Secure File Vault

Final Report

Author: Jack Hooton Byrne

Student ID: C00230173

Project Supervisor: James Egan

Recipient: Institute of Technology Carlow

Date: Friday 30th April 2021

Abstract

Secure File Vault protects individual files or data by encrypting them using different encryption methods. When the user wants to view the files, they login into the cloud to view the files which have been decrypted for the user. The cloud is a Raspberry Pi which stores the valuable data.

Have you ever run out of storage space on your laptop or computer and had to invest in external memory? Have you ever thought about any other options?

Unfortunately, this has happened to me and has led me to create my project. Many people each day run out of storage space on there on devices. But not many people think about using cloud storage to store their data. People are afraid of storing their data in a cloud because they don't know where the cloud is. A cloud is just a set of servers that are stored on an offsite location. Users are also afraid of how their data is being stored. Recently there have been many cyber-attacks on cloud systems. According to the 2020 Trustwave Global Security Report, attacks on cloud services have doubled from 2019 and have accounted for 20% of investigation incidents. Cloud systems are now the third most targeted environment for cyber-attacks. The purpose of my project is to create the most secure cloud system on the market.

Table of Contents

Introduction

The following document will provide a detailed description of the overall progress and the final version of the Secure File Vault application. The product description will cover a detailed outline of the features and functionalities that the application aimed to satisfy the user.

Project description

Secure file Vault is an android application that allows a user to upload an encrypted file and then stored it securely on a separate database. The database which stores the encrypted files will be stored on a Raspberry Pi. When the user wants to view the files, they select the file and be given two options download or delete. At this stage, the file will be decrypted. When a user wants to register an account, it is done by using XAMP. The email and password are encrypted to the highest security standard. The application will also be secure against many common android Application vulnerabilities.

System components

Secure file vault consists of two significant components hardware and software. The hardware components are the raspberry Pi and the hard drives for the cloud.

Hardware Components

Raspberry Pi [3]: The Raspberry Pi will be used for creating the cloud server. To create the cloud server, the hard drives will be connected through the USB slots. The image below shows a Raspberry Pi and how it can connect to the hard drives.

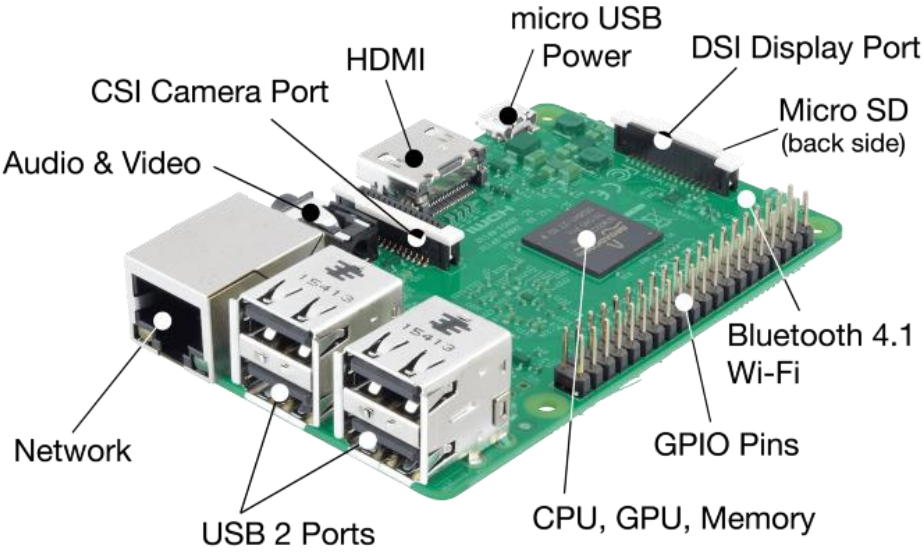

Figure 1: Raspberry Pi

Hard Drives: Multiple hard drives will be connected to the Raspberry Pi to form a server for the cloud.

Software components

Java Application development will be used for making my application for the cloud. I will be using many APIs to form the security aspects of the application. I will also use APIs for the cryptography part of the application as they provide outstanding security and integrity for the user's files and credentials. I will also use strong security techniques to prevent common vulnerabilities.

Application user interface

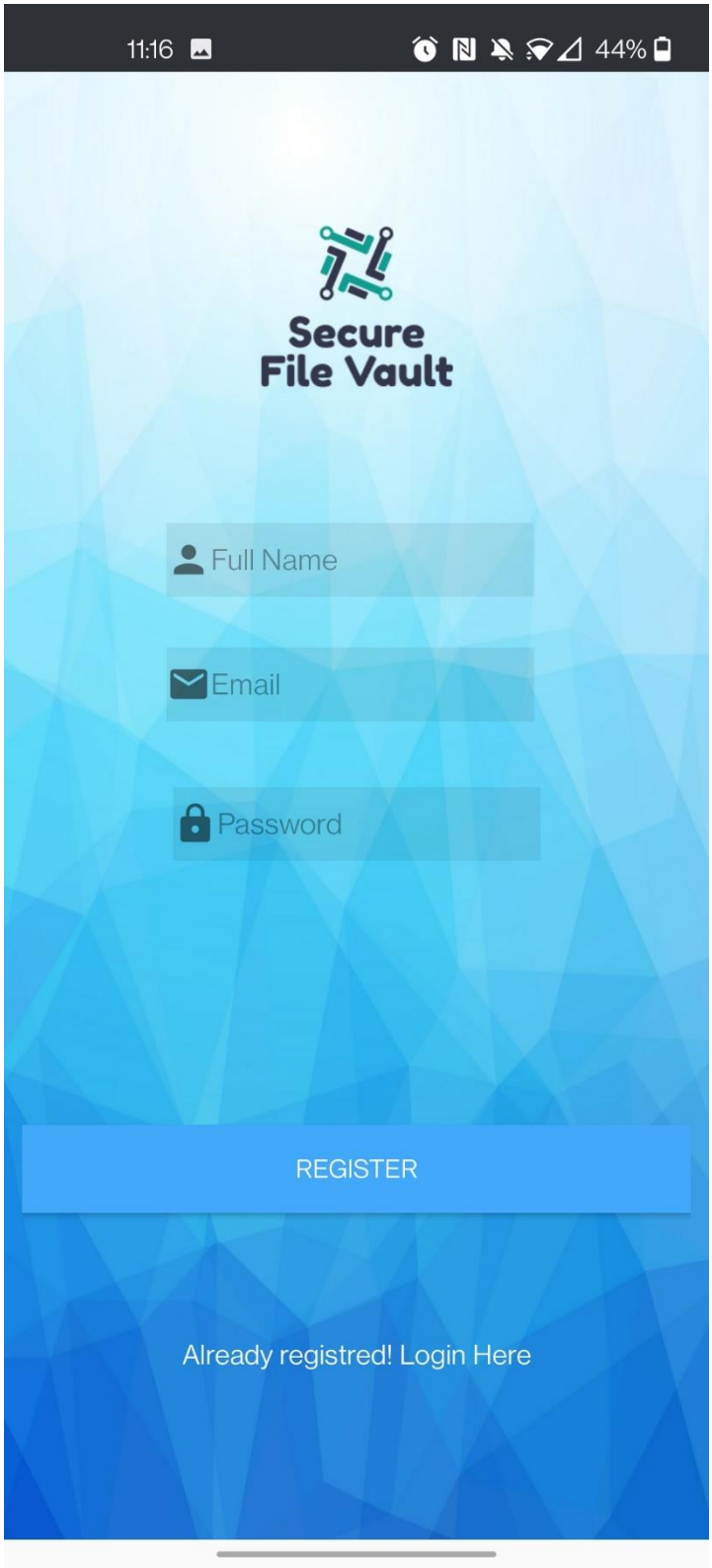

Figure 2: Secure File Vault – Registration

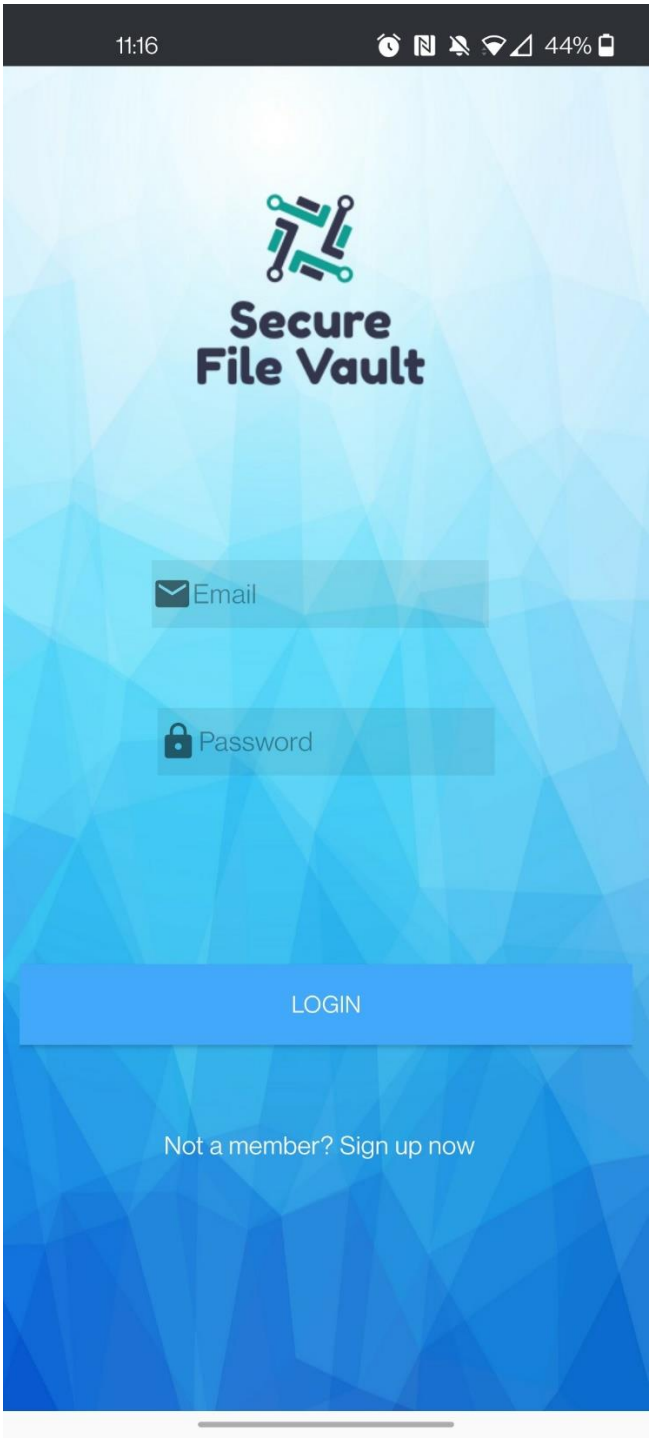

Figure3: Secure File Vault - Login

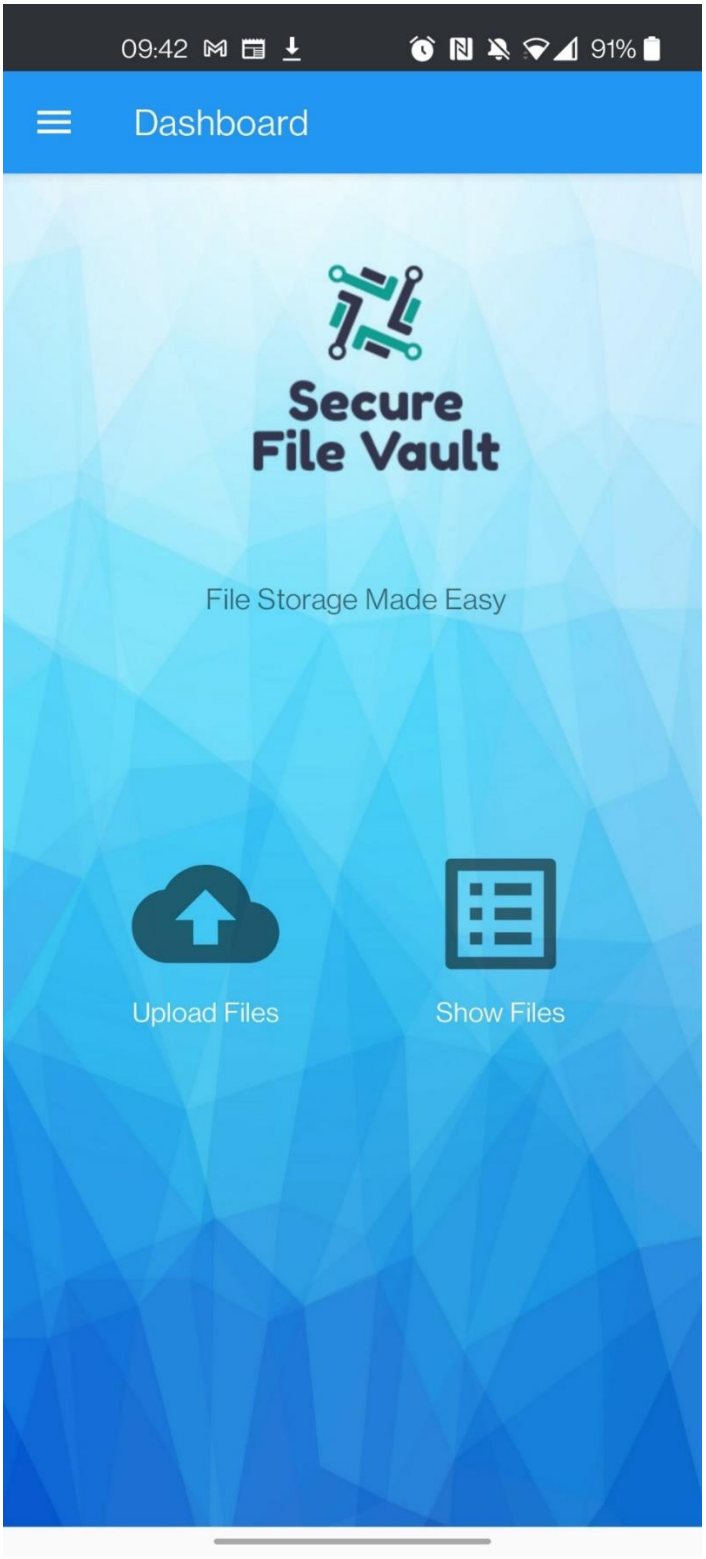

Figure 4 Secure File Vault - Dashboard

The Dashboard page gives the user easy access to the application's primary functions, such as the file upload and show files.

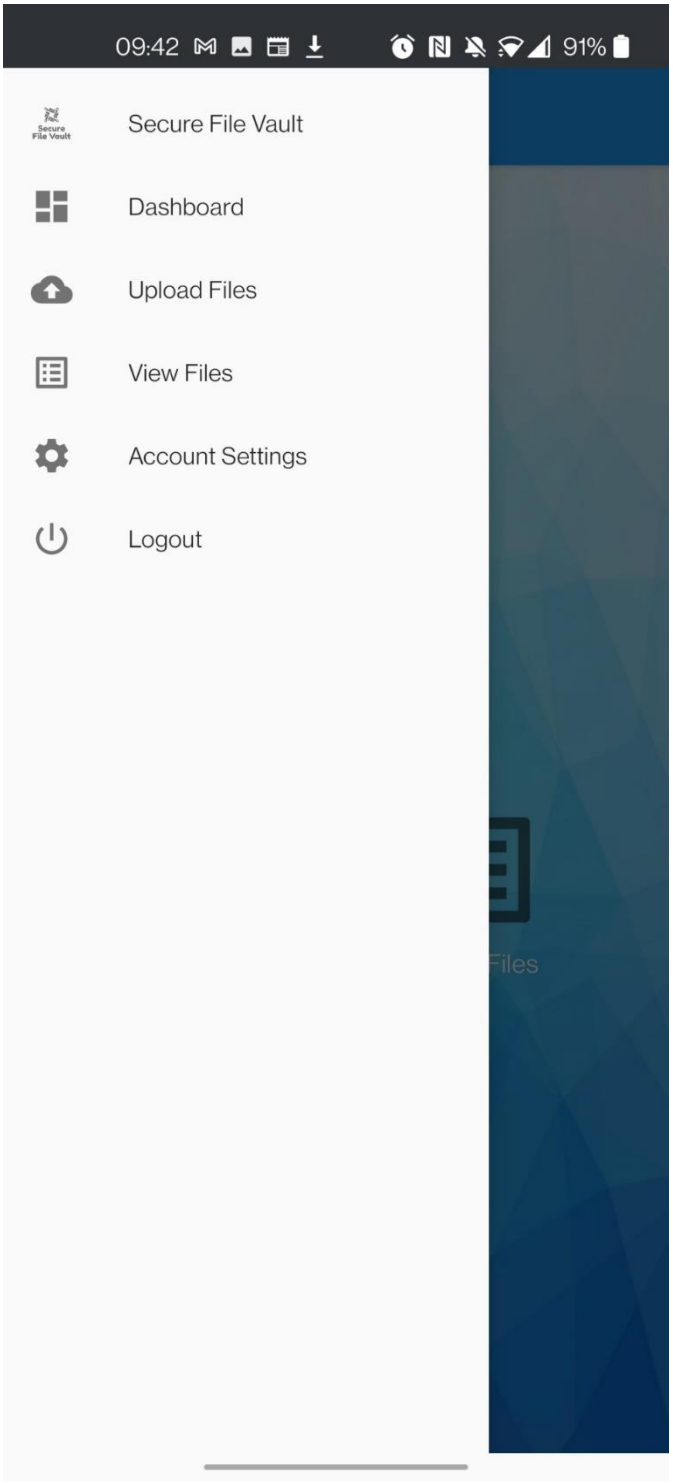

Figure 5 Secure File Vault – Side Navigation

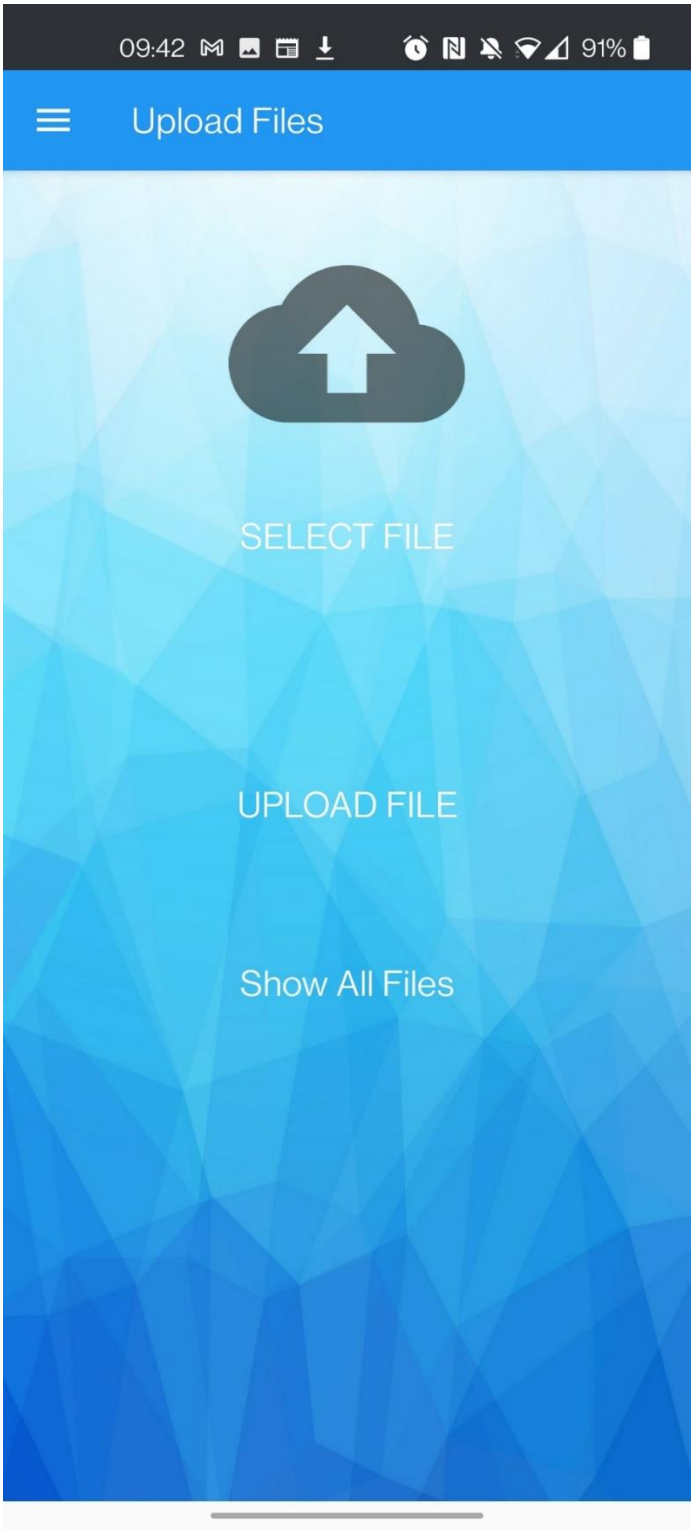

Figure 6: Secure File Vault – Upload Files

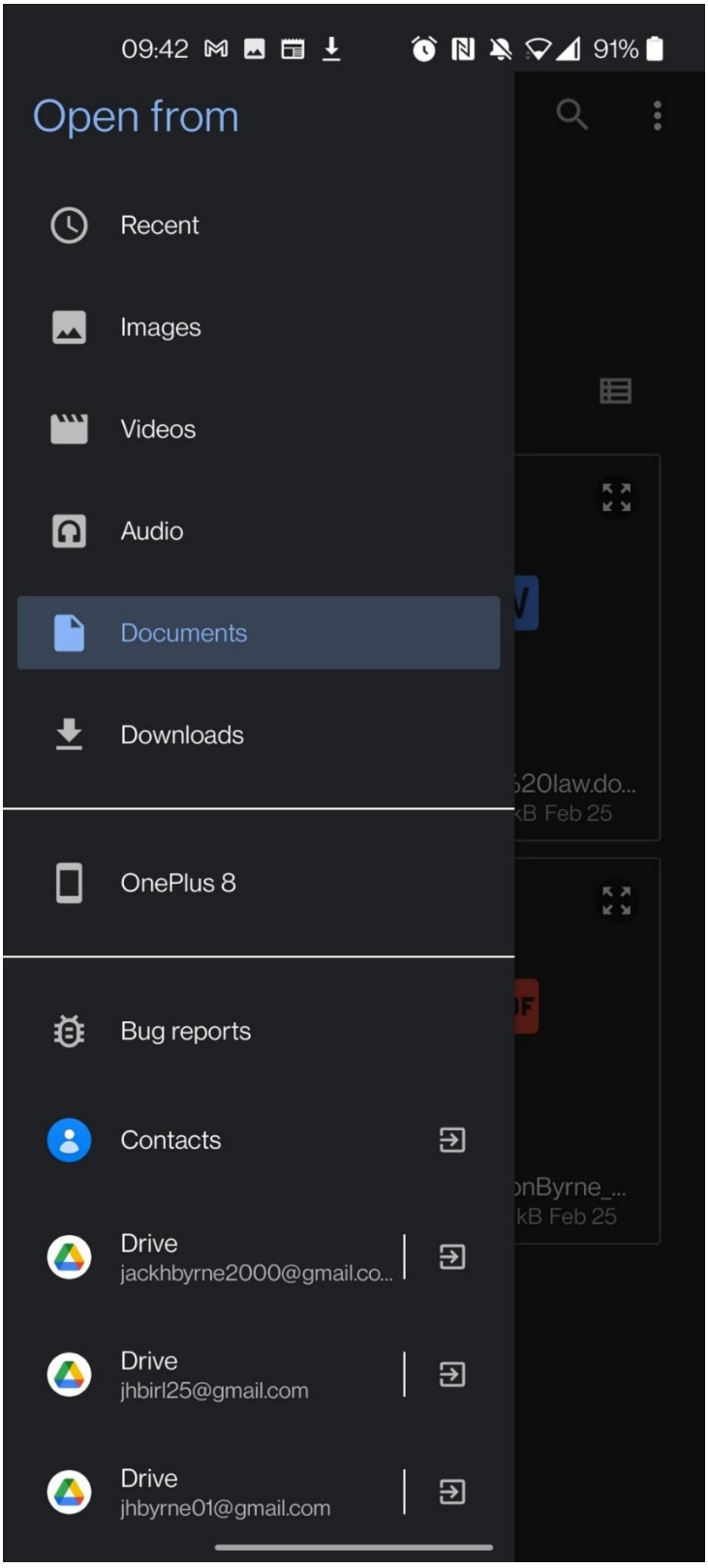

Figure 7: Secure File Vault – File Selection

Once the user clicks on select file, they are brought to the device's file manager application. From there, the user can select any file and upload it to the cloud.

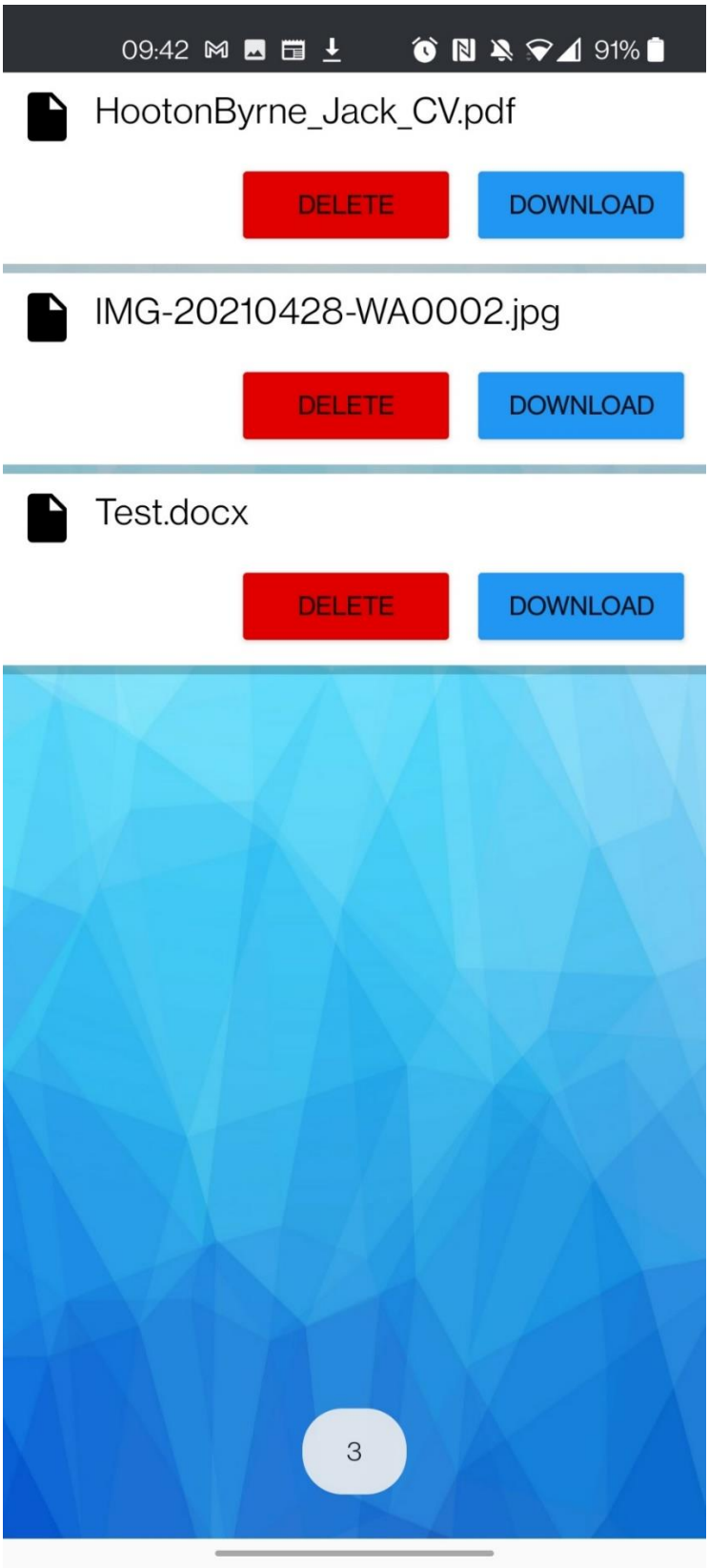

Figure 8: Secure File Vault – Show Files

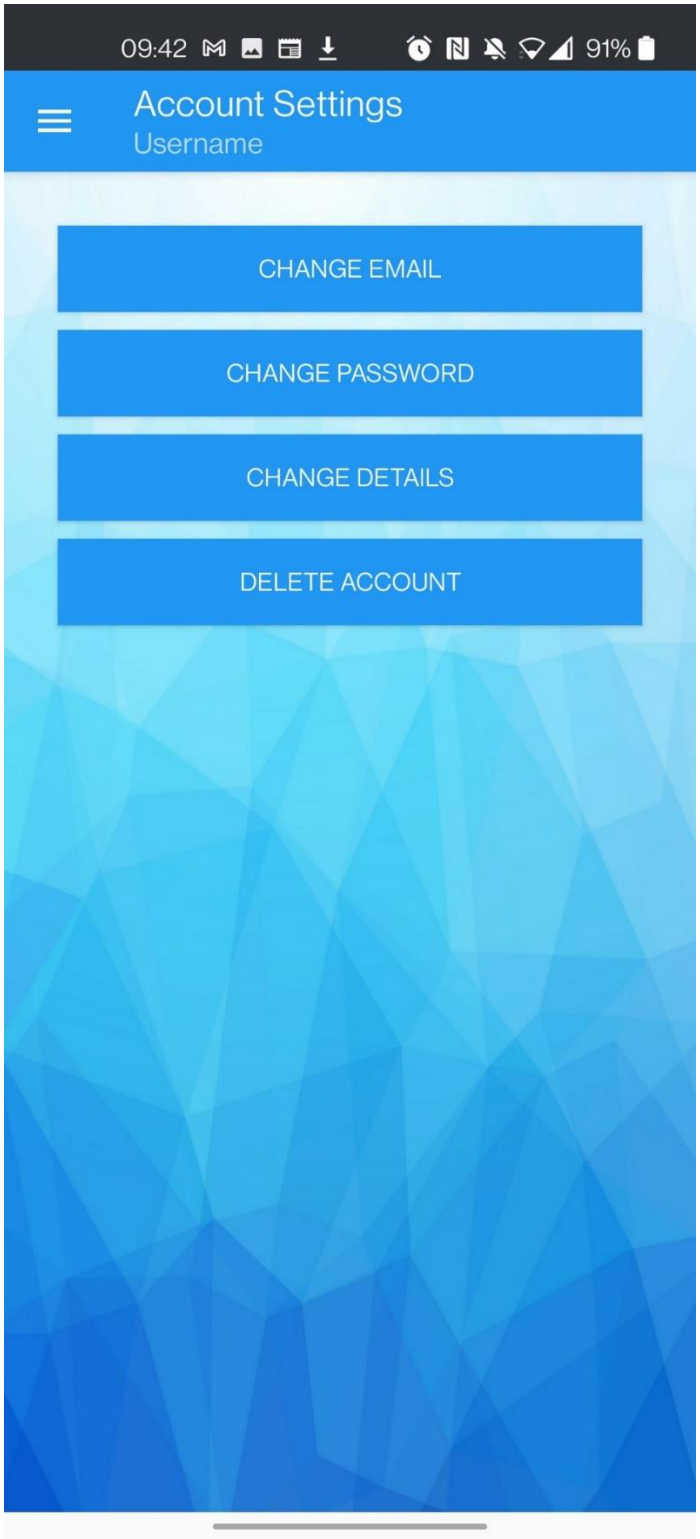

Figure 9: Secure File Vault – Account Settings

Conformance to specification and design

In this part of the document, I will contrast the submitted proposal and specifications with the project today. Overall, the final project submission conforms almost precisely to the initial proposal and subsequent final specification. All main functionality is provided apart from the decryption, which I will discuss later in this report.

Initially, at the start of the development of the project, I was discussing with my project supervisor James Egan, and we found out that since this is an android application, full disk encryption will not be possible as of android nine it is impossible to have full-disk encryption as the feature no longer exists on android Phones. I also discussed that the application should be for personal use as there is nothing like that on the market now. Since this is for personal use only, there will be no need for an Admin account. The common vulnerabilities the application protects itself from were based on OWSAP Top 7 mobile vulnerabilities.

Learning Outcomes

In the past 5 to 6 months of this project, there have been many learning outcomes that resulted from researching and working on this project. In the beginning, I wanted to push myself and expand my horizons in terms of technologies I am familiar with. Since I am not the best at programming, having some experience with Java made me pick it. Using the Raspberry Pi was also challenging as I rarely use Linux. However, I believed that this was the best chance for me to learn new technologies.

Technical Achievements

As previously stated, I don't have much experience with Java and Raspberry Pi. Learning Android application development was a lot harder than learning how to use the raspberry pi. However, learning new technologies and becoming familiar with them is always challenging.

Java

Java was not too difficult to learn as I have learned the basics in college. For Android Application development, using Java was complex. I followed YouTube videos for the first two months to learn android studio, and once I felt confident in my capabilities, I started to develop my application.

Raspberry Pi

Although I had some experience using the Raspberry Pi, I still had trouble getting grasps of the technology. In the end, I feel very confident in using the raspberry pi.

Testing

Security Testing

Security Testing is a critical part of my final year project. As discussed in the Research document and Specification document, there are seven main common vulnerabilities to be aware of when developing an Android Application. One of the main vulnerabilities is Binary protection. This can only perform when the application is delivered through an APK file format. If my application were going to be on the Google play store, it would have to be an APK file. This exposes the vulnerability, one way to prevent this is to add a security feature that lets the application detect tampering and react accordingly.

The following vulnerability is insufficient transport Layer protection. Encryption is used for all authenticated sensitive communications. Also, all the user's data is encrypted and stored safely. SSL or secure sockets layer is used when the files are being encrypted and decrypted for the users, enhancing the security of the application.

Insufficient Authorization and Authentication are next to be tested. Insufficient Authorization results when an application does not perform adequate authorization checks. Secure File Vault has sufficient authorization by using correct authentication techniques. Also, when the user leaves the application, they can go back in and will still be signed in if they haven't reached the max timeout.

Cryptography-Improper Certificate Validation is the fourth test to be performed. HttpURLConnection has been used to secure the connection between the android device and the Raspberry Pi. This stops the certificate from being invalid and susceptible to malicious attacks.

Brute Force attacks are usually only performed on Login and Registration pages. To prevent this, each time the users must enter valid information, it is validated and can only be a certain length, such as the password is a minimum of 8 characters long and a maximum of 16 characters.

Session Expiration which has been mentation above, is a critical part of any application. The max time the user has on the application is ten minutes. Once the ten minutes are up, it is logged out securely and can log back in again.

SQL injection is the final test to be performed on the application. To do this, I entered a basic tautology in the login page to gain access to the application email = 'OR 1=1 and the password = "". If this email and password condonation grants me access to the application, SQLi can be performed. Luckily enough, I have the correct precautions in place, and this can't be achieved. I also tried to use email = 'OR 1= 1 "@gmail.com" and password = "" shown below as the application only allows valid email addresses. This, however, still be not work as I used prepared statements in the PHP code for the login and Registration.

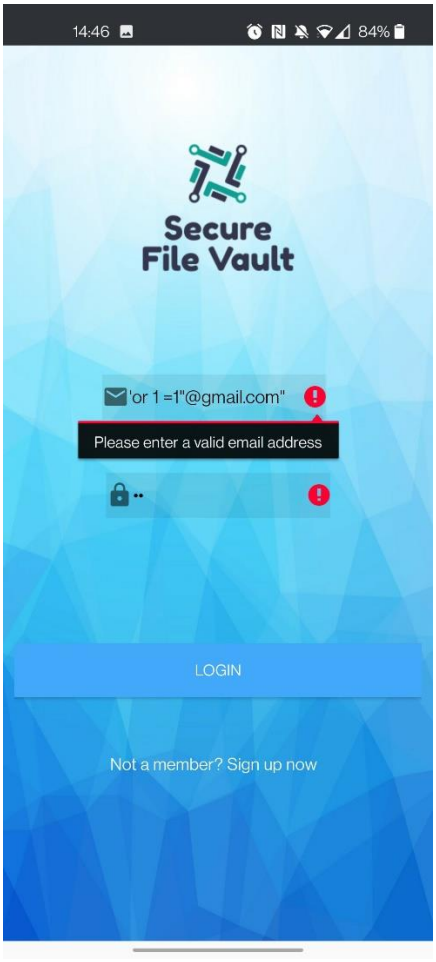

Figure 10: SQL Injection

User Testing

User testing was performed on the application to see how other people use the application. I ask my family members to test the application for me. They gave me some helpful feedback on the application and noticed one flaw that was when the user is on the account's settings page, they are unable to log out. The feedback was taken on board, and the changes have been made to the application, making it more pleasing for the user.

Project Review

This segment of the document will review aspects of the project that were successful and Unsuccessful.

Positive Aspects/Aspects Achieved

This entire project I would consider a great success. Most of the core functionalities stated at the start of this document were completed, except for a significant part of the application's decryption of the files stored on the database. The user can log in securely and upload files to the Raspberry Pi

database. The application is also sure against common mobile vulnerabilities. A month before the final submission deadline, I thought I wouldn't get a finished application because of the lack of progress that was being made and how much time I had to put into other college assignments and CA's. Nonetheless, I was able to finish the application and got most of the main functionality working correctly on the application.

Aspects Not Achieved

As stated above, I unfortunately ran out of time in the end and wasn't able to get the decryption working correctly. This is a massive part of the application as none of the files can be stored securely. The encryption works perfectly, but the decryption doesn't. Also, since I had invested so much time into trying to get other parts of the application to work, I could not add a change password or email to the application.

Aspects gone wrong/Problems Encountered

During the development of Secure File Vault, I faced many problems. One of the biggest problems I faced would have been connecting the raspberry Pi to the android application to allow the files to be transferred over to the application. This set me back, and it was looking likely that I could never get this to work. I kept pressing at issue, and eventually I was able to get it to work. It was a network security error and where I had to allow clear text traffic to be permitted on the android device. Also, another issue I faced the decryption for the files was not working, and I ran out of time on this aspect.

Things I would change

If I had to start developing this project initially, I would change quite a few things that would make the whole process better and make the application better.

Time management: One of the main problems I faced while creating this application was that I always seem to have no time to work on the project as I always had so many assignments and CA's due. I started in December I started developing the project, from the start, there was a lack of consistency from the development side. I would begin a task, and it would take me a while to finish it as I had to multitask with all of the assignments I was getting. I also spent a considerable amount of time trying to get the application to upload files to the Raspberry Pi. At the start, I wasted time on the design when I should have been developing the application. If I had to restart, I would focus more on the application's actual functions and focus more on finishing a task before moving onto the next task.

More Research: Programming one is of my least favourite subjects I've had in college, but Android Studio has made me like programming more. However, I had many issues and had to research a lot to solve the problems. If I had to start over, I would have followed a proper online course for android app development and researched more and android app development.

Future Features

As mentioned above, the decryption functionality does not work. As a future feature, decryption would be an excellent choice. Also discussed above, for the user a change password and change email function was supposed to be implemented but unfortunately wasn't. This could be added as a future feature. Another future feature would be to add more security to the application as technologies are constantly changing, and security needs updating every year, or anytime there is a new vulnerability. Another future feature was discussed with my supervisor, and it was to add a website application to make it easier for the user to upload and download files. This feature would also allow full disk encryption to be performed as it is currently unavailable with android nine and above.

Acknowledgments

There are a lot of people that I would like to acknowledge in this section, starting with my project supervisor James Egan for whose advice, guidance, opinion, and support have helped me so much through the whole process of the development of the application. Also, I would like to thank all of the lectures from the first year to this year for sharing your knowledge with the IT security and Cybercrime class of 2021. Finally, I would like to give a special thanks to Thomas Jordan and Patrick Alabi for their support throughout the process and their feedback on the application.