

National College of Ireland

Bachelor of Science (Honours) in Computing

Cyber Security

2020/2021

Sean Dolan

X17467042

X17467042@student.ncirl.ie

DocSafe

Technical Report

Contents

Executive Summary.....	3
1.0 Introduction	3
1.1. Background.....	3
1.2. Aims	4
1.3. Technology	5
1.4. Structure	6
2.0 System.....	6
2.1. Requirements	6
2.1.1. Functional Requirements	6
2.1.1.1. Use Case Diagram.....	7
2.1.1.2. Requirement 1 User Registration	8
2.1.1.3. Description & Priority.....	8
2.1.1.4. Use Case for User Registration.....	8
2.1.1.5. Requirement 2 Email Verification.....	9
2.1.1.6. Description & Priority.....	9
2.1.1.7. Use case for Email Verification Authentication	9
2.1.1.8. Requirement 3 Authentication	10
2.1.1.9. Description & Priority.....	10
2.1.1.10. Use case for Authentication.....	10
2.1.1.11. Requirement 4 Add Files.....	11
2.1.1.12. Description & Priority.....	11
2.1.1.13. Use case for Adding Files.....	11
2.1.1.14. Requirement 5 Delete Files	12
2.1.1.15. Description & Priority.....	12
2.1.1.16. Use case for Deleting Files	13
2.1.2. Data Requirements.....	14
2.1.3. User Requirements	14
2.1.4. Environmental Requirements.....	14
2.1.5. Usability Requirements	14
2.2. Design & Architecture	15
2.3. Implementation	18
2.4. Graphical User Interface (GUI)	26
2.5. Testing	29

2.6	Evaluation	33
3	Conclusions	36
4	Further Development or Research.....	36
5	References.....	37
6	Appendices.....	38
6.1	Project Plan.....	38
6.2	Reflective Journals.....	38
6.3	Other materials used.....	40
1.0	Objectives	41
2.0	Background	42
3.0	Technical Approach.....	43
4.0	Special Resources Required	43
5.0	Project Plan	44
6.0	Technical Details	44
7.0	Evaluation	44
8.0	Invention Disclosure Form (Remove if not filled)	44

Executive Summary

In this report I will be discussing the development of my final year project, DocSafe. When reading this report I will take you through the background of the idea and why I feel it is a complex and innovative topic that relates to my specialisation in Cyber Security. The report will start with the aims of what I set out to achieve and what sort of technologies I will be using to achieve them. You will see as the report goes on how I addressed the issues I have set out at the beginning of the report and what sort of methodologies I followed in order to ensure I reached the goals I have set for myself. This report will touch on the importance of cyber security with addressing some of the aspects and problems that come with cyber security. I will be demonstrating how my project and research will address common and complex security issues that come with the project idea. My report will give you a better understanding on some cyber security principles such as, secure communication, authentication and implementation of emerging technologies. I will also be demonstrating the use of Google Firebase which is a backend service provider and I will be showing the fantastic capabilities this emerging service will have to offer and how it helps with the development of my project. This project idea is innovative and complete as it works as it should and shows some commercial potential, it was a difficult project to develop but as you will see through the report I have managed to meet the requirements and get it done.

1.0 Introduction

1.1. Background

I chose to do this project as I feel it will be very useful for users that would like to store some data that they feel needs to be protected and that only they can access this information whenever they need too. I understand how important the protection of personal information can be for some users so I wanted to create and advertise a name that they can rely on and trust. Storage of your personal files has become more of a necessity for most technical and non-technical users and it is important that this data is secure and ensures integrity. The innovation behind this project idea came from a personal need, I tend to travel the world as much as I can and I found myself carrying around personal documents needed for travelling on my person. I want to be able to eliminate the idea of carrying paper documents and securely store them on a service I can trust and is super easy to use and set up.

I wanted to be able to demonstrate my understanding of safe storage and proper authentication to the best of my ability. I have always had a keen interest in web development and JavaScript and I wanted to further develop my knowledge of these and apply it with my newly gathered understanding of cyber security. I want to advertise this project as safe place that users can trust, the main aim is to let users sign up and use the application anywhere in the world and access their data as they need it. When it comes to protecting any app or system against attackers it can be very difficult to do so, I want to be able to address these complex issues such as secure communication, proper management of users and verification steps that need to be put in place in order for a user to actually use this web application. *“Cyber Security is important because it protects all categories of data from theft and damage ” (Tunggal, 2021)*

1.2. Aims

- **How can I ensure secure storage?** – This question is one that is very important for a project of this standard, it can be the most complex situation that should be addressed. Ensuring the storage of user data is safe and secure is a huge responsibility and very important in cyber security. I plan to use firebase to the best of its ability to handle the storage of user files. I need to make sure that each user that uploads a certain file will have a directory created under the unique ID assigned to them to ensure that their files can only be accessed by them when they are logged in. Users will need to be able to download and delete any file that they upload, once a file is deleted it should not be able to be restored and users will not be able to recover them. I need this to be as secure as possible and safe for the user to trust and in order to do this I must ensure I craft my code in such a way that it uses the technology within firebase to an advantage and ensures the user files are in a safe place. If we put the secure storage of user files to the side and actually talk about the storage of user information, I want to be able to ensure the safe storage of password and in order to ensure this I will need to implement passwords hashing before storing them so if there was to be a system compromise then we can ensure that user information is encrypted.
- **How will I tackle Session Management?** – This is very important when we consider cyber security and web development. Session hijacking is very much a big problem for many people and it is more common than you would think. According to the most recent OWASP Top 10 list, *“Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users’ identities temporarily or permanently.”* (D'Amico, 2019)

I want to be able to ensure that there is proper session management put in place so that when a user forgets to log out of the website and closes the tab then they should automatically be logged out. I also need to ensure that requests are handled properly between the user and application as this plays a big part in session management, requests made by users must respond to that particular user. I need to be able to make sure that what the user requests is actually related to them and I plan to do this with the creation of unique IDs assigned to the user when they sign up with their email and password. When the user is logged in, I need the application to listen to the authentication changes happening and return the appropriate material to the user.

- **How Can I ensure Secure Communication?** – When I talk about secure communication I am referring to end-to-end encryption so that third-party listeners cannot detect what is going on over a network. To do this I will host the website through a host provider that provides an SSL certificate. The sole purpose of SSL certificates is to protect and ensure safe sever-client communication. As I plan to pass through emails and passwords from a device to a server I need to be able to ensure that the data is transferred safely and I can do this with SSL certs as every bit of information will be encrypted and only the browser or server will be able to decrypt this type of data passed through. This will prevent packet-sniffing by attackers on a network which is very common and the most effective form of hacking a user for their credentials. *“HTTPS authenticates the communicating end points and provides confidentiality for the ensuing communication.”* (Naylor et al., 2014)
- **How Will I Pass A User Through Authentication / Validation?** – This is a very important aspect of my aims, I have put these two together because I plan to work them alongside each other. Authentication is such an important factor when it comes to cyber security and it can be very complex and hard to address. *“These kinds of flaws can be extremely serious in web applications. They put businesses at a very high risk.”* (D'Amico, 2019)
As my project is described as secure storage application I want to implement a way for users to be verified through their email address. My plan would be to allow the user to sign up to the application but in order to actually upload documents to their dashboard they will need to verify their email address before doing so. This ensures the authorization of human and machine, it will also help against attackers trying to implement brute force attacks with false emails as they would need to confirm the email before executing any sort of the functionality. When a user is signing up I need to pass them through a serious of validation checks too. This is for their email and password to ensure they have set up the correct email they want to use and to stress the importance of a strong password. I would like to also incorporate some sort of two-factor authentication when it comes to signing the user into their account as an added layer of security.

1.3. Technology

To develop this project I will be using HTML and vanilla JavaScript.

For storage, authentication and hosting I will be using Google Firebase. Firebase is an emerging backend service with industry leading tools that developers can use to develop their applications to the highest standard. I am using firebase as it is built on Googles infrastructure so it adds a high level of security standards. Firebase provides me with proper authentication checks and storage for my user files that they want to upload. I will be using firebase to deal with customer request and session management, I plan to write my own code around firebase references that are contained within the documentation provided by Google. *“Firebase is Google’s mobile platform that helps you quickly develop high-quality apps and grow your business”* (Firebase, 2021)

Firestore is fantastic for start-up developers that are wanting to build and grow their business from the ground up. Starting your own back end service can be very cost-effective and require a lot of technical work, it can also pose a huge security risk for the developer who is lacking experience. *“Google Released firestore in 2016. Its goal is to provide the tools and infrastructure that you need to build great apps, grow a successful business, and earn from your hard work”* (Moroney, 2017)

1.4. Structure

Section 2

- Requirements – this is where I will run through my functional requirements. I will also address data requirements, user requirements, environmental requirements and usability.
- Design and Architecture – This will show how the system is designed and ran.
- Implementation – Here I will discuss how I addressed and added what is set out in the requirements and system design and how I implemented that into my code
- GUI – Here is where I will discuss my system UI. Screenshots will be given with detailed descriptions of what is happening in each.
- Testing – Here is where I run my system testing for my application. I will discuss the tools I used in order to run my tests.
- Evaluation – Here is where I will look back at my project aims and evaluate them.

Section 3 – In this section I will be giving my conclusion on the report. The advantages/disadvantages, limitations and strengths of my project will also be discussed.

2.0 System

2.1. Requirements

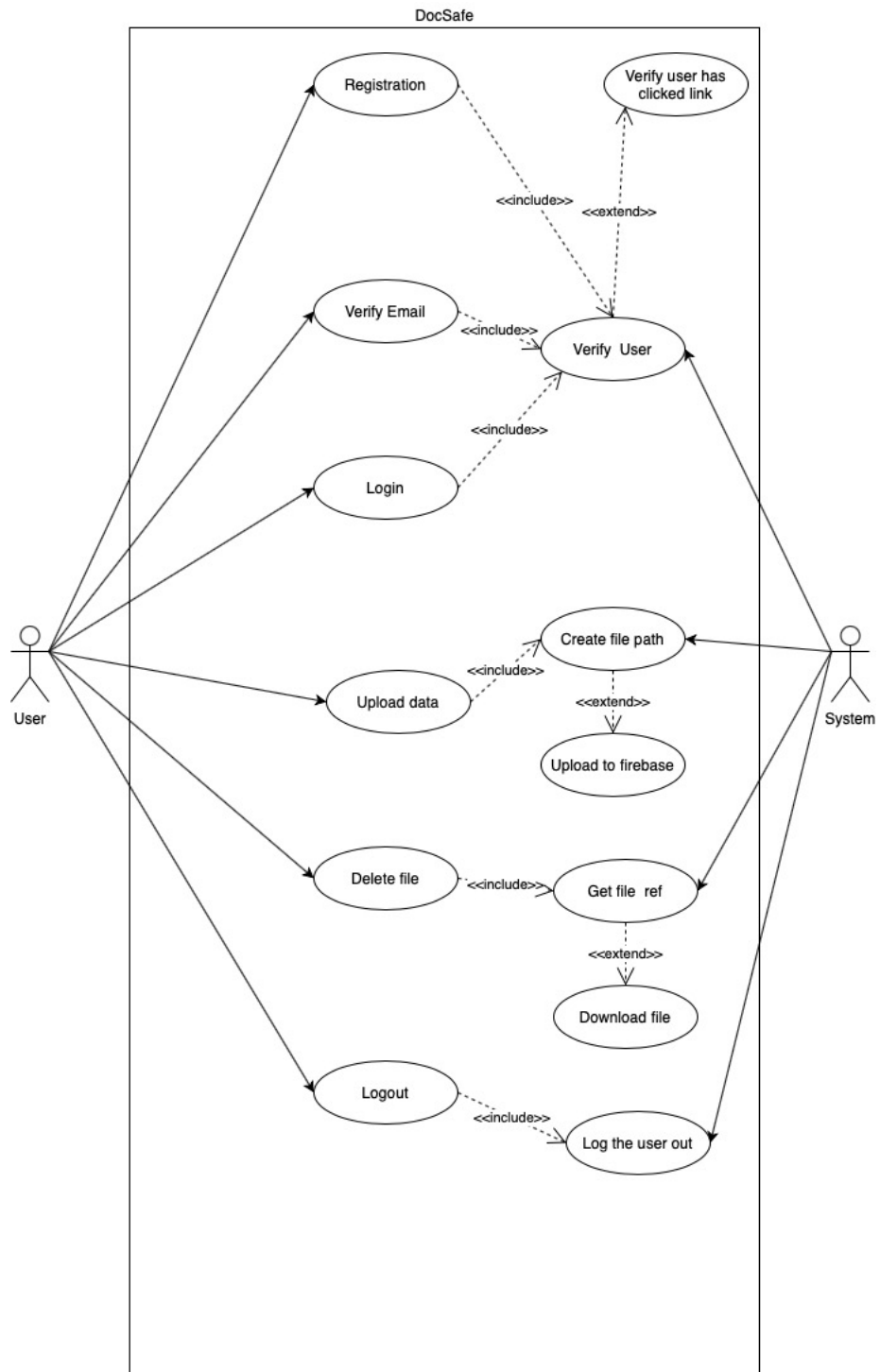
When first developing the idea for this project I had intended to make it a standalone android application and I wanted the application to only work with smartphones. My supervisor intelligently pointed out the limitations this was going to have from a commercial stand point and suggested I steer away from the developing an android studio application and move onto the idea of a web application instead. By making this sort of change it really opened up the amount of limitations I was setting myself with a phone based application and made me realise that I could in fact develop a more stable and functional project that would address many more issues in relation to cyber security.

In saying this my requirements for this project changed quite a bit, not for bad but for good reasons. The whole concept of the idea behind what I want to do remains the but instead of doing it all on a smart phone I shifted the idea over to a web application.

2.1.1. Functional Requirements

1. User Registration
2. Email Validation
3. Authentication/Login
4. Add Files
5. Delete Files

2.1.1.1. Use Case Diagram



2.1.1.2. Requirement 1 User Registration

Here we discuss the requirement for user registration. This is virtually the most important part of any application that is using user interaction, we need to ensure a user has signed up and successfully created an account.

2.1.1.3. Description & Priority

This is where a user will be asked to sign up with their email and password, they will be asked to enter a strong password that is equal to or more than 10 characters. They must follow password guidelines such as password length, upper/lower case characters in order to have a successful registration. This is a high priority as users cannot use this application unless they provide a strong password.

2.1.1.4. Use Case for User Registration

Scope

The scope of this use case is to ensure a user can register following the appropriate guidelines put in place to ensure proper security.

Description

This use case describes the process a user will take in order to register for the app. The user will first have to enter their email that will be validated and then go onto their password that will also be validated.

Flow Description

Precondition

The user registration form is on display. The user is not registered.

Activation

This use case starts when a user clicks register

Main flow

1. The system identifies the user action for register.
2. The user enters their email and password.
3. The system checks if the email exists, and the password follows verification.
4. The system creates the account and sends a verification email to the user.
5. The user closes the alert for email sent.
6. The system will display the dashboard for the user.

Alternate flow

A2: Invalid email and password

- a. The system checks the and confirms there is an account already registered for the user under the email provided.
- b. The user will then try to log in with the email recognised.
- c. The user enters a password less than 10 characters
- d. The system will display the alert "Password too short"

Exceptional flow

E4: Unable to create account

- e. The system has not taken the account details from the user
- f. The user has clicked submit when registering but the website crashes.
- g. The system has failed to connect to firebase.

Termination

The system presents the dashboard upon successful account creation.

Post condition

The tab is closed and a new session is created.

2.1.1.5 Requirement 2 Email Verification

Here we discuss the requirement for email verification. This is the 2nd most important part of this application as it is needed for when a user wants to add data to their account.

2.1.1.6 Description & Priority

When a user successfully passes the registration process they will be sent an email verification to the email they have set up. This step is high priority as it enables us to check if the email used is a real email and that it is human and machine we are dealing with. A user must perform this step in order to upload data to their account.

2.1.1.7 Use case for Email Verification Authentication

Scope

The scope of this use case is to ensure a user validated their email so that they can use the full features of the web app.

Description

This use case describes the process a user will take in order to gain access to the application. The user has already set up an account and has used the app before.

Flow Description

Precondition

The user is registered and has passed all verification checks.

Activation

This use case starts when the user has just registered.

Main flow

1. The system has completed the registration for user and sends email for verification.
2. The user opens their inbox and clicks the link for verifying their email.
3. The system verifies the email and returns the request back to the server.
4. The user will refresh the page and their email verification will return true.
5. The user can now upload data to their account.

Alternate flow

A1: Invalid email entered

6. The user entered in the wrong email.
7. The system sends the verify to the wrong email inputted by user.
8. The user needs to create a new account with the correct email.
9. The system will send an email to the newly created account

Exceptional flow

E1: Network Error

10. The system was unable to send the email due to network error
11. The user has clicked submit when registering but the website crashes.
12. The system has failed to connect to firebase.

Termination

The system updates the user verifications.

Post condition

The tab is closed and a new session is created. Tab is closed

[2.1.1.8 Requirement 3 Authentication](#)

This is where a user is authenticated before logging into the application. I planned to add two-factor authentication into my web app but it turned out to be cost effective and I was unable to do. I had to make use of the firebase authentication methods instead.

[2.1.1.9 Description & Priority](#)

In order to access this application we must look at how you can enter safely and have the correct credentials before gaining access. This feature is important as it prevents unauthorized personnel from gaining access to your application.

[2.1.1.10 Use case for Authentication](#)

Scope

The scope of this use case is to ensure a user is validated and already registered within the application and their details are found in firebase.

Description

This use case describes the process a user will take in order to gain access to the application. The user has already set up an account and has used the app before.

Flow Description

Precondition

The user is registered and wants to access the app

Activation

This use case starts when a user clicks login.

Main flow

1. The system prompts the user for login.
2. The user enters their email and password.
3. The system checks if the email exists and the password matches.
4. The system verifies the user credentials and gives access.
5. The user can now use the application

Alternate flow

A3: Invalid pin and password

6. The system checks the and confirms there is an account already registered for the user under the email provided.
7. The user enters a password or pin that does not match.
8. The system displays “invalid password” message.
9. The system allows three attempts before the account is locked.

Exceptional flow

E4: Unable to log in / Failed to get connection

10. The system has taken in the appropriate details from the user
11. The system cannot get a connection to the server
12. The system returns 404 error

Termination

The system has verified the user.

Post condition

The tab is closed and a new session is created. Tab is closed

[2.1.1.11 Requirement 4 Add Files](#)

It is important that a user has been able to pass the first 3 requirements before moving onto this requirement which is where the core functionality is taking place.

[2.1.1.12 Description & Priority](#)

This is a main functionality of the application and it is the main use therefore it is high priority. The user should be able to add any file they would like. The file should be created under the unique ID created for the user and stored under that ID.

[2.1.1.13 Use case for Adding Files](#)

Scope

The scope of this use case is the process a user takes to add files.

Description

This use case describes the process a user will take in order to add files to the application from their device.

Flow Description

Precondition

The user is logged in, verified and wants to add files

Activation

This use case starts when a user clicks upload.

Main flow

1. The user clicks Add Files.
2. The user chooses the file they wish to add.
3. The user clicks Upload.
4. The system checks the device for the file they have selected.
5. The system will then transfer the file from the device to firebase.
6. The system will create a reference to the user ID and put the file in.
7. The system will refresh the page and present the user with the file added.

Alternate flow

A4: File failed to upload

1. The system checks the file the user wants to add
2. The system cannot add the file to the application
3. The system displays "error: try again" message.

Exceptional flow

E3: System crash / network interrupted

1. The system has failed to reach server
2. The system has crashed while trying to upload document
3. The device the user is on drops in signal

Termination

The system has uploaded the file for the user and is now displaying the file.

Post condition

The tab is closed and a new session is created. Tab is closed

2.1.1.14 Requirement 5 Delete Files

It is important that a user is able to delete files as well as adding them. A user should be able to delete any file they want from their dashboard, once the file is deleted then it gone and cannot be recovered.

2.1.1.15 Description & Priority

As well as adding files to the application the user should also be able to delete these files too. This is also a very high priority requirement as the user should be in control of what they want on their profile.

2.1.1.16 Use case for Deleting Files

Scope

The scope of this use case is to ensure a user can delete files from their profile.

Description

This use case describes the process a user will take in order to delete files from their account.

Flow Description

Precondition

The user is logged in, verified and wants to delete files

Activation

This use case starts when a user clicks delete file.

Main flow

1. The user selects the file they want to delete.
2. The system verifies the file.
3. The system will delete the file and display an alert saying, "file deleted".
4. The user will now see their file is deleted from the dashboard.

Alternate flow

A2: File failed to delete

4. The system checks the file the user wants to delete
5. The system cannot delete the file from the application
6. The system displays "error: try again" message.

Exceptional flow

E1: Server down / internet connection failed

4. The server is down and unable to process request.
5. The system has lost connection.
6. The device the user is using has lost signal.

Termination

The system has deleted the file for the user and is now displaying the updated dashboard.

Post condition

The tab is closed and a new session is created. Tab is closed

2.1.2 Data Requirements

Email

The system needs a valid email in order to perform the core functionality behind the project aims. If the user provides an email they cannot verify then they will not be able to upload any documents to their account. The system is able to determine whether or not the user has actually verified their email and will block the activity of uploading documents until they do.

Password

A user should be able to provide the system with a strong password at the registration stage. The system will require the user to enter a password with more than 10 characters, one upper case character, one lower case character and at least one number. If a user fails to provide this sort of password then they will not pass the validation checks and fail registration.

User Files

A user needs to be able to upload files to their account. The application will need access to your file system on your device in order for a user to choose a file they would like to upload. Some users might have security settings on where they may not be able to add files to a untrusted source they have never visited before.

2.1.3 User Requirements

User requirements are what the user will need from the system.

- User is able to register an account
- User is able to verify their email
- User is able to upload data as they please
- User is able to download their data
- User is able to delete their files
- User is able to securely log out

2.1.4 Environmental Requirements

An pc with up to date browsers on it. A stable internet connection. Any sort of mobile device. These are just basic requirements needed in order to access this web application.

2.1.5 Usability Requirements

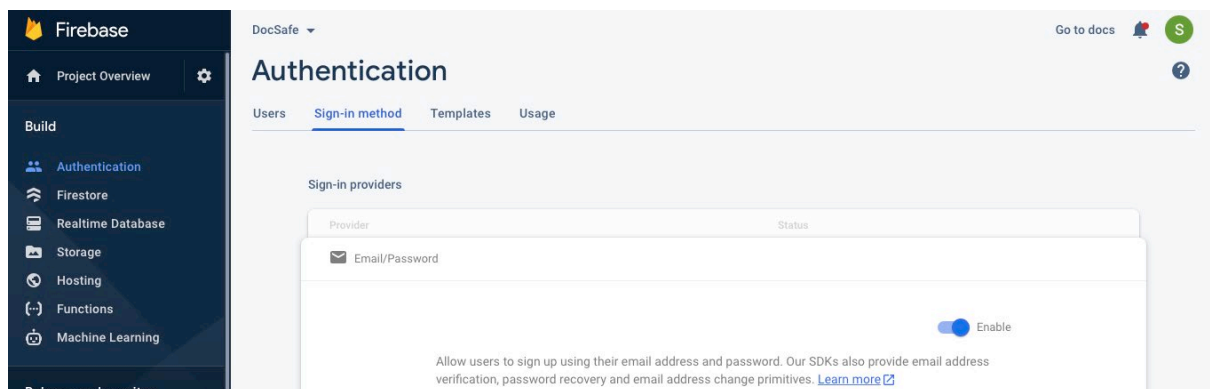
This application will work on any device that it connected to the internet. A secure internet connection that is private is more ideal than a connection that is open to the public, it is always good practice to steer clear of public networks as they are unprotected. As stated, the user needs to provide a valid email address for this application to work and a strong password for registration.

If I developed this application as a standalone Android app then I would have been limiting myself to only storage on a smartphone device which would cause many limitations. It made much more sense from a technical and commercial stand point to develop this project as a web application so that it can be used across a wide spread of devices.

2.2 Design & Architecture

I want to touch on how the backend of this application is designed and how I implemented it into my project. For my backend I am using Google Firebase, firebase is backend service provider that helps developers develop high level applications in a safe and secure way. Once you have created your project within the firebase console you will then be prompted to then add in the firebase SDKs into your code before you can actually use any of the services.

First I will discuss how I am running my authentication through Google Firebase. When I consider my project I need to be able to take in an email and a password, firebase offer a whole array of ways you can create users and authenticate them. You can use the standard email/password approach or even use a phone number, google sign-in, Apple ID or add in a custom authentication system that suits business needs. As I only need an email and password for my project I need to enable this within the firebase console under my project.



Now upon enabling this I need to setup my sign-in methods which I will discuss in the next section, Implementation, but I want to talk more about how firebase works with my project and why I feel it is a fantastic tool for developers. Firebase allows me to have full control over my security rules and by authenticating users I can fully restrict read/write access to my firebase data. Firebase authentication is basically making use of token generation and what token generation means is that we are taking identifiable, confirmed user data from an authorised source and then passing securely through to firebase. When we talk about encryption it can be very daunting and can scare a lot of developers but with firebase we don't have to worry about it as much, when creating a user a with an email/password firebase will take the plain text password and handle all the encryption for us and actually hash our passwords passed through by the user. Password hashing is fantastic as it adds a huge level of security for our users if there is ever a system compromise it means user accounts will be secure and it is one less security risk we need to worry about.

Authentication is playing a big part in this project but I also need to consider how I plan to store user files within firebase storage. Firebase offers a couple of options for storing data, such as, Fire store which is a cloud based, NoSQL database and also a Realtime Database. I actually only want to take in a minimal amount of user details so I will only be using firebase storage and the reason I chose to do is for design and security reasons. I want to make the use of this application to be very basic and user friendly so that when a user signs up they can do so within seconds, I also wanted to limit the amount of data I was storing from a user so that in case of a system compromise attackers would only really get the user email. I have designed my firebase storage to only store user data for verified users and that anything stored by a user is under their unique ID generated for them by firebase, this ensures that their data can only be viewed by them.

In order for firebase to actually talk to my application and read/write data, I am provided with the firebase SDKs when my project is created in the firebase console. Google will prompt you to add these SDKs into your application before you make use of any firebase services

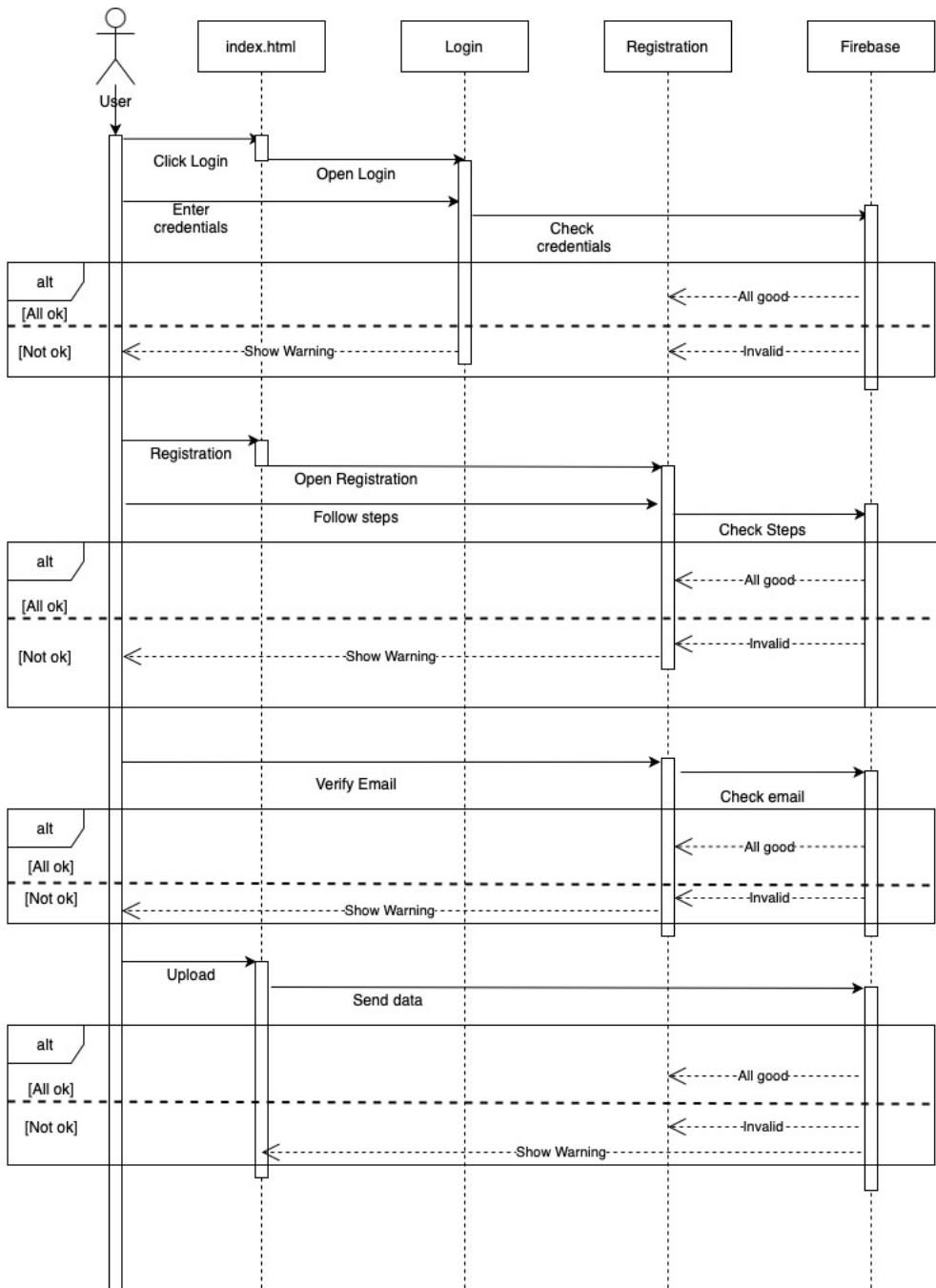
```
111 <!-- The core Firebase JS SDK is always required and must be listed first -->
112 <script src="https://www.gstatic.com/firebasejs/8.3.1/firebase-app.js"></script>
113 <script src="https://www.gstatic.com/firebasejs/8.3.1/firebase-auth.js"></script>
114 <script src="https://www.gstatic.com/firebasejs/8.3.1/firebase-firestore.js"></script>
115 <script src="https://www.gstatic.com/firebasejs/8.3.1/firebase-storage.js"></script>
116 <!-- TODO: Add SDKs for Firebase products that you want to use
117 | | https://firebase.google.com/docs/web/setup#available-libraries -->
118 <script src="https://www.gstatic.com/firebasejs/8.3.1/firebase-analytics.js"></script>
119
120 <script>
121   // Your web app's Firebase configuration
122   // For Firebase JS SDK v7.20.0 and later, measurementId is optional
123   var firebaseConfig = {
124     apiKey: "AIzaSyBliNsx9b2nDxvj8e5KQSEWoPZeoG1IVw0",
125     authDomain: "docsafe-912b0.firebaseio.com",
126     projectId: "docsafe-912b0",
127     storageBucket: "docsafe-912b0.appspot.com",
128     appId: "1:653159950777:web:f936333c2d72ea267eae68",
129     measurementId: "G-FT4JMP3B3N"
130   };
131   // Initialize Firebase
132   firebase.initializeApp(firebaseConfig);
133   firebase.analytics();
134
```

Above we can see the firebase SDKs, provided by Google Firebase, implemented into my code so that I can make calls to the firebase console. If we consider the top four script tags, these are how I can call different services from firebase. All I need to do is create a reference to these services so that they can be used throughout my application. Below I have included a screenshot of how I am creating these references and in my implementation section I will discuss where I use these references and how I structure my code around them for it to work how I want.

```
135 //auth and firestore ref
136 const auth = firebase.auth();
137 const db = firebase.firestore();
138 const store = firebase.storage();
139
140 // update setting
141 db.settings({timestampsInSnapshots: true});
142
```

For the front end development of my project I used HTML and vanilla JavaScript. Firebase will actually provide you with some documentation for implementing their services into JavaScript, however, this documentation can be challenging if you do not understand the basic concepts of firebase. I have two JavaScript classes in my project and one HTML file too, I used a variety of JavaScript tools in order to manipulate my index UI. For my design and CSS I used a mix of my own CSS class and materialize CSS. Materialize is also developed by Google, it is a UI component library and will handle responsive aspects of CSS and work with all modern browsers.

Sequence Diagram



2.3 Implementation

I want to talk about how I implemented my authentication into my project with the use of Google Firebase. As I described above I need to add in the firebase SDKs and create some references to the firebase service I want to use, in this case this is how I call my authentication service.

```
const auth = firebase.auth()
```

Now that I have the reference to my firebase authentication I first use it for creating my users with their entered email/password. For my signup method I have quite few things going on and I want to address my approach to signup method and how I implemented firebase authentication. First I need to actually grab the values from my create-form located in my index.html file. Now from the screenshot below you can see I am making a reference to my create form and also using the query selector method to grab the form, I then use this reference and add it to an event listener that will grab the user email/password values and run them through a series of checks before moving onto the account creation stage.

```
2  const signupForm = document.querySelector('#signup-form');
3  signupForm.addEventListener('submit', (e) => {
4    e.preventDefault();
5
6    const email = signupForm['signup-email'].value;
7    const email2 = signupForm['signup-email-retype'].value;
8    const password = signupForm['signup-password'].value;
9    const pass2 = signupForm['confirm-password'].value;
--
```

Once my event listener has been activated and it has grabbed each value from the four references I have created above, I run the values through a few if/else statements to validate the users input.

```
11  if(email2 !== email){
12    alert('Emails do not match.')
13    return false;
14  }
15  else if(password.length<=10){
16    alert('Password must be more than 10 characters.')
17    return false;
18  }
19  else if(password.search(/[a-z]/) === -1){
20    alert('Password must contain one lower character.')
21    return false;
22  }
23  else if(password.search(/[A-Z]/) === -1){
24    alert('Password must contain one uppercase character.')
25    return false;
26  }
27  else if(password.search(/[0-9]/) === -1){
28    alert('Password must contain at least one number.')
29    return false;
30  }
31  else if(password !== pass2){
32    alert('Passwords do not match.')
33    return false;
34  }
```

Above you can see the checks I have put into place to ensure the user has entered the correct values needed in order for this application to sign them up. Take note of the password check as I want users to create long password with the addition of numbers and upper/lower case characters to increase complexity and security. If a user does not follow the correct format then they will be shown an alert to guide them into the right steps.

Once a user has successfully passed the validation checks of this application I can now move onto actually authenticating the user and sending them their verification email. Here is where my firebase authentication reference comes into play. Below you can see the function being called which is, **auth.createUserWithEmailAndPassword()**, this function does exactly what it is set out to do, it creates a user with the email and password they have entered. This function is firebase function and it will grab the values of the email and password in plain text and handle all the encryption for us.

```
35     else{
36         auth.createUserWithEmailAndPassword(email, password).then(() => {
37             var user = firebase.auth().currentUser;
38             user.sendEmailVerification();
39             alert("A verification email has been sent to you.
Please verify your email so you can use our services!");
```

If you look at the above code on line 38, you will see another firebase function that I run when I create new users. This function will create a verification email for us and send it over to the email address the user has entered at the registration stage. This email is sent from the firebase console to the user, it will contain a link the user must click in order to verify their email. By default user verification is set to false so when a user clicks this link firebase will update the user to be true. It is completely up to the developer on how they want to make use of this feature so what I wanted to do was actually log the user into the main dashboard of their account and restrict upload privileges until they have verified their email address.

```
102 var uploadBtn = document.getElementById('upload-button');
103 uploadBtn.addEventListener('click', function(){
104     //if statement to check.
105     if(!firebase.auth().currentUser.emailVerified){
106         alert("Please verify your email");
107     }else{
108         el = document.getElementById("modal-create");
109         el.style.visibility = "visible";
110         document.getElementById("modal-create").style.height = "30%";
111     }
112 });
```

The above code is how I restrict access to the upload button, if we look at line 105, I have an if statement put in place to check if the user email is verified. Here we can the use of **firebase.auth()** to grab the value of the current user and to check if their email is set to false or true. If the email is set to false then they will not be able to launch the upload modal on the dashboard and an alert will be shown to remind them to verify their email address. As well as restricting access to the button for my upload method I want to also put in the extra security rule check on my firebase storage within the firebase console. This security rule will restrict read/write access for a user if their email is not verified, this adds the extra layer of security when it comes to account creation. We are able to ensure that the user behind the email is human and not machine trying to exploit our authentication system put in place. Below are the security rules I have set on my firebase storage. First I check if the user is actually signed in with auth and then I check the UID and token related to that UID to ensure the email verified returns true. If the email returns false then the user has insufficient permissions and will not be allowed to upload to firebase.

```
1  "$uid": {
2    ".read": "auth != null && auth.uid == $uid && auth.token.email_verified === true",
3    ".write": "auth != null && auth.uid == $uid && auth.token.email_verified === true"
4  }
```

Moving on from how I register, create and verify my users I want to touch on how I sign my users in. For this part I had planned to implement two-factor authentication but, unfortunately this turned out to be cost-effective and did not fit in with my budget of this project. As it is a huge loss in regard to security features I have still tried to make it as safe as possible for users to log into their account. As I am using firebase for user creation then I will also need to use it for user login, firebase is nice enough to pass details through with encryption and it will also encrypt and decrypt passwords as they are communicated from device to server. One of the best strongpoints for firebase is the authentication.

```
170 const loginForm = document.querySelector('#login-form');
171 loginForm.addEventListener('submit', (e) => {
172   e.preventDefault();
173   // get user info
174   const email = loginForm['login-email'].value;
175   const password = loginForm['login-password'].value;
176   // log the user in
177   auth.signInWithEmailAndPassword(email, password).then((cred) => {
178     // close the signup modal & reset form
179     const modal = document.querySelector('#modal-login');
180     M.Modal.getInstance(modal).close();
181     loginForm.reset();
182     location.reload();
183   }).catch(function(error) {
184     // An error happened.
185     alert(error+" Please sign up!");
186   });
187 });
```

We can see the code above is how I sign my users into firebase and currently this is the most secure way firebase recommend you use it. I am again using my **auth()** reference to call back to my firebase console and check the values entered by the user to see if they match any users withing the authentication page of firebase. Here is the reason I wanted users to create a long and complex password, it means the password is hard to guess and it limits the chances of brute force attacks on my system. As my passwords are hashed when coming to and from firebase it makes it extremely hard for attackers to be able to gain access to a user account.

When it comes to logging a user out we can use the **auth()** reference again to securely sign out users from their account and from firebase. In the below code you can see that I call **auth.signOut()** on line 165, this will sign the user out completely and take them back to the landing page.

```
161 //LOGOUT
162 const logout = document.querySelector('#logout');
163 logout.addEventListener('click', (e) => {
164   e.preventDefault();
165   auth.signOut();
166   location.reload();
167 });
```

Now if we consider the above code snippets I have mentioned in relation to user activity, I need to implement a way for the system to listen to these authentication changes so that I can manipulate the UI according to the users sign in status. To do this I can again use my **auth()** reference to listen the user state changes and this actually another method provided by firebase.

What this method allows me to do is to determine whether a user is signed in with the authenticated credentials from firebase and I can decide what I want to show depending on the sign in status and if the user is authenticated. The below code allows me to get the authentication status of a user and return my **setupUI()** method that I have create and will discuss next.

```

1  auth.onAuthStateChanged(user => {
2    if (user) {
3      setupUI(user);
4    } else {
5      setupUI();
6    }
7  });

```

As seen from above we are listening to the authentication status of a user and returning a method that will display the UI for the user that is currently signed in. My setup UI is quite a long piece of code so I am to break it down in to a simple format with the most important pieces of code.

```

1  const setupUI = (user) => {
2    if(user){
3      db.collection('users').doc(user.uid).get().then(doc => {
4        //innerHTML will be displayed for when a user is signed in.
5        const html = `
6          <div>Logged in as ${user.email}</div>
7          <p>If you would like to make changes to your account then please contact us!</p>
8          <p> Contact: admin@docsafe.ie </p>
9        `;
10       accountDetails.innerHTML = html;
11     });
12     //clear account details
13     accountDetails.innerHTML = '';
14     loggedInLinks.forEach(item => item.style.display = 'block');
15     loggedOutLinks.forEach(item => item.style.display = 'none');

```

This part of the code is the start of my setup UI function. On line 3 you can see I am calling my database collection of users and I am going to use this on line 6 to get the email value of the current user that is signed in. I am using innerHTML to display data back to users as it is great for DOM manipulation and it is something that is new to me and I wanted to further develop my skillset on JavaScript literals. This particular innerHTML will allow the user to open the account details modal and view the email they are logged in with. On lines 14 and 15 you will notice I have logged in links and logged out links, I want the UI to only show the logged in links for users who are logged in so once users are logged in they will no longer be able to see the login or signup links located at the top of the page.

Next I want to display a head title for the user dashboard that will just simply say “Your Uploads” and this will display just before I list all the user files back to the user.

```

--
16  //This is just a header for the dashboard and will display when the user is signed in
17  const Head = `
18    <h3><u>Your Uploads.</u></h3>
19    <br />
20  `;
21  setUpDash.innerHTML += Head;
--

```

You see on line 21 that I am calling the innerHTML and applying it to the dashboard.

Now it is time to actually display the files back to a user. To do this we need to first create a reference to the storage service of firebase and it demonstrated in the below screenshot. What this reference is actually doing is getting the current user and their unique ID in order to get the files related to their UID.

```
24 //display files
25 var storageRef = firebase.storage().ref(user.uid);
26
```

Now I want to list all the items under the storage reference I have created above and for each result written back to the system will be ran through a function I created called **displayFile()**. So in the below code on line 28 we can see that each result that is found within the **listAll()** function will be passed through my **displayFile()** function so that the files can be seen by the user that is currently signed in.

```
27 storageRef.listAll().then(function(result) {
28     result.items.forEach(function(fileRef) {
29         displayFile(fileRef);
30     });
31 }).catch(function(error) {
32     // Handle any errors
33     console.log("error found");
34 });
```

Now that we have the list and results of the files that we want to write back to the user, we need to run them through the below function in order to do so. So first we get the file reference of the file we want to display and that is followed by the download URL created by firebase. On line 38 we can see I create a variable called name and what this does is get the name of the file from the file reference and I display it back to user on line 39. Below is essentially how I am displaying the files back to the user, on line 39 I am displaying the name of the file and under the file name I am giving the user two options, delete and download. Then on line 46 is where I add the innerHTML to the user interface.

```
35 function displayFile(fileRef) {
36     fileRef.getDownloadURL().then(function(url) {
37         var name = fileRef.name;
38         const html = `
39             <h5 class="center"><b>${name}</b></h5>
40             <button class="btn yellow darken-2 z-depth-0" onclick="deleteFile()">Delete</button>
41             <a class="btn yellow darken-2 z-depth-0" href="${url}">Download</a>
42         `;
43         <br />
44         <br />
45     `;
46     setUpDash.innerHTML += html;
--
```

A user should be able to do as they please with their data whenever they need to so I have implemented two features that allow them to download their files or delete their files as they please. To delete the file set up is very basic and easy, each file will have their own delete button under them so that a user doesn't delete the wrong file by accident because once a file is deleted it cannot be recovered. Below you can see within my innerHTML I have assigned an onclick to the button that will run a **deleteFile()** method when clicked.

```
const html = `
  <h5 class="center"><b>${name}</b></h5>
  <button class="btn yellow darken-2 z-depth-0" onclick="deleteFile()">Delete</button>
  <a class="btn yellow darken-2 z-depth-0" href="${url}">Download</a>
`
```

Below is how I am deleting the file from the user interface. I get the file reference of the file that the delete button is under and delete the file from firebase completely. I then reload the page and catch any errors in the meantime.

```
50 window.deleteFile = () =>{
51   // Delete the file
52   fileRef.delete().then(() => {
53     alert("Delete success");
54     //Reload page.
55     location.reload();
56     // File deleted successfully
57   }).catch((error) => {
58     // Uh-oh, an error occurred!
59     alert("Delete failed");
60   });
61 }
62 }).catch(function(error) {
63   // Handle any errors
64 });
65 }
66 }//end setupUI()
```

For a user to download any file they have uploaded we need to get the download URL created by firebase when a file is uploaded. Doing this is made simple with the use of the **getDownloadUrl()** function assigned to us by firebase.

```
36 fileRef.getDownloadURL().then(function(url) {
37   var name = fileRef.name;
38   const html = `
39     <h5 class="center"><b>${name}</b></h5>
40     <button class="btn yellow darken-2 z-depth-0" onclick="deleteFile()">Delete</button>
41     <a class="btn yellow darken-2 z-depth-0" href="${url}">Download</a>
42     <br />
`
```

You can see on line 36 we get the file reference and then the download URL. I then call the URL function on line 41 and it will allow the user to download any files they have uploaded. When files are uploaded to firebase, they are assigned an almost unrecognisable URL that is unique to the user and the file so there is no way of anyone actually knowing the URL unless it was shared.

Now I have mentioned how a user is able to delete and download files so now I want to talk about how I implemented the actual process of adding files to their account. So I start by getting my create-form where I will nest all the functionality for uploading data. I also need to get the ID of my choose file button as this is how a user will be able to choose the file from their device.

```
1  const uploadForm = document.querySelector('#create-form');
2  var fileButton = document.getElementById('fileButton');
3
```

Next I have set up an event listener for the choose file button and what happens is that when the user presses the choose file button and selects a file from their device this event listener will look for a change in the choose file upload.

```
4  fileButton.addEventListener('change', function (e){
5  |   var file = e.target.files[0];
```

Once this is done and the system has took in the file from the choose file property I now need to make a reference as to where I want this file to go when a user submits it. This part is the most important part as it is where I create the reference related to the current user signed into the application and grab their unique ID and create a path under that ID for the file they want to add in.

The below variable has been used through-out the application to gather information about the files the user has added into the firebase storage under their unique ID. This is how I make sure I have separate file paths for each individual user so that only the user with the UID that matches the file path can actually access the data that is stored within the firebase console.

```
7  var storageRef = firebase.storage().ref(auth.currentUser.uid+"/"+file.name);
```

Next I have an event listener set up to listen for the submit of a button for a file getting uploaded. I create an upload task to get the number of bytes being uploaded so that I can let larger files upload before the page is refreshed and the user is presented with their newly added data.

```
8  uploadForm.addEventListener('submit', (ex) => {
9  |   ex.preventDefault();
10 |   //This variable is created as for upload task of a file
11 |   var uploadTask = storageRef.put(file);
12 |   uploadTask.on(firebase.storage.TaskEvent.STATE_CHANGED,
13 |     function (snapshot) {
14 |       //This is where we get the task progress of a file getting uploaded. I have it login
15 |       //the console where we would be able to see the number of bytes getting uploaded and al
16 |       //total number of bytes that are to be uploaded.
17 |       switch (snapshot.state) {
18 |         case firebase.storage.TaskState.PAUSED: // or 'paused'
19 |           console.log('Upload is paused');
20 |           break;
21 |         case firebase.storage.TaskState.RUNNING: // or 'running'
22 |           console.log('Upload is running');
23 |           break;
24 |       }
25 |     }
26 |   );
27 | }
```

One last feature I would like to mention is how I manage user sessions. Let's say for example a user is logged into a computer in an internet café and they access DocSafe on this computer to download a certain piece of data they need. The user closes the browser without clicking logout on the application. I want to make sure that the user is forced to logout after a session ends so that I can ensure that the next person going to use this computer cannot access their account as they will be logged out.

I managed to achieve this by implementing the following code statement.

```
1
2  firebase.auth().setPersistence(firebase.auth.Auth.Persistence.SESSION)
3  .then(() => {
4    return firebase.auth().signInWithEmailAndPassword(email, password);
5  })
6  .catch((error) => {
7    // Handle Errors here.
8    var errorCode = error.code;
9    var errorMessage = error.message;
10 });
```

Looking at line 2 I am setting the existing and future authentication states to be persisted in the current session only. So by doing this it means that when a user closes the browser or window that the sessions will be cleared of any existing states which forces the user to be logged out for when they forget to do so.

Line 4 is where I return my sign in statement which is prompting the user in a new session to log in again. This helps with the prevention of session hijacking and increases security with in regard to session management.

The last topic of implementation I would like to touch on is the use of HTTPS and SSL certificates. I am using the firebase CLI to deploy my application onto the internet to demonstrate HTTPS in action. HTTPS is known as the secure version of the HTTP protocol, it guarantees a secure connection between device and server. Firebase will automatically set up an SSL certificate for all custom and assigned domains which is exactly what I need in order to gain another extra layer of security. *"Firebase automatically provisions SSL certificates for all your domains so that all your content is served securely"* (Firebase Hosting, 2021)

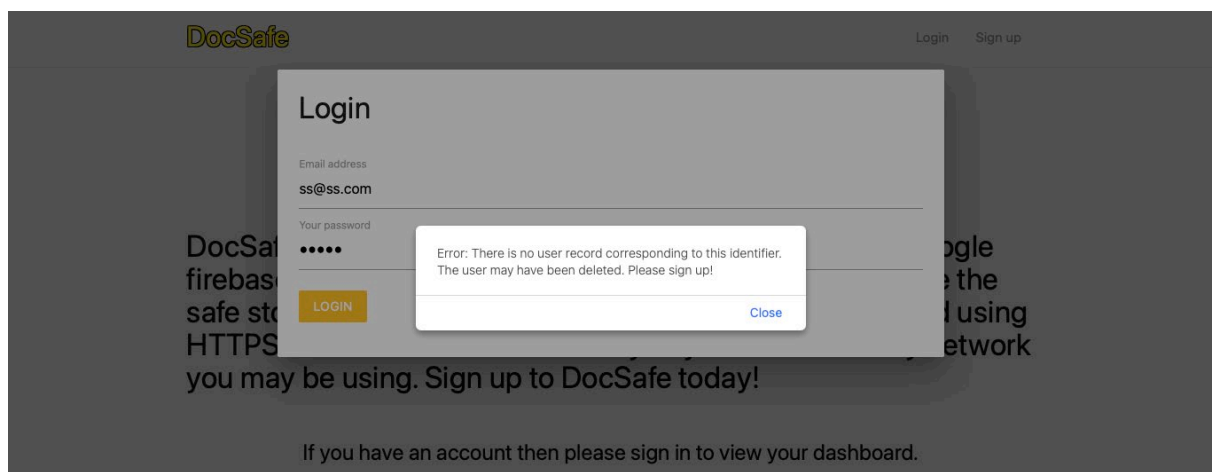
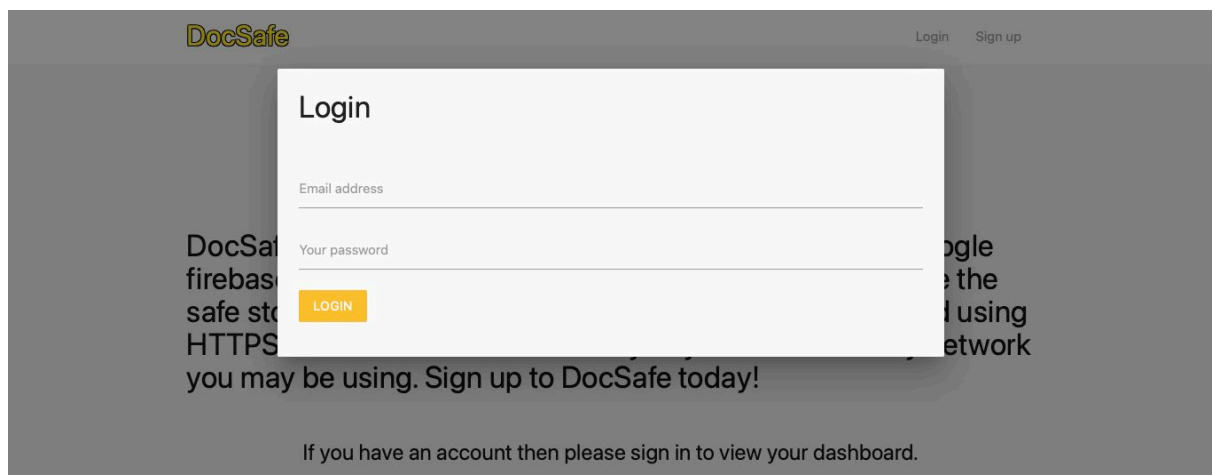
That about wraps it up for the main functionality within my project, I will be able to demonstrate this code working in the next section and live through my video presentation. I feel I have implemented all the necessary algorithms and methodologies for the development of my project, I am happy with how the project runs and I am glad I was able to achieve what I had set out for myself.

2.4 Graphical User Interface (GUI)

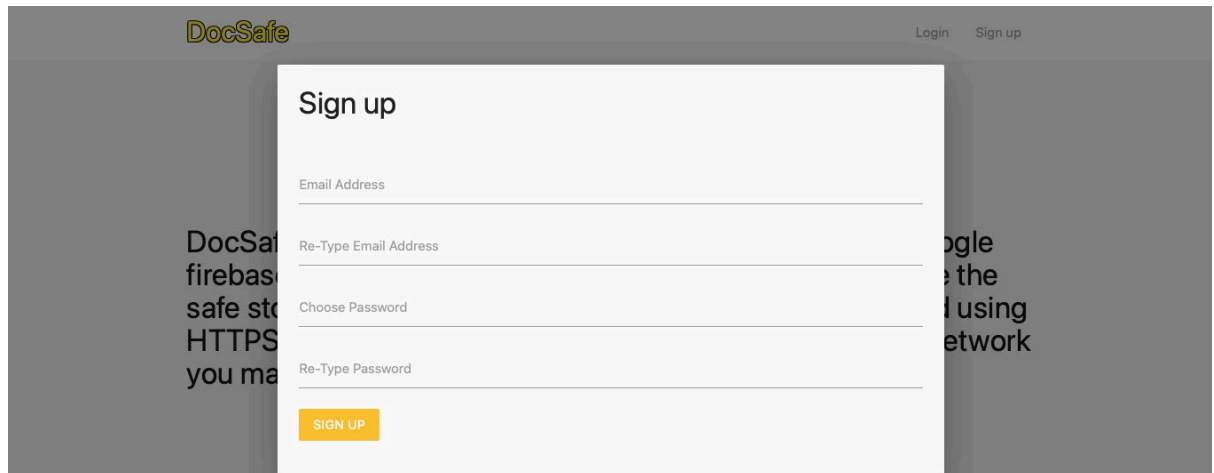
Here is a snippet of the landing page for DocSafe. This is what the user will see first when they access the DocSafe domain, from here a user can decide whether they want to log in or sign up.



This is what is shown when a user clicks Login. A user will need to already have an account created to get passed this step. If a user enters information that is not stored within firebase authentication then an error will be thrown to let the user know that they have not entered valid credentials and are to try again. Error message shown in second screenshot.

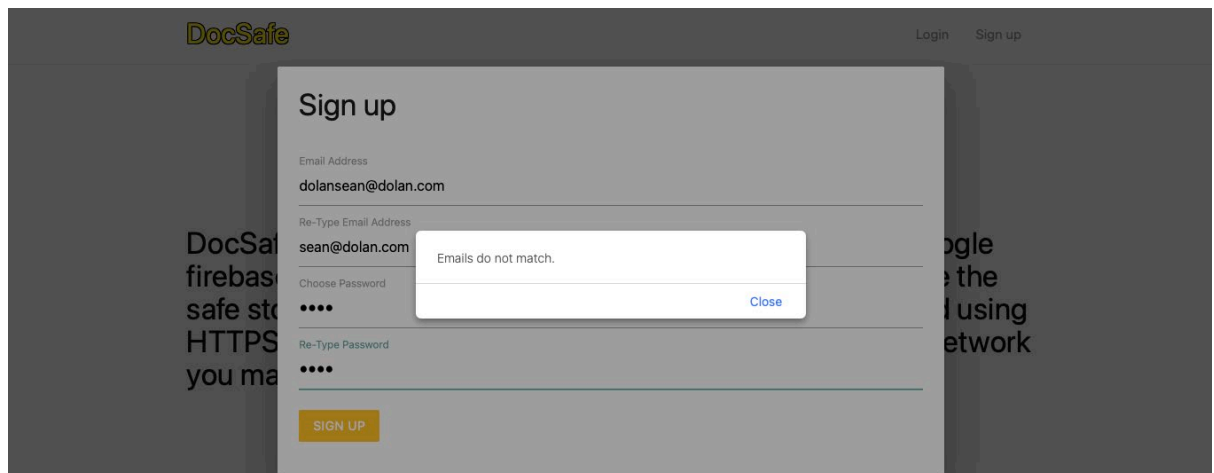


Here is what shown to a user who wants to sign up for DocSafe. They need to enter a valid email twice and a strong password twice to ensure there are no errors when signing up.



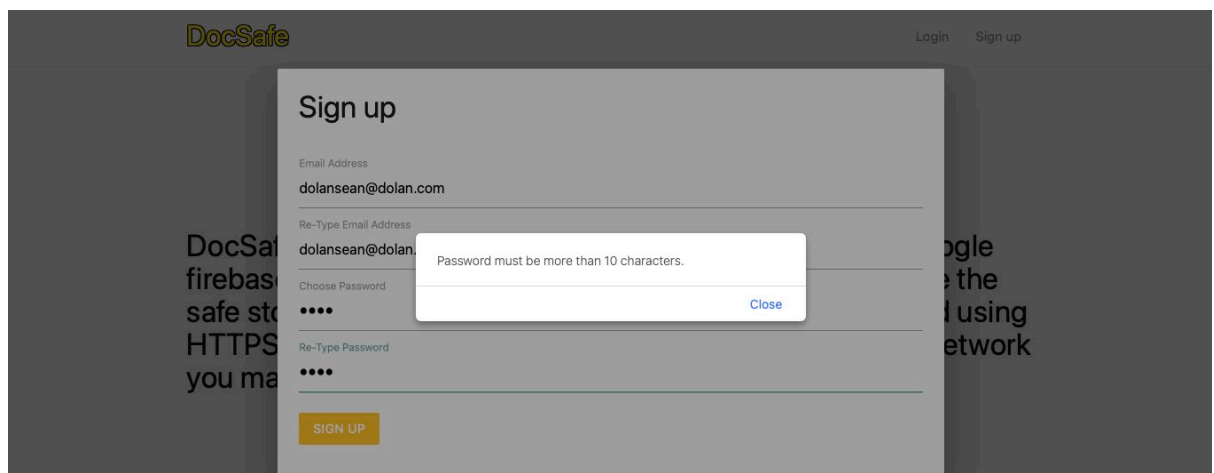
The screenshot shows the DocSafe sign-up form. At the top left is the DocSafe logo. At the top right are links for 'Login' and 'Sign up'. The form is titled 'Sign up' and contains four input fields: 'Email Address', 'Re-Type Email Address', 'Choose Password', and 'Re-Type Password'. Below the fields is an orange 'SIGN UP' button. The background is a dark grey with some text visible on the left and right sides.

Alert shown when emails do not match.



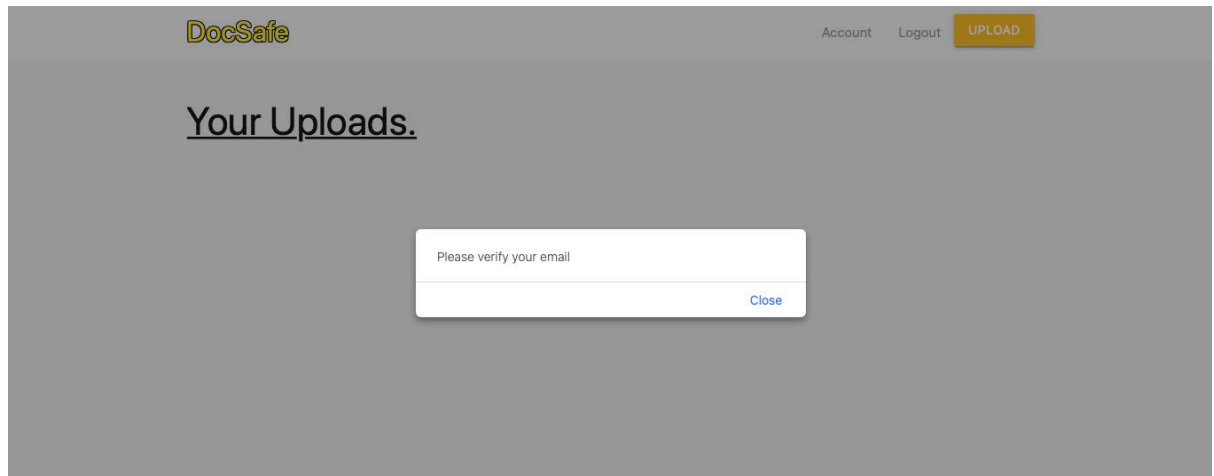
The screenshot shows the DocSafe sign-up form with an alert. The 'Email Address' field contains 'dolansean@dolan.com' and the 'Re-Type Email Address' field contains 'sean@dolan.com'. A white alert box with a blue border is centered over the form, displaying the message 'Emails do not match.' and a 'Close' button. The 'Choose Password' and 'Re-Type Password' fields are filled with dots. The 'SIGN UP' button is visible at the bottom.

Alert shown for password checks – I have implemented more password checks but I feel it is not necessary to show the rest of them in this document. The alert will show up the same for passwords that don't contain an uppercase/lowercase character and a number.

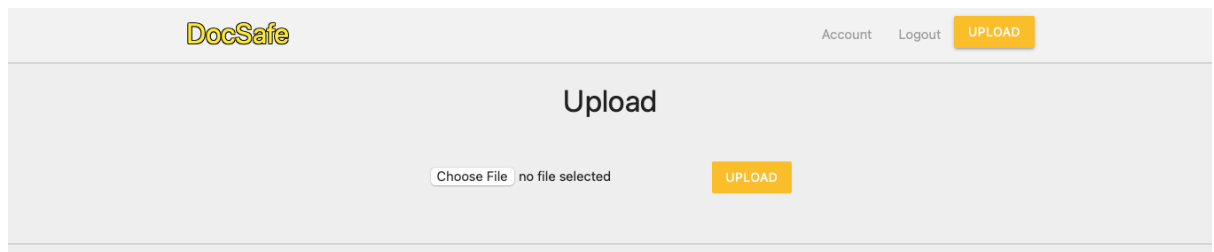


The screenshot shows the DocSafe sign-up form with a password length alert. The 'Email Address' and 'Re-Type Email Address' fields both contain 'dolansean@dolan.com'. A white alert box with a blue border is centered over the form, displaying the message 'Password must be more than 10 characters.' and a 'Close' button. The 'Choose Password' and 'Re-Type Password' fields are filled with dots. The 'SIGN UP' button is visible at the bottom.

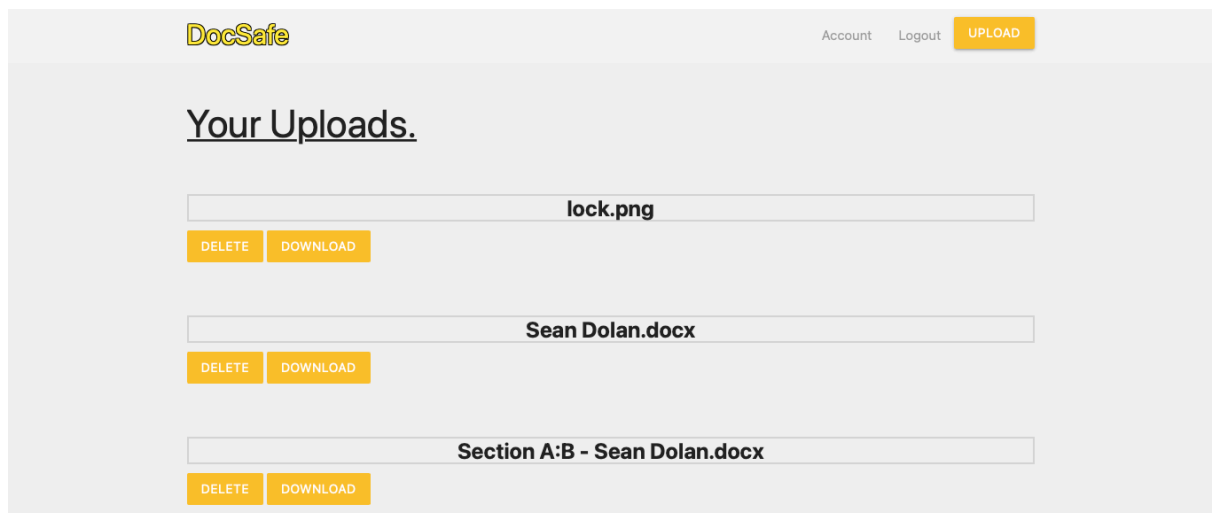
Here is the alert shown for users that want to upload data to their account without verifying their email beforehand. They need to verify their email before doing this.



Here is screenshot of what happens when a user who is verified presses the upload button.



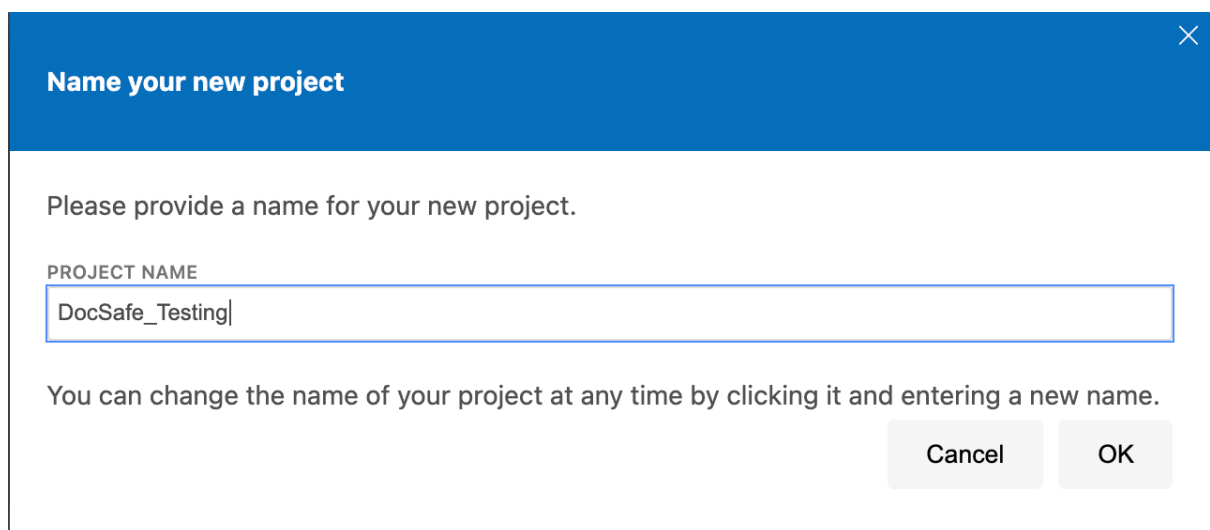
Here is an example of how a dashboard will look for a user that has uploaded a couple of documents. We see the download and delete button under each document to ensure a user doesn't delete the wrong file.



2.5 Testing

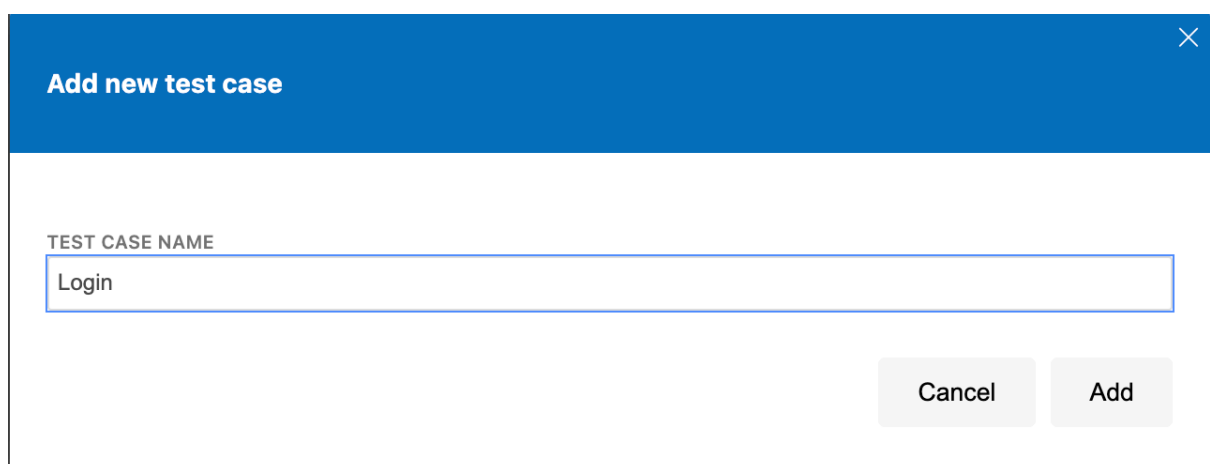
For testing my web application I will be using Selenium IDE. Selenium will record and playback any tests you would like to perform on your web application. *“The new Selenium IDE is designed to record your interactions with websites to help you generate and maintain site automation, tests, and remove the need to manually step through repetitive tasks.”* (About Selenium, 2021).

To get started with Selenium you need to add the IDE extension into your Google Chrome browser. You start up the plugin and create a project that you will be conducting the testing under.



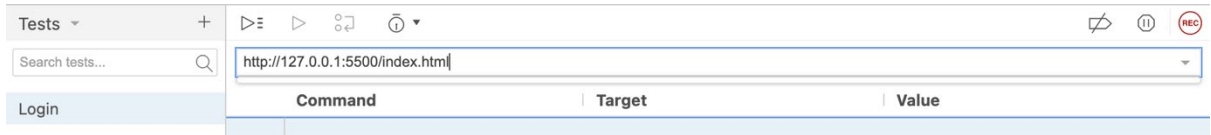
The screenshot shows a dialog box titled "Name your new project" with a close button (X) in the top right corner. The main text reads "Please provide a name for your new project." Below this is a text input field labeled "PROJECT NAME" containing the text "DocSafe_Testing". A note below the field states: "You can change the name of your project at any time by clicking it and entering a new name." At the bottom right, there are two buttons: "Cancel" and "OK".

Once the project is created, we now need to create a test case that will be related to the type of test we are trying to run. Below I am creating a test to run for my login for a user.

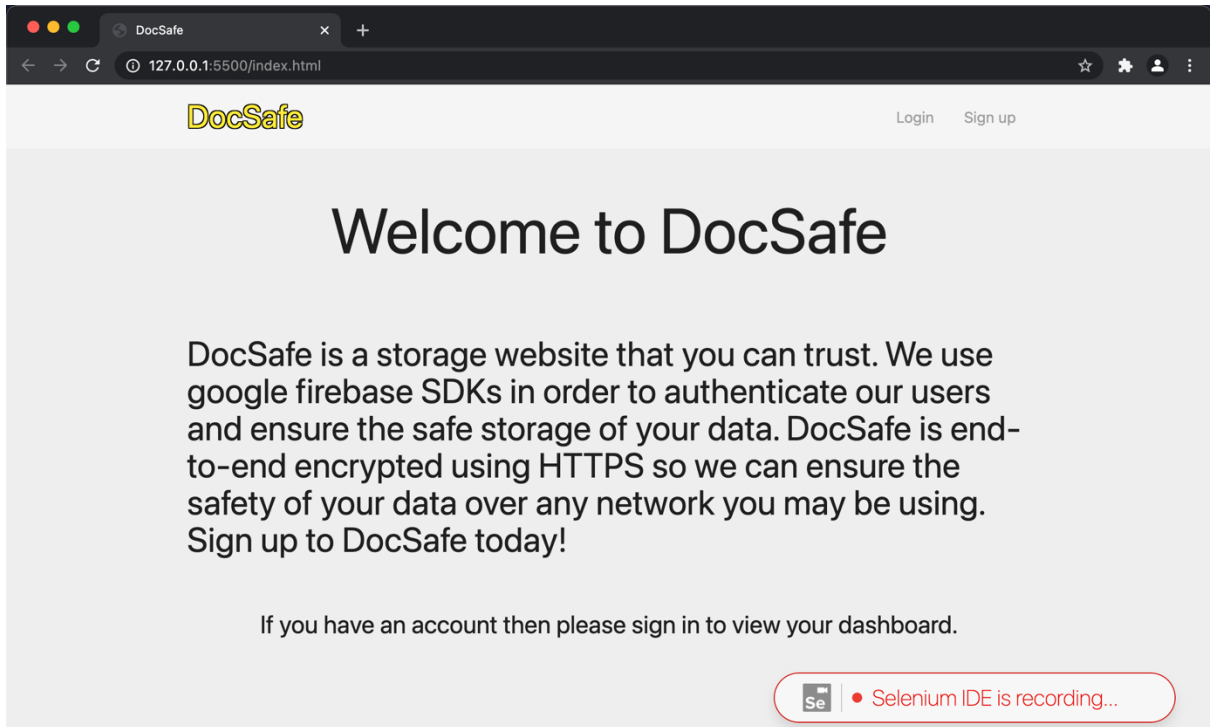


The screenshot shows a dialog box titled "Add new test case" with a close button (X) in the top right corner. The main text reads "TEST CASE NAME" above a text input field containing the text "Login". At the bottom right, there are two buttons: "Cancel" and "Add".

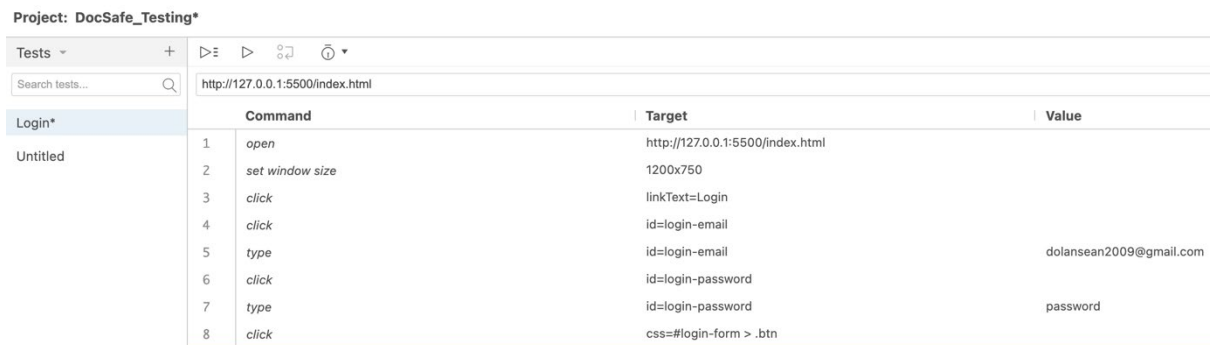
To get started with the test I need to get the URL of my website, for testing demonstrations I am hosting my project locally. Next all we have to do is press the record button on the top right corner and this will open a new page for where Selenium will start recording my steps I will take in order for me to run a test on them later.



This is the webpage that is opened up by Selenium IDE and we can see on the bottom right of the screenshot that it is recording the browser to take in any steps I want to perform for test purposes. What I am going to do next is login to my website with credentials I have already created and Selenium is going to record these steps for me so that I can run a test on it after.



We can see below are the steps recorded by Selenium IDE. What I need to do next is to run the test to see if Selenium runs into any sort of issues along the way.



We run the test by pressing the small play button at the top of the page.



Below is the results Selenium IDE has returned to me from running the current test. We can see that Selenium has successfully ran through the steps it has recorded from when I manually done the run through of logging a user in. This tells me that my login is working as it should and this test has been successful.

Log	Reference
Running 'Login'	
1. open on http://127.0.0.1:5500/index.html	OK
2. setWindowSize on 1200x750	OK
3. click on linkText=Login	OK
4. click on id=login-email	OK
5. type on id=login-email with value dolansean2009@gmail.com	OK
6. click on id=login-password	OK
7. type on id=login-password with value password	OK
8. click on css=#login-form > .btn	OK
'Login' completed successfully	

Now that I have explained how Selenium works and how the test results are shown I will now show the test results for other features of my web application.

Register Test

Log	Reference
4. click on id=signup-email	OK
5. type on id=signup-email with value dolansean2008@gmail.com	OK
6. click on css=#signup-form > .input-field:nth-child(2) > label	OK
7. type on id=signup-email-retype with value dolansean2008@gmail.com	OK
8. click on css=.input-field:nth-child(3) > label	OK
9. type on id=signup-password with value PasswordPassword1	OK
10. click on css=.input-field:nth-child(4) > label	OK
11. type on id=confirm-password with value PasswordPassword1	OK
12. click on css=.btn:nth-child(5)	OK
13. assertAlert on A verification email has been sent to you. Please verify your email so you can use our services!	OK
'Register' completed successfully	

Email not verified test

Log	Reference
3. click on linkText=Login	OK
4. click on id=login-email	OK
5. type on id=login-email with value dolansean2008@gmail.com	OK
6. click on id=login-password	OK
7. type on id=login-password with value PasswordPassword1	OK
8. click on css=#login-form > .btn	OK
9. Trying to find id=upload-button...	OK
10. assertAlert on Please verify your email	OK
11. click on id=upload-button	OK
12. assertAlert on Please verify your email	OK
'Email Not Verified' completed successfully	

Email is Verified – User will not be able to open upload modal is not verified.

Log	Reference	Description
Runs: 1	Failures: 1	
4.	click on id=login-email OK	
5.	type on id=login-email with value dolansean2008@gmail.com OK	
6.	click on id=login-password OK	
7.	type on id=login-password with value PasswordPassword1 OK	
8.	click on css=#login-form > .btn OK	
9.	Trying to find id=upload-button... OK	
10.	click on id=fileButton OK	
11.	type on id=fileButton with value C:\fakepath\credWire.png Failed: {"code":-32000,"message":"Not allowed"}	

'Email Is Verified' ended with 1 error(s)

Now this test failed due to the fact that Selenium does not have any access to my system so it cannot choose the file I uploaded in my steps for this test. But the user was able to open up the upload modal so that indicates that the email is verified.

Logout Test

Log	Reference
4.	click on id=login-email OK
5.	type on id=login-email with value dolansean2008@gmail.com OK
6.	click on id=login-password OK
7.	type on id=login-password with value PasswordPassword1 OK
8.	click on css=#login-form > .btn OK
9.	Trying to find id=logout... OK
10.	mouseOut on id=logout OK
11.	click on id=logout OK
12.	click on linkText=Login OK
13.	click on css=.modal-overlay OK

'Logout' completed successfully

Password Check Test

Log	Reference
10.	click on css=.input-field:nth-child(4) OK
11.	type on id=confirm-password with value password OK
12.	click on css=.btn:nth-child(5) OK
13.	assertAlert on Password must contain one uppercase character. OK
14.	click on id=signup-password OK
15.	type on id=signup-password with value passwordP OK
16.	click on id=confirm-password OK
17.	type on id=confirm-password with value passwordP OK
18.	click on css=.btn:nth-child(5) OK
19.	assertAlert on Password must contain at least one number. OK

'Password Check' completed successfully

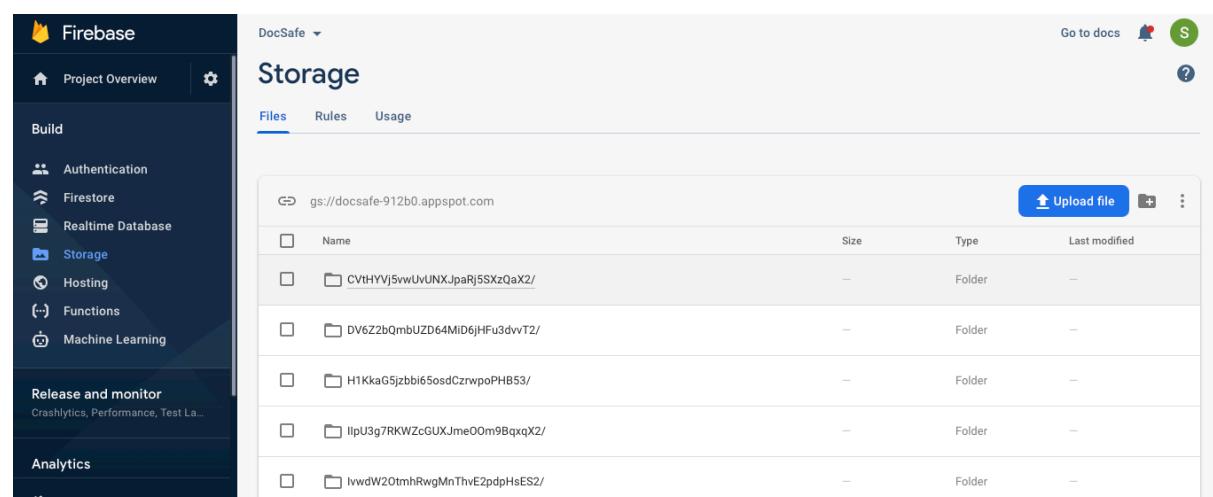
For this test I was making sure that the password checks that are put in place are working as they should. You can see on line 13 that an alert is shown saying the password needs to contain an upper case letter and then another alert on line 19.

2.6 Evaluation

When I look back at the start of this report, I set 4 aims on what I wanted to achieve in this project. I needed to develop and research an idea that would address the complex issues of cyber security and how I plan to tackle these aims within my project. I wanted to be able to use emerging technologies and methodologies in order to succeed in what I set out to do and make my idea innovative and approachable. When it comes to evaluating these key points I feel as if I have addressed them to the best of my ability. If we just go back to the 4 aims I set out at the start, secure storage, communication, authentication/validation and session management. I feel I was able to address these topics of concern to a high standard of development.

When we take a look back at secure storage, I was able to design and create a system through Google Firebase that will store user data in such a way that only the user that matched the unique ID of their file path could actually access their data. By creating this sort of system it eliminates the concern and possibility of other users being able to access data that is essentially assigned to you.

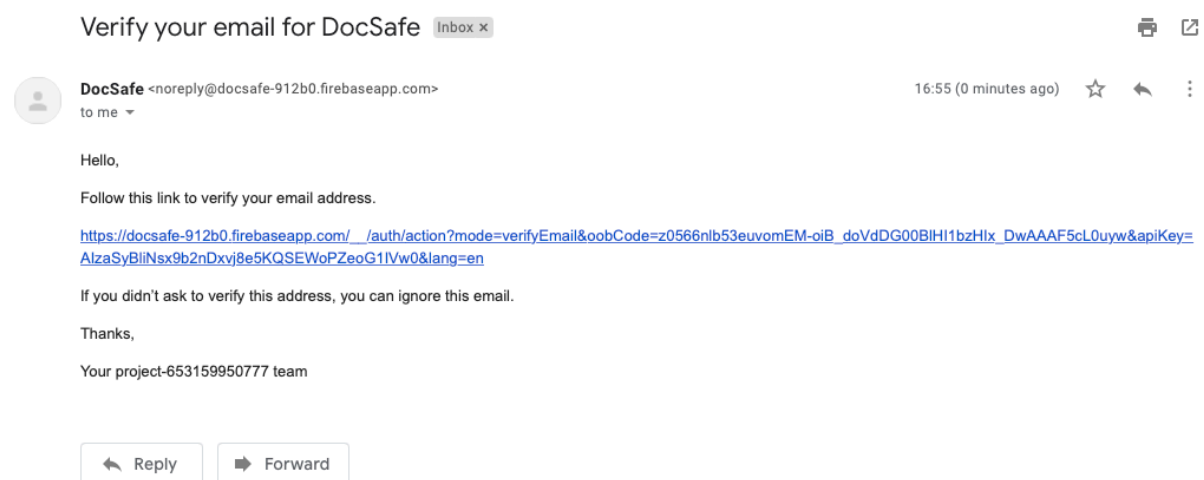
I will show an example of how my firebase storage deals with this type of file path below:



The above image shows a list of different user IDs that are created when a user makes their first upload. I have told firebase to create this path for users so that data that is related to them is uploaded to a file path unique to the current user that is signed in.

Now having this secure storage method above is great but we also need to ensure that a user is signing up securely and also logging in securely too. With the help of Firebase Authentication I was able to develop and address the complexity of working authentication in a safe and secure manner. Firebase authentication is a fantastic tool made for developers and it can be used in many different ways, I was able to use firebase authentication to sign my users up, verify their emails and also log them in and out. Firebase authentication is basically making use of token generation and what token generation means is that we are taking identifiable, confirmed user data from an authorised source and then passing securely through to firebase.

I wanted to be able to address two-factor-authentication when it came to a user logging into the application, but, unfortunately this service turned out to be a paid service and extremely cost-effective when put up against my project. Two-factor authentication would have really set the bar high for extra security measures put in place but thankfully with the use of Firebase Authentication, it's pretty secure without it. I set up an algorithm that users will be put through in order to enhance the level of secure authentication, all users that sign up for the application will need to verify their email before they can upload any data to firebase. I have implemented an email verification check for all users, they will be sent an email as soon as they pass the registration process and this email will include a link to verify their email. An example is shown below.



I have also made sure that when a user is signing up to DocSafe that they are confirming their emails by re-typing the email into the signup form and that they are also creating a strong password. I have put in a good degree of password validation checks into my project and this is to ensure a strong password that is hard to guess. A user must enter a password that contains 10 or more characters, once uppercase character, once lowercase character and at least one number. The idea behind these password checks is to prevent an attacker from essentially guessing the passwords and it also ensures that users do not create an account with the famous password, "password". Hackers can actually get a long list of common and compromised passwords and initiate a brute force bot attack on a website that may have leaked a list of user emails, this type of attack will cycle through the emails and attempt to essentially guess the password of a user.

Firebase will deal with all the encryption end of things when it comes to passing through plain text to the server. All passwords that are stored in firebase are passed through a hashing algorithm to ensure there is proper integrity for storage of user passwords. If we combine the password validation checks and the password hashing algorithm, it makes things very difficult for an attacker to try and gain entry to a user's account. I feel that I have addressed authentication and validation to a good standard even with the concerns of not having two-factor authentication in place, I am still confident that my authentication technique is strong and reliable.

As well as using firebase for the sheer backend service of my project, I also made the decision to use it for hosting my application with it too. My project is built up of HTML + JavaScript + CSS and these are all known as static assets and firebase hosting is built for hosting these types of projects. Firebase hosting is said to work straight out of the box and set for ease of use, it's secure and super easy to use. I stated in my aims that I wanted to ensure that I tackle the task of implementing secure communication to my application and how I managed to do was with the use of SSL certificates and HTTPS. Firebase hosting allowed me to serve my content over a secure connection as they offer an SSL certificate that requires zero configuration straight into the deployment of my web application. Google provide a step by step guide on how to approach the deployment of your web application onto their services which is a fantastic feature they have to offer.

My final aim that I wanted to achieve was session management and I feel I have addressed that to an appropriate standard. I managed to ensure every time a user closes their session, a new session is created. I implemented a system that works in such a way that each time firebase is accessed by any user that it is a fresh new sessions with almost zero traces of the previous user. Session management is an on-going issue for many developers and businesses so it is really important that it is addressed properly, I wanted to really make sure I was ticking the boxes when it came to the safety and security of the user data. Using firebase as my backend service has really allowed me to think outside the box when it came to the development of this project, I was successfully able to achieve the aims I have set out at the start of this document and I feel I have addressed the sheer complexity of how important cyber security is.

I would have like to have worked on the front-end design of this application a lot more, I created a basic user interface that suited the needs of my backend functionality. The overall design of my project is set out to do what suited the user, I feel I could have added much more to it in regards to usability and UI. The design of any application is important as it is what stands out to a customer and really sets an application up for good competition.

3 Conclusions

DocSafe is an incredibly easy to use and safe environment that users can trust. Developing this project and doing the relevant research has been very difficult and challenging as there was so many aspects I needed to take into account. Storage of data and customer information can be a difficult task to tackle as it is nearly impossible to 100% guarantee that your data is safe, hackers will always find new and creative ways to exploit new technologies and the best way I can tackle this is to keep up to date with new ways of keeping the bad people out. I feel my project has plenty of commercial potential as there is working functionality put in place that tackle these types of cyber security aspects to a good degree. I am happy with the overall result of how my project has turned out, it was definitely a difficult project to get developed but I personally feel I have addressed my issues well and achieved what I had set out to do in the beginning.

Advantages

- Secure storage
- Secure communication
- Working and Secure Authentication
- Super Easy To Use / User Friendly / Distributed
- A Trusted Backend Service

Disadvantages

- No Two-Factor Authentication
- Not elegant enough
- I could use more experience in firebase
- Could do with better front-end

Limitations - I feel the project is limited in regard to authentication. I really wanted to address two-factor authentication and demonstrate it within my project but it turned out to be cost-effective. If I went ahead with my original to just develop this project as a standalone phone application then I would have really limited myself in terms of what devices can access the application and it wouldn't really make sense to do so. I am glad I took the approach I did with this project as it did not set me much limitations in regard to usability.

4 Further Development or Research

I would have 100% implemented multi-factor authentication if I had more resources, this would have added multiple layers of security to my project. I would have liked to develop mobile applications that would allow users to easily access their data rather than having to type in a URL in order to access DocSafe. I would have liked to spend more time on the front-end side of things to make the application more appealing and to add some more functionality to the project to try and make it stand out from the rest of the storage provider competitors. I would spend more time and resources on marketing the project to try and boost its commercial potential and allow users to safely and freely store data that is important to them.

5 References

D'Amico, A., 2019. *Broken Authentication & Session Management / AppSec*. [online] Code Dx. Available at: <<https://codedx.com/blog/broken-authentication-and-session-management/>> [Accessed 14 May 2021].

Firebase. 2021. *Firebase Hosting*. [online] Available at: <<https://firebase.google.com/docs/hosting>> [Accessed 13 May 2021].

Firebase. 2021. *Firebase*. [online] Available at: <<https://firebase.google.com>> [Accessed 13 May 2021].

Moroney, L., 2017. *The Definitive Guide to Firebase*. 1st ed. Berkeley, CA: Apress, p.1.

Naylor, D., Finamore, A., Leontiadis, I., Grunenberger, Y., Mellia, M., Munafò, M., Papagiannaki, K. and Steenkiste, P., 2014. The Cost of the "S" in HTTPS.

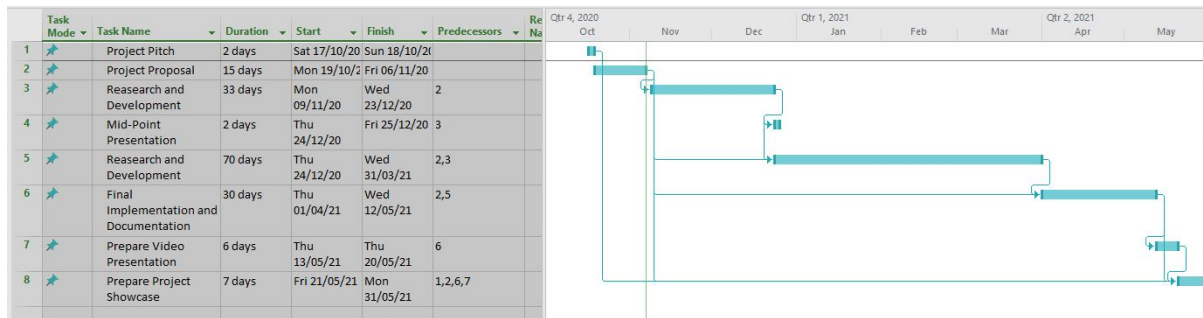
Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies, p.1.

Selenium.dev. 2021. *About Selenium*. [online] Available at: <<https://www.selenium.dev/about/>> [Accessed 14 May 2021].

Tunggal, A., 2021. *Why is Cybersecurity Important?* / *UpGuard*. [online] Upguard.com. Available at: <<https://www.upguard.com/blog/cybersecurity-important>> [Accessed 13 May 2021].

6 Appendices

6.1 Project Plan



**** I was unable to access Microsoft Access to update my project plan to meet the new deadlines due to the ransomware attack on the college.**

6.2 Reflective Journals

September

- This is my first month back in college so everything is a bit new and overwhelming with starting during a pandemic. I am but unsure on what to develop for my final year project as it has to have something to do with cyber security which is all a bit new to me. I need to research and come up with a project idea by the middle of October so until then I will be researching type of projects to pitch and come up with some new ideas.

October

- This month my project pitch was due. I was still a bit unsure on what I wanted to develop for my project so I made a up draft project idea that was intended for personal trainers. I want to develop an application for gyms and personal trainers to deliver an online coaching service during the pandemic. I wasn't sure on this idea as it was not really related to my programme specialisation and I wanted to see what ideas my supervisor would advise me to do. Other than this idea I was thinking about making a mobile application of some sort as I would like to use Android Studio to develop my knowledge of mobile application development.

November

- This month I have received feedback for my project pitch. I was told that my proposed project will not be accepted a good selection for my final year. This was expected by me as I knew I was still on the fence on what to develop and how to go about it. My supervisor gave me some good feedback and ideas on what I could possibly develop. I came up with the idea of a secure storage application for your mobile device. I wanted to develop a mobile application that would allow users to safely store data with google firebase as a backend service. This new idea would touch on cyber security quite a bit and it was later approved by my supervisor and ready to be added to my project proposal document due this month too.

December

- December was quite hectic for me. We had a lot of assignments due for other modules as there was plenty of deadline but in place. I had to sacrifice some assignments in order to give my mid-point presentation some time to develop and submit. My mid-point presentation was basic and it did not include a project prototype which I feel would drag my grade down a good bit. I did not give my project enough time this semester as I was too focused on trying to get the heavy workload of other modules done, I feel we had a lot of stuff due and simply did not leave myself any time for my project what so ever.

January

- This month I received feedback from my mid-point presentation. I got some negative feedback and concerns about how I am developing this project. My supervisor pointed out that I am limiting myself with an android application and that I should consider developing the same project idea but to make it more diverse and increase the usability. He told me that I should make a web application instead of the standard android application as it didn't really make much sense to limit the application to a smart phone. I took on board what he suggested and decided to go with his newly proposed Idea.

February

- I spent most of February researching how I was going to go about developing my new changes to my project. I took my supervisors advice on board and I wanted to shift my project into a web application that is accessible on all devices. I found some useful information on how I should develop this application and I decided to code an application from scratch using HTML and JavaScript. I will still be using Google Firebase as my backend service that will help my application function and I plan to do this with JavaScript.

March

- Most of March was spent working on the development stages of my project. I had a lot of trouble with designing a framework that would touch on my cyber security specialisation. I needed to research and come up with a way that I can implement this into my project. Right now I have focused on getting the core functionality of this application up and running with Google Firebase. I have been following a couple of tutorials on how to use firebase and JavaScript together but it is still difficult to find the information I need that relates to my project.

April

- I am still working on the development of my project and struggling to get the core functionalities working. I managed to get user authentication set up and I plan to expand on this more once I get my core functionality working. I have made a basic web page that works well with firebase and I have just been adding to the backend code of my project in order to get as much functionality in as possible.

May

- I managed to get the core functionality of my project working and I was able to achieve the aims I had set out for myself in my project report. I have a fully working project that has some commercial potential too. I was unable to add in some features I had set out for myself so I am bit annoyed with that. However, I feel I have done the project well and still managed to get a working project with some functionality that addressed the complex issues that come with cyber security. I am happy with the overall outcome of my project as it has been extremely stressful to develop and report but I managed to get there in the end. I am excited to be finishing college as I have been studying for 5 years so I hope that the work I had put in this year has paid off and I get the results I am expecting!

6.3 Other materials used

DocSafe

06/11/20

BSc in Computing

Cyber Security

2020/2021

Sean Dolan

X17467042

x1746702@student.ncirl.ie

Contents

1.0	Objectives.....	41
2.0	Background	42
3.0	Technical Approach.....	43
4.0	Special Resources Required	43
5.0	Project Plan	44
6.0	Technical Details	44

7.0	Evaluation	44
8.0	Invention Disclosure Form (Remove if not filled)	44

1.0 Objectives

The purpose of this idea is to give users some relief knowing that some of their documents that include sensitive data such as personal information, company information and anything that could be held for ransom is in a secure place with no possibility of an attack.

I want to create an application where a user can add documents or images to this application that they feel needs to be kept locked away and for only their own eyes to see it. For a user to be able to use this application they would need to set up a Fingerprint ID or Face ID alongside two-factor authentication. When the user wants to access the application they need to use their fingerprint or facial recognition alongside the two-factor authentication code that will be either sent via text message or Google Authenticator.

I would like to make this application as secure as I possibly can to prevent attackers from being able to access this data in this app. To make this application secure there is a number of strategies I could implement.

As this app would be communicating with other apps much like Microsoft Word and Photos I would need to make sure this communication is safe. I can do this by enforcing safe communication, when you safeguard data that you exchange between your app and this app you improve the app's stability and protect the data being sent.

As mentioned before you would need to enter some credentials before showing sensitive information, when requesting credentials from users you can ask for either a PIN/Password or biometric credential such as face recognition and fingerprint recognition.

The most important aspect of this application is that the data is stored safely, I can do this by using the device's internal storage. Other apps should not be able to access this sensitive data and as an added security measure, when the user uninstalls this app it deletes all files that the app saved within the internal storage.

Before deployment of this application I would need to ensure that all app dependencies like libraries, SDKs and other dependencies are up to date. I would also intend to store any non-sensitive data in cache files.

I would like to incorporate a feature that in the case of a user losing their phone/device that they would be able to wipe their details/data from this device.

2.0 Background

The storage of sensitive data has always been an inconvenience for many users and organizations so the idea behind this application is to give users a bit of clarity when it comes to storing their information that is meant for only them.

This application falls under a category of network security and that is access control, access control is very important fundamental of cyber security Access Control is a very important security technique in the IT industry and in-fact many other organizations too. Access Control regulates who or what can view, use or modify resources in a computer network, and it is a fundamental concept in cyber security that will minimize risks for the business or organization.

There are generally two types of access control, physical and logical. Physical control would generally focus on access limitations to college campuses, office buildings, meeting rooms and some physical IT resources. Logical control is more aimed at the computer network, it will limit connections on the computer network as well as setting permissions on system file locations and work to protect sensitive data. This application will focus mainly on the more logical approach rather than the physical one but it will still include some aspects of physical access control.

Access control systems for most organizations would generally perform identification, authentication and authorization of users on computer systems and entry into a building or room. This is done by evaluating the required login credentials such as; username and password, personal identification numbers (PIN), security tokens or other authentication factors like biometric scans. A very common thing to see in more modern times is the use of Multifactor authentication (MFA) or as most people would recognize two factor authentication (2FA). This is when a login requires two or more authentication factors, for example, some computer systems would prompt you to enter your personal password and then ask for a secondary password that would be generate from a key fob, a key fob is a small security hardware device with built-in authentication that displays a randomly generated access code that changes periodically. The reason behind MFA and 2FA is because it is seen as part of a layered security defence to protect access to sensitive company data and prevent system attacks on organizations.

The main and most important goal behind access control is to minimize the risk of unauthorized access to physical and logical systems as much as humanly possible. With access control being a fundamental of cyber security it plays a huge part in security compliance programs, these programs and procedures are put in place to ensure that security technology and access control is very well practiced and thought-out to protect confidential company data.

Access Control systems should be complex and challenging to manage in a dynamic IT network that would generally include cloud services and on-premises systems. With hacking becoming a very common problem in the 21st century and with many high-profile breaches, most organizations have steered away from single-sign on (SSO) and moved onto more complex and safer ways of access control for cloud and on-premises environments.

3.0 Technical Approach

This project will be an application that will be solely developed on android studio, I have used android studio in the past for previous college projects but never in the way I intend to use it for this application. When using android studio in the past it's only really been for basic applications with basic requirements but for the app I would like to branch out more and explore the many technologies that android studio has to offer.

When looking more into android studio it offers plenty of information on how to make an application developed with this software on their forums, it provides source code on how to do certain measures of security.

I will use the functions of google APIs for the use of google authenticator so that I can enable a safe and reliable multi-factor authentication.

I want to make this application as user friendly as possible and easy to use, I would use YouTube tutorials for tips on how to design a good GUI that would appeal to the human and help the user navigate the application as seamlessly as possible.

I would like this application to be developed for both Android and IOS however, Apple make it difficult for applications to be posted onto their store for users to use but with Android and the use of the Google Play Store they make it easier to trial applications on your personal device to ensure good testing. I would work mainly with Android and hope that in the future that I could implement this application onto IOS app store.

In relation to make this application as appealing as possible I will make use of photoshop to create and use a suitable application logo to go with this app when it is launched to the google play store. I have never used photoshop before so I will be interested to see how I get on with this.

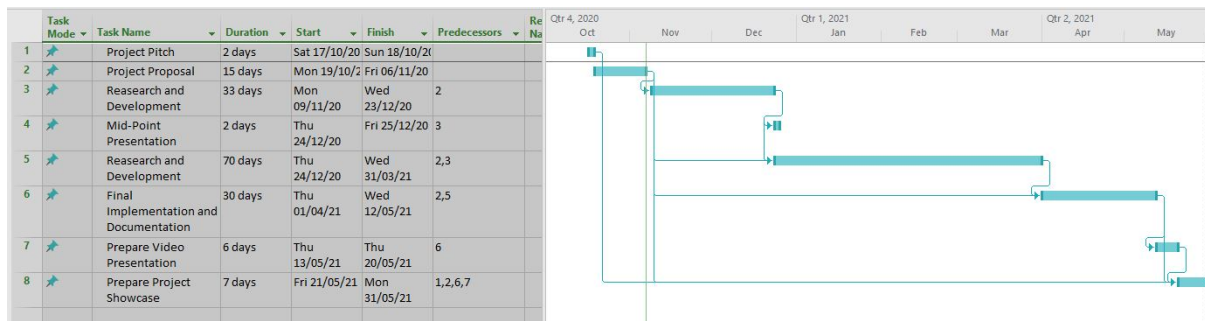
As I mentioned above in the application objectives, I would like for users to be able to wipe their data off a device they have misplaced or lost. To do this I would need to think of way that I could let the user access their account from another device so that they could deactivate the device they have lost linked to the account thus wiping any sensitive data from said application.

Something I would also like to implement is the use of an RFID card that a user could keep in their home or wallet as another form of security. The app will prompt the user to present their RFID card to back of their phone in order to gain access to the app itself. This is just an idea and I may not be able to implement this to my application just yet but it's something I would definitely look into down the line.

4.0 Special Resources Required

RFID card, key fob.

5.0 Project Plan



*Please note that this project idea was only approved late Friday afternoon so I will be remodelling this Gantt Chart.

6.0 Technical Details

This application will be fully developed using android studio, android studio uses Java and C++.

7.0 Evaluation

I will start by building a basic application that when launched it will show an image saying 'coming soon' just as a test to make sure everything works as it should. I will then move onto the implementation of how to add files to the application using the devices internal storage.

This application is something I want to develop for myself, I would love to be able to store information that is important to me and me only safely and as easy as possible. Knowing that I personally built an app that I can trust and not have to worry about is a good thing.

8.0 Invention Disclosure Form (Remove if not filled)

Please fill in the following sections, if you think your idea is innovative:

1. Title of Invention

DocSafe

2. Inventors

Name	School/Research Institute	Affiliation with Institute (i.e., department, student, staff, visitor)	Address, contact phone no., e-mail	% Contribution to the Invention
Sean Dolan	National College of Ireland	Student	X17467042@student.ncirl.ie 0852835067	100%

3. Contribution to the Invention

Each contributor/potential inventor should write a paragraph relating to his/her contribution and include a signature and date at the end of the paragraph.

I will develop the whole application alone with no help from others.

4. Description of Invention

(Please highlight the novelty/patentable aspect. Attach extra sheets if necessary, including diagrams where appropriate). What is novel, the 'inventive step'? For more information on patents, please look at <http://www.patentoffice.ie/en/patents.aspx>

This app will be a safe and secure way for users to be able to enjoy their day knowing that their data is protected in this app.

5. Why is this invention more advantageous than present technology?

What is its novel or unusual features? What problems does it solve? What are the problems associated with these technologies, products or processes? Explain how this invention overcomes these problems (*i.e.*, what are its advantages).

This an app for personal use only and lets you steer away from big companies like apple and google. Many of these companies have suffered large data breaches leading their users to think they don't focus enough on security, but this app will be as secure as possible and give users a sense of protection.

6. What is the current stage of development / testing of the invention?

This app is still in the very early stages and has not yet been developed.

7. List the names of companies which you think would be interested in using, developing or marketing this invention

Google, Apple, Microsoft

--

8. Funding Partner(s)

Government Agency & Department	
% Support	
Contract/Grant No.	
Contact Name	
Phone No.	
Address	

Industry or another Sponsor	
% Support	
Contract/Grant No.	
Contact Name	
Phone No.	
Address	

9. Where was the research carried out?

I did all of my research through google by checking various tech websites like stack overflow and such and I also made use of NCI's trap to try and come up with an innovative idea

10. What is the potential commercial application of this invention?

A secure structure for your personal files.

11. Was there transfer of any materials/information to or from other institutions regarding this invention?

If so, please give details and provide signed agreements where relevant.

No.

12. Have any third parties any rights to this invention?

If yes, give names and addresses and a brief explanation of involvement.

No.

13. Are there any existing or planned disclosures regarding this invention?

Please give details.

N/A

14. Has any patent application been made? Yes/**No**

If yes, give date: _____ Application No.: _____

Name of patent agent: _____

Please supply copy of specification.

15. Is a model or prototype available? Has the invention been demonstrated practically?

Not yet as this project idea was only approved late Friday evening the 6th of November.

I/we acknowledge that I/we have read, understood and agree with this form and the Institute's *Intellectual Property and Procedures* and that all the information provided in this disclosure is complete and correct.

I/we shall take all reasonable precautions to protect the integrity and confidentiality of the IP in question.

Inventor: **Sean Dolan**

S.Dolan.

08/11/20

Signature

Date