Secure File Vault
Design Manual

Author: Jack Hooton Byrne
Student ID: C00230173
Project Supervisor: James Egan
Recipient: Institute of Technology Carlow
Date: Friday 13th December 2020

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 that stores 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 detailed information on the elements of building the secure file vault and how each function is expected to behave. This design document will show the user an overview of the system architecture, the database design, and detailed use case and sequence diagrams.

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 make 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.

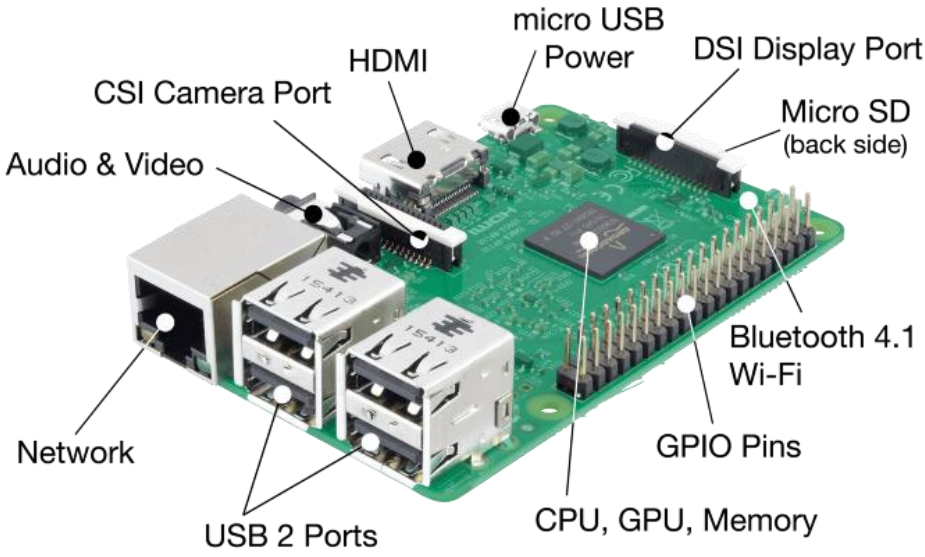

Figure 1: Raspberry Pi

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 robust security techniques to prevent common vulnerabilities.

User Interface

The android application should have an elegant design, be simple to navigate, and make it as simple as possible for the user to use the application.

Design of the Application

The following diagrams are prototypes of how the android application should look and what the flow would be.

Login Design

Figure 2: Secure File Vault – Login Screen

Registration screen

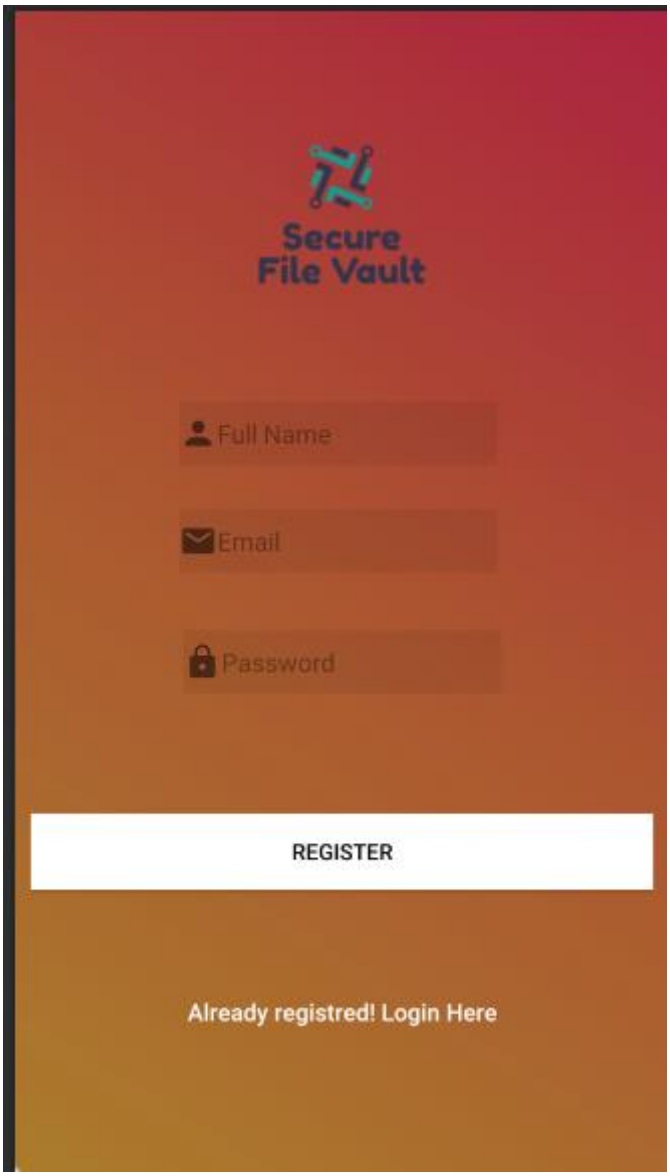

Figure 3: Secure File Vault – Registration Screen

Side Navigation Bar

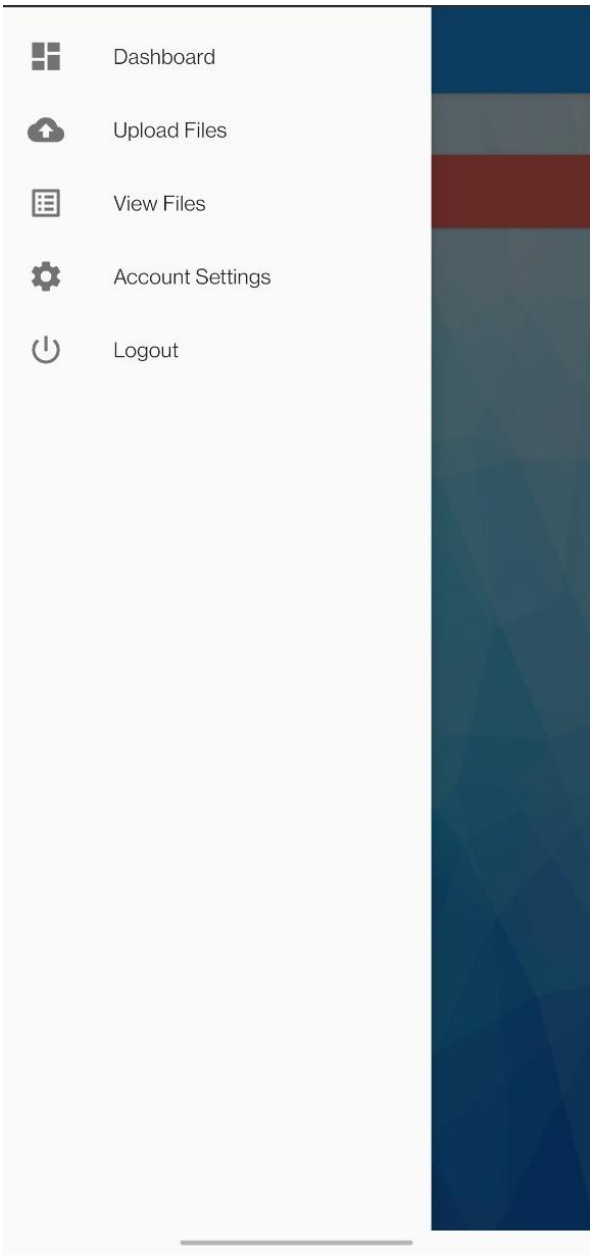

Figure 4: Secure File Vault – Side Navigation

Upload Files

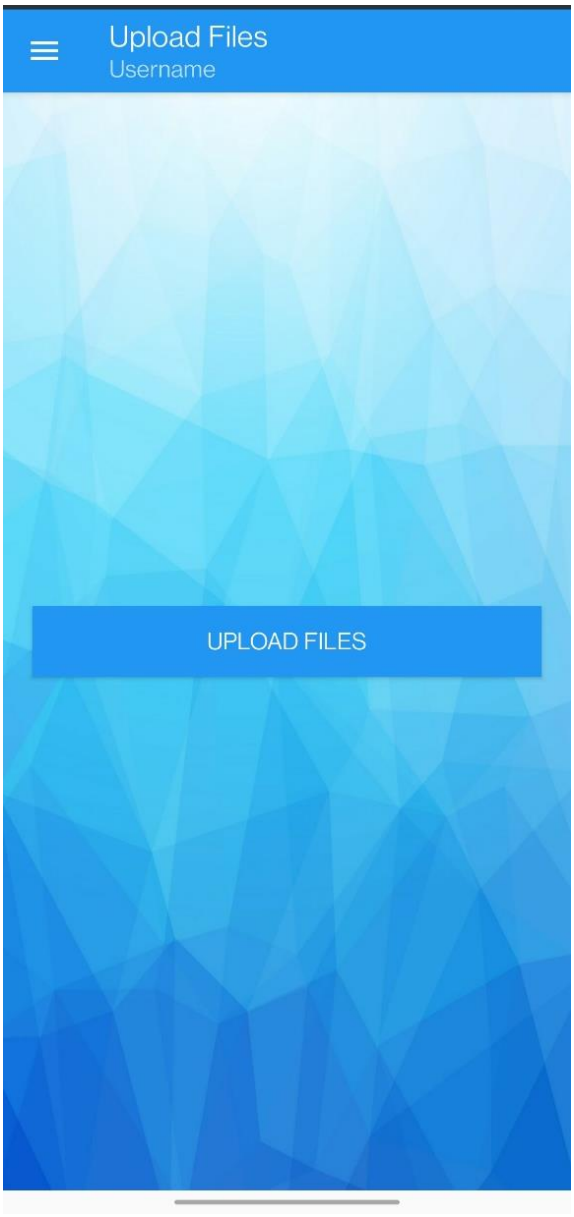

Figure 5: Secure File Vault – Upload Screen

View files

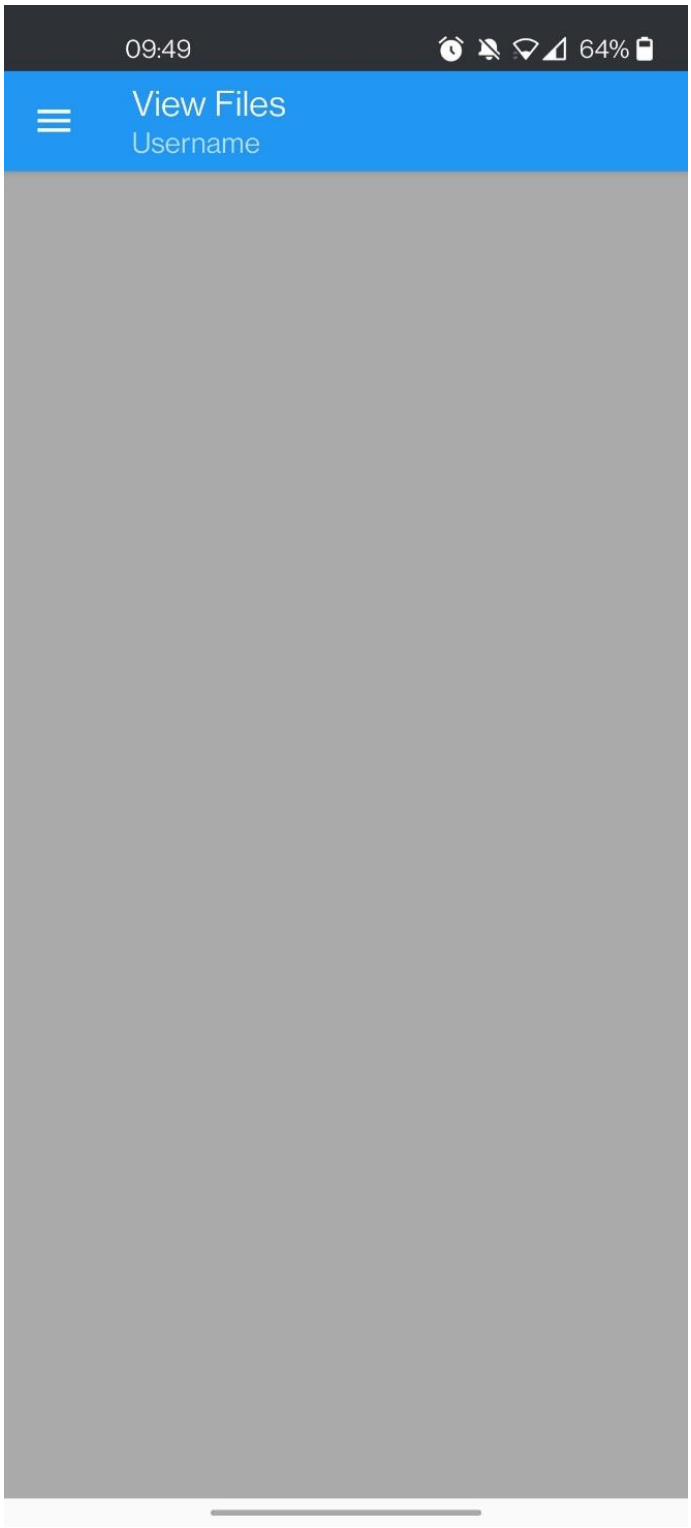

Figure 6: Secure File Vault - View Files

User Account Settings

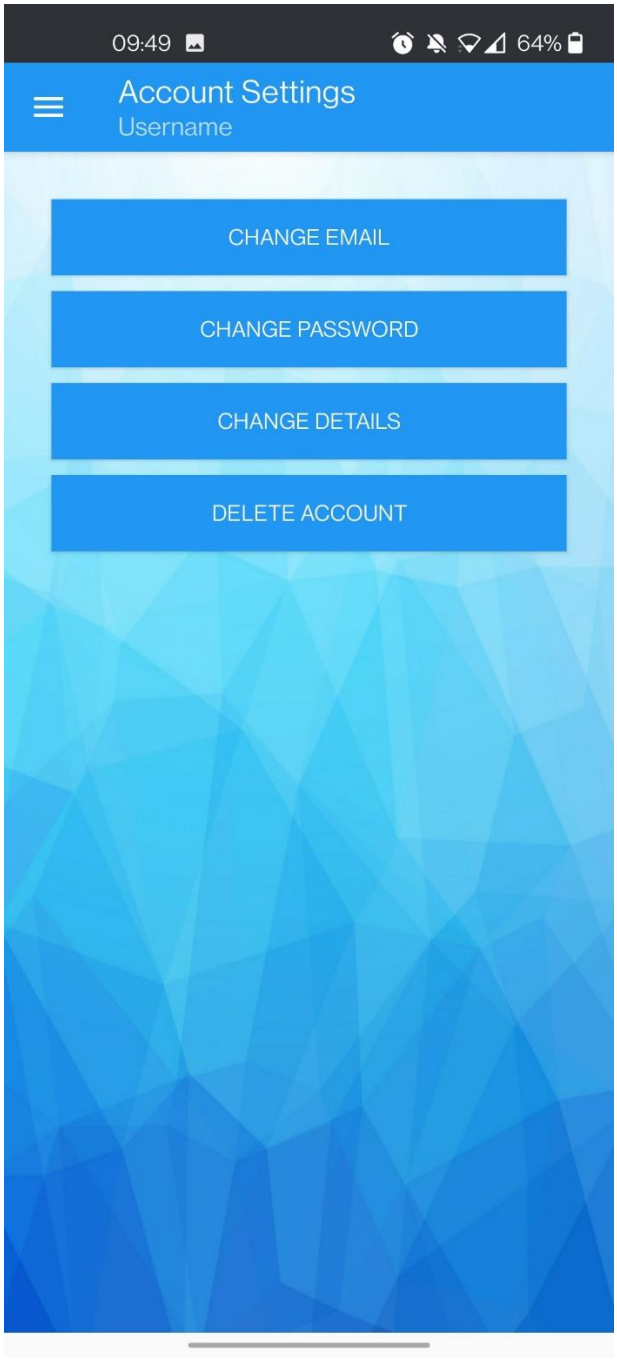

Figure 7: Secure File Vault – Account Settings

Database Design

The database for the system is created on raspberry pi using phpMyAdmin. The database is quite simple when there is only one table, the user's login table.

User Login Table

Database Name: SfvDB

Table Name: Sfv_users

Description: Used to store the user's login credentials

Table Structure and Sample data:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	unique_id	varchar(23)	latin1_swedish_ci		No	None			Change Drop More
3	name	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
4	email	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
5	encrypted_password	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
6	salt	varchar(200)	latin1_swedish_ci		No	None			Change Drop More
7	created_at	datetime			Yes	NULL			Change Drop More
8	updated_at	datetime			Yes	NULL			Change Drop More

Figure 8: Users table design

	id	unique_id	name	email	encrypted_password	salt	created_at	updated_at
Edit Copy Delete	1	6061a2a3a6bf88.02495303	Jack	Jack@gmail.com	VmvYcqyaax2xKhabuEoP4nMpAm8zNTU1YWQzMGE1	3555ad30a5	2021-03-29 10:49:23	NULL

Figure 9: Users table sample data

Detailed Use Cases

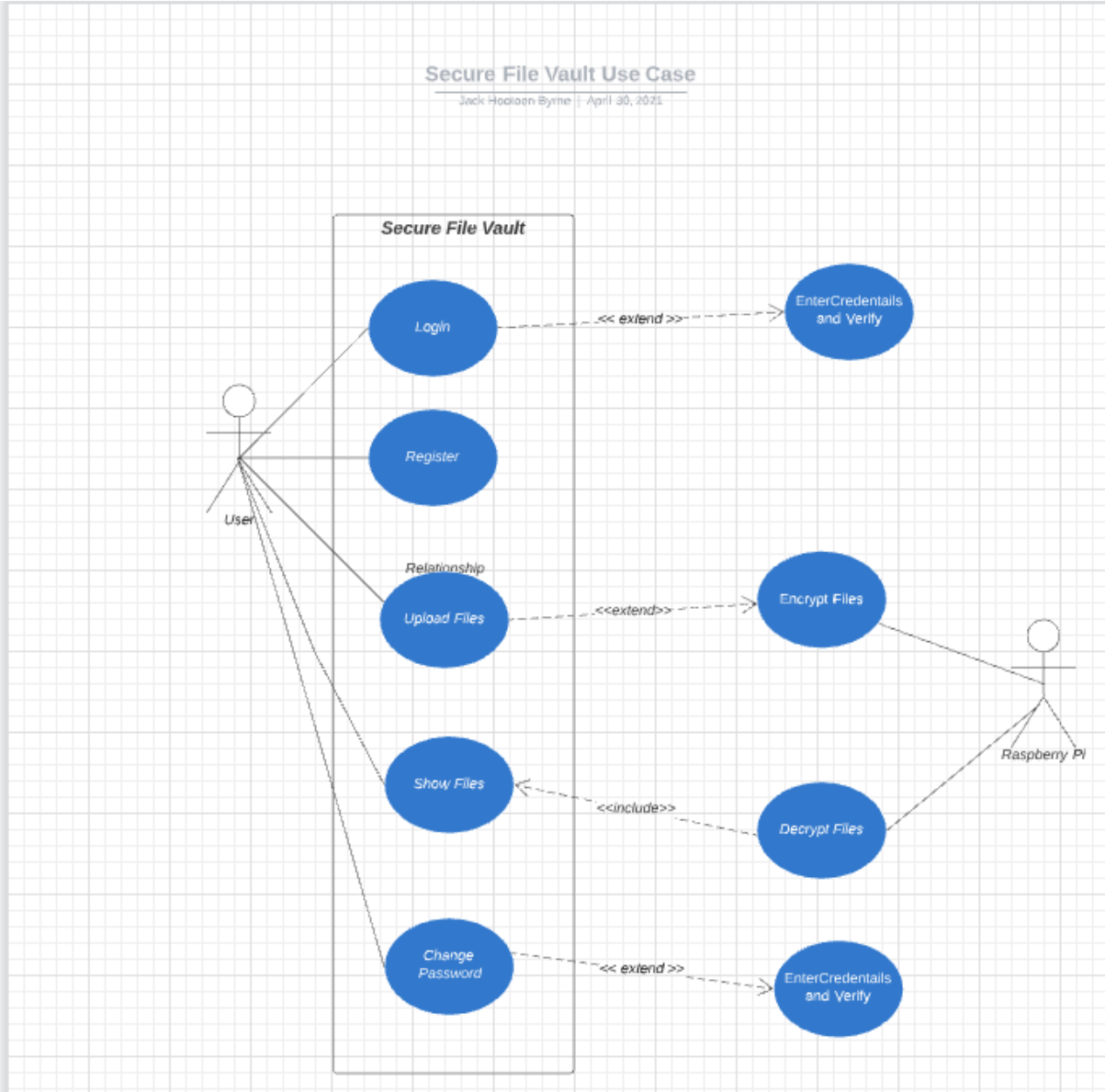

Figure 10: Detailed Use Case

Registration

Primary Actor

User

Preconditions

None

Success Guarantee

The user is now able to register on the app. Their Username and password are saved to the database.

Main Success scenario

1. The user downloads the app from the google play store
2. The user clicks the button to register
3. The user enters a valid username and password
4. If the Username is available and the password is strong enough, the user's account will be set up. The user's credentials will be saved to the database

Alternative flow

- The Username or password is not available. The user is prompted to try again with different credentials.

Application Login

Primary Actor

User

Preconditions

The user must have made an account by registering.

Success Guarantee

The user successfully authenticates with the app and can now log in.

Main Success scenario

1. The user visits the app.

2. The user clicks the button to log in.
3. The user enters a valid username and password.
4. If the Username and password are valid, the user can log in successfully.

Alternative flow

- The Username or password is not valid, and the user is prompted to try again with different credentials.

Authenticated user File upload and download

Primary Actor

User

Preconditions

The user has successfully logged in.

Success Guarantee

The user is now able to upload files and can also download the files they have uploaded to the cloud.

Main Success scenario

1. The user visits the app and logs in.
2. The user can upload any file to the cloud.
3. The user can download any file from the cloud to view it.

Alternative flow

- The user uploaded the same file twice and asked if they want to overwrite the old file.

Change Password

Primary Actor

User

Preconditions

The user has successfully logged in

Success Guarantee

The user is now able to change their password by entering their current password and entering the new password twice

Main Success scenario

1. The user visits the app and logs in.
2. The user clicks on change password.
3. The user enters their current password and their new password twice
4. The user will be prompted that their password has been changed successfully

Alternative flow

- The user will be prompted to try again if their current password isn't correct or if the new password is too weak.

System Sequence Diagrams

Sequence diagrams are an essential part of every android application plan. They are diagrams that show in detail how each aspect of the application is performed on the system. Sequence diagrams are time-focused. They offer the order of the interaction with the application visually by using the vertical axis of the diagram to represent what messages are sent and when.

(Paradigm, 2021)

Android App Registration Sequence Diagram

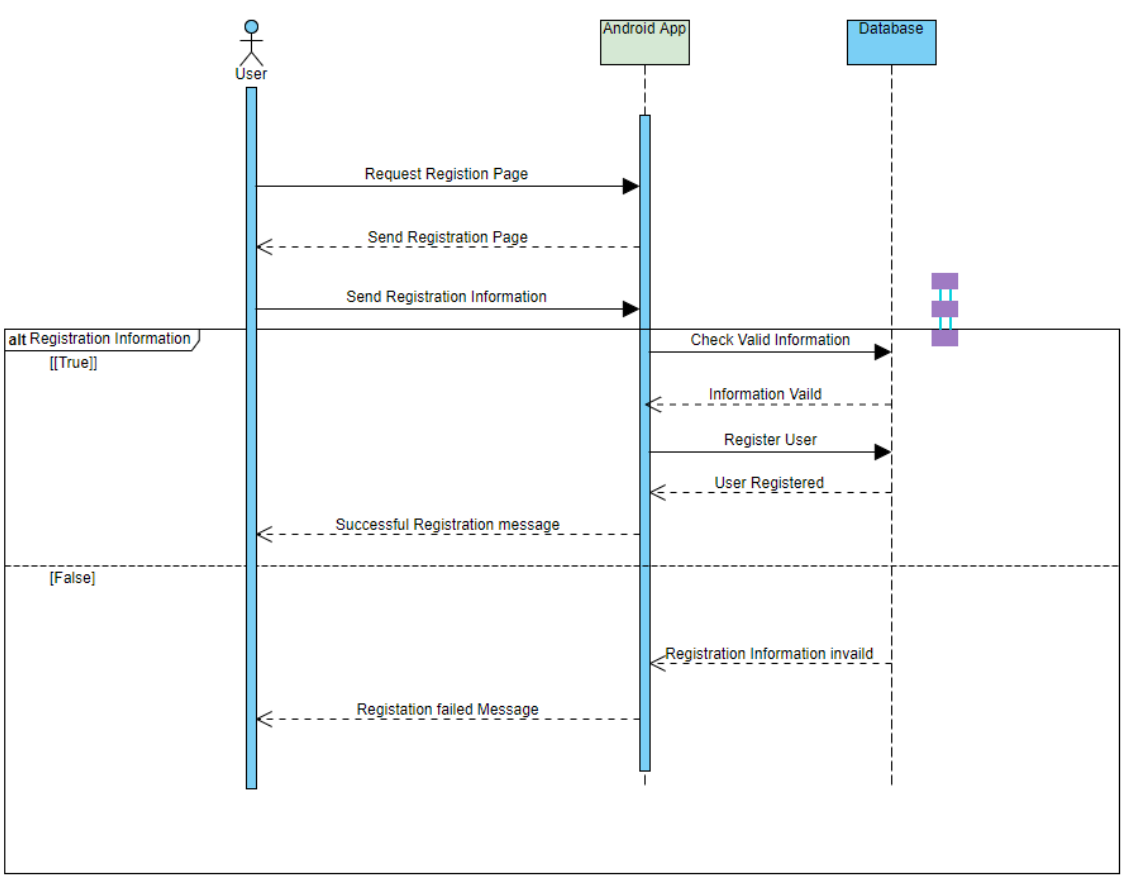

Figure 11: Registration Sequence Diagram

Android App Login Sequence Diagram

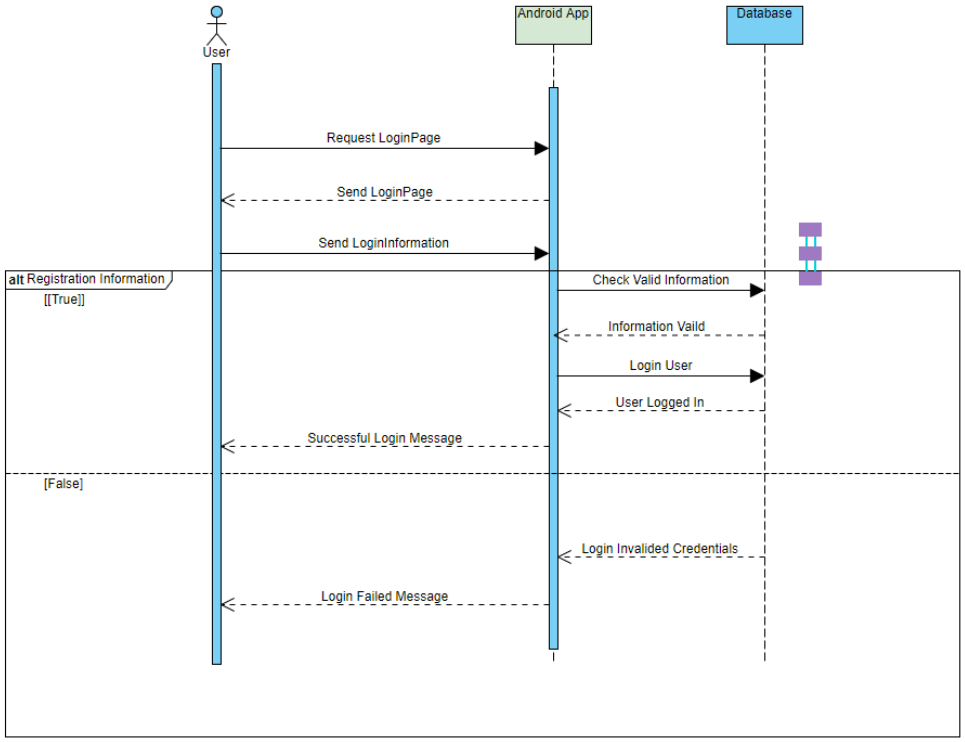

Figure 12: Login Sequence Diagram

Android App General Use

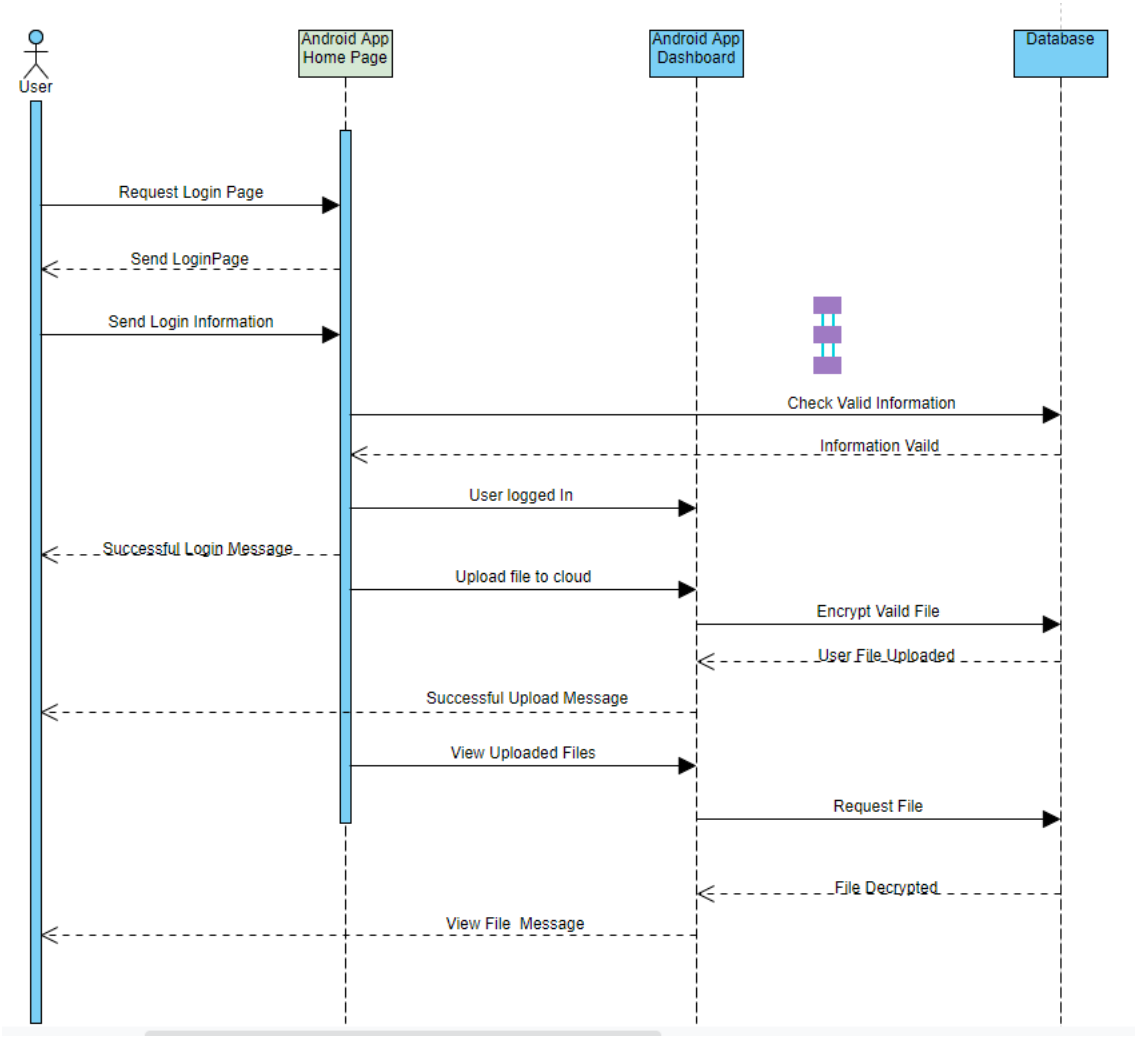

Figure 13: General Use Sequence Diagram

Android App Logout

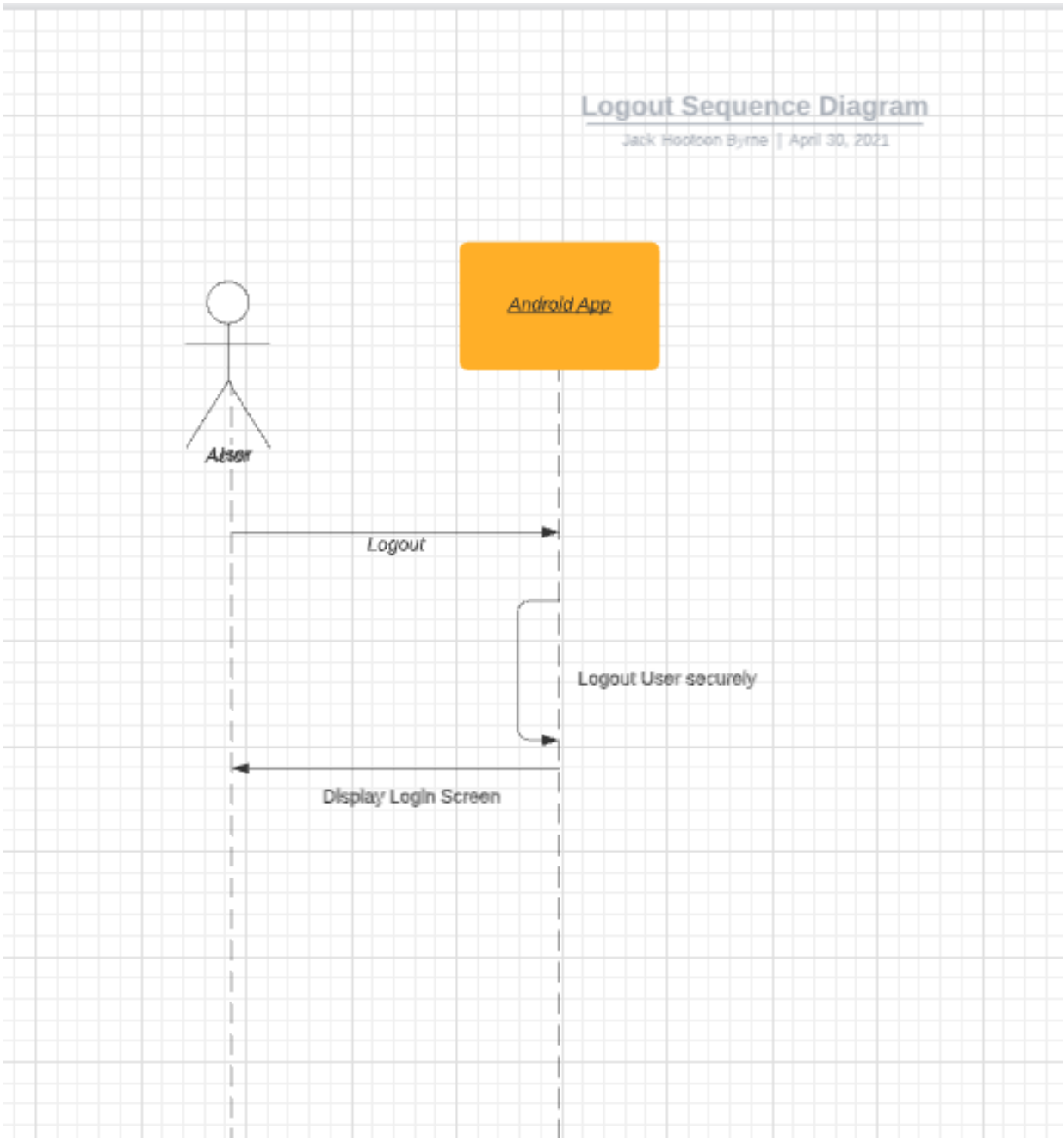

Figure 14: Logout Sequence Diagram

Bibliography

Mulonda, Y., 2020. Raspberry Pi 3 — Shell Scripting — Door Monitor (An IoT Device). [online] Medium. Available at: <https://medium.com/coinmonks/raspberry-pi-3-model-b-shell-scripting-door-monitor-b44944f82d87> [Accessed 26 November 2020].

Paradigm, V., 2021. *What is Sequence Diagram?*. [online] Visual-paradigm.com. Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/> [Accessed 11 December 2021].

Table of Figures