules for authentication if it wishes to later on. This is definitely a feature worth looking into for the future for making Portio Pass more secure.

3.5.2 Firestore Server-Side Encryption

Apart from Simplicity, one of the reasons Google's Firestore Cloud was chosen for this project is it's server-side encryption abilities [13].

All data that travels over the internet during any read and write operations for the app is done VIA the Transport Layer Security (TLS). Furthermore, when writing data to disk, every object and metadata is automatically encrypted under 256-bit AES[14]. Currently, no one has been able to break 256-bit AES. Moreover, every encryption key that is used for AES is itself encrypted with a regularly rotated set of master keys.

Chapter 4 Conclusion

4.1 Challenges Faced

This chapter will go into detail about challenges faced during the design and development phase of creating this project.

4.1.1 XCode Challenges with Portio Pass

One of the biggest challenges of this project was getting familiar with XCode Interface Builder and Swift. There are many non-intuitive intricacies when it comes to dealing with Interface Builder.

One of the issues that stood out to me regarding Interface Builder was how unintuitive is to drag and drop elements from the user interface into the code. Apple created this as a way of connecting interface element nodes to the code. In practice the system is extremely fragile and easy to break. If you're not careful you can easily break the interface builder by breaking relationships between elements in the UI and the code. The error response from Xcode isn't always very obvious about this either, so it might take hours trying to debug this.

Another issue with xcode is how they handle their constraint system; there is no easy way in the interface builder UI to help the user deal with constraints. Interface builder permits users to add multiple conflicting constraints to the same elements, causing the Xcode to break.

4.1.2 Encryption Challenges with Portio Pass

Currently, the biggest challenge I'm facing is that all Portio Pass passwords are stored in plaintext. This is a huge security risk as, if the server owner's personal account is compromised, the attacker can view all account passwords for every user that is using Portio Pass. There needs

to be more research done on this to see what the best way is for encrypting passwords that are shareable.

4.1.3 Cloud Firestore Challenges with Portio Pass

One of the issues that arose with Firestore was deleting an account that had reached its share expiry date. I implemented a Google Cloud Function to iterate through all accounts and delete the expired accounts. The issue I faced however was that although Google Cloud Function returned no errors, it would not apply the changes to the database. This is important to address as a user will still be able to see credentials that they shouldn't have access to.

4.2 Portio Pass Recommendations for Future Work

List of recommendations to address for Portio Pass:

- Send in-application notifications to the account owner, notifying them to change their password if a recipient's password has been expired. This is important as no matter what we do, the recipient will always be able to see the password.
 - A password can't be shared as a ciphertext to the recipient.
- Enable Portio Pass to import users' contacts from their phone who are also subscribers to Portio Pass. This way, instead of listing all users in the users collection, only the users from the user's contacts will be displayed.
- Have a Privacy Policy disclaimer that enlists everything Portio Pass will have access to, everything it can share, and so on. This will enable transparency between the users and Portio Pass.
- Portio Pass must store passwords as ciphertext in the cloud.

- Portio Pass should have another page that displays to the owner, all users that are sharing a particular account, in addition to their privileges, and the share expiry date.

4.3 Summary

Although Portio Pass is just a proof of concept and still needs a lot of work, I believe it definitely shows promise in facilitating password sharing between individual users. Having an application that is designed with the two simple goals of being usable and secure, and whose only purpose to facilitate secure sharing will result in fewer compromised accounts for all users.

Bibliography

- [1] <https://www.statista.com/topics/1594/streaming/>
- [2] <https://flixed.io/complete-list-streaming-services-2020/>
- [3] <https://www.cnn.com/2020/12/13/media/streaming-price-disney-netflix/index.html>
- [4] <http://www.convergenceonline.com/downloads/2020OTT.pdf>
- [5] <https://www.welivesecurity.com/2020/11/11/why-you-should-keep-netflix-password-yourself/>
- [6] <https://www.lastpass.com/>
- [7] <https://1password.com/>
- [8] <https://www.usenix.org/system/files/conference/soups2014/soups14-paper-stobert.pdf>
- [9] https://www.scienceopen.com/document_file/0f2d1cb1-271c-4f3a-82a4-f5e0c91ea448/ScienceOpen/BHCI-2018_Merdenyan.pdf
- [10] <https://developer.apple.com/documentation/security>
- [11] <https://firebase.google.com/docs/firestore/security/overview>
- [12] <https://cloud.google.com/>
- [13] https://cloud.google.com/firestore/docs/server-side-encryption?fbclid=IwAR2yzYJnxFHT2cNebW4lifmvRwAhWZEXHholpTjtn_LR9SbJ8KbY5vaC_6U
- [14] <https://www.solarwindmsp.com/blog/aes-256-encryption-algorithm>
- [15]

