

# Architecture Document Final

Group 25 - Secure Cloud Architecture

	<b>Links</b>
<b>Github</b>	<a href="https://github.com/Plymouth-University/comp2003-2023-25">https://github.com/Plymouth-University/comp2003-2023-25</a>
<b>Monday</b>	<a href="https://riley636485.monday.com/boards/1329741599">https://riley636485.monday.com/boards/1329741599</a>
<b>Trello (Product Backlog)</b>	<a href="https://trello.com/b/VfEbrDfz/ether-storage-product-backlog">https://trello.com/b/VfEbrDfz/ether-storage-product-backlog</a>
<b>Trello (User Stories)</b>	<a href="https://trello.com/b/5DkP9SsO/ether-storage-user-stories">https://trello.com/b/5DkP9SsO/ether-storage-user-stories</a>
<b>Miro</b>	<a href="https://miro.com/app/board/uXjVNtayiRk=/">https://miro.com/app/board/uXjVNtayiRk=/</a>
<b>Testing Form</b>	<a href="https://docs.google.com/forms/d/e/1FAIpQLSeGPbvnH4GMdi7PQaEt1n5YjQlvTgda1vn8LMoYz4CzuUcj0w/viewform">https://docs.google.com/forms/d/e/1FAIpQLSeGPbvnH4GMdi7PQaEt1n5YjQlvTgda1vn8LMoYz4CzuUcj0w/viewform</a>

<b>Group 25 - Ethereal cloud.....</b>	<b>3</b>
Stakeholders and project team (Ben).....	3
Introduction (Ben).....	4
Project Management (Riley).....	6
Communication (Riley, Ben).....	6
Risk plan (Ben).....	7
Development Methodology (Riley).....	11
Product Backlog (Riley).....	12
Sprint Plan (Riley, Ben).....	16
Detailed System Design (William).....	25
Website + Web API (William, Eli).....	25
Storage Controller(William).....	25
Database(William).....	26
Bucket(William).....	27
User Story (Riley, Eli).....	27
ERD - (Riley).....	28
Prototype Snippets (Ben).....	29
HIPO Chart (Riley, Ben).....	31
Sequence Diagrams (Ben).....	32
Flowcharts (Ben).....	38
Quality Assurance (Eli).....	43
Functional Testing (Riley).....	44
Non-Functional Testing (Ben, Eli).....	52
User testing form (Eli).....	59
Conclusion (Will).....	66
References (Ben, Riley, Will, Eli).....	67
Appendix.....	69

# Group 25 - Ethereal cloud

Start up company, supervised by Anthony Edwards

## Stakeholders and project team *(Ben)*

The following is a list of stakeholders for our given project and their role within either the team or the project as a whole, alongside their general responsibilities with the project. These are not limitations on what each stakeholder can do, but just their baseline responsibilities, each person is capable and allowed to go above and beyond their responsibilities should time allow it in order to support the team further and progress our solution faster and more efficiently.

**Anthony Edwards (Tony)** - Client / Startup Advisor Overseer of the project, provided guidance and suggestions for the rest of the team

Team Member	Role	Responsibilities
Benjamin Sanders-Wyatt	Developer / Scrum Master / Full Stack developer	Developer, Scrum Master Website to controller functionality (Web-API), Website frontend developer, Trello, Monday, HTTPS
Eli Bowen	Developer / Integrator Website frontend developer	Developer, Integrator Website frontend developer, Docker, Hosting tests
Riley Coulstock	Team Leader / Product Owner / Maintaining live documents	Team Leader, Product Owner Maintaining live documents (Monday, Trello, Miro), handling meeting minutes, database developer, general research of project components
William Harding	Developer / Architect Backend developer	Developer, Architect Backend developer, architecture document creator, database developer

# Introduction *(Ben)*

Ethereal Cloud is a startup project overseen by Tony which is aimed to provide a secure web-based cloud storage solution.

Our vision for this project was to allow everyone to quickly and efficiently access, store and share files with a user-friendly interface. To achieve this we hosted our cloud storage solution on a web application which gives it the capacity to run on any device.

With the global shift towards online technology and the internet becoming a crucial part of the everyday world for both companies and individuals, as a team we wanted to tap into the growing demand for secure cloud storage solutions. Our goal was to make this product accessible, affordable, and useful for all of our users.

Within this report, we will cover the project management, evaluating our overall approach to planning, risk strategies, and communication methods. We will then justify our system design, referring to diagrams and snippets of code to explain each component of the product in detail. After this, we will carry out testing on both a function and non-functional level, with proof of results which will be evaluated.

Throughout the development Ethereal Cloud has been targeted towards two audiences.

## **General Users:**

Anyone can easily access the Ethereal Cloud web application from their devices, such as computers, tablets and smartphones which a user would only require an internet connection to interact with our product. The general public are the primary audience to build up the Ethereal Cloud brand at which point we can expand to businesses.

## **Businesses:**

To target businesses an on premise system would be sent to the business for them to host themselves. This option would be beneficial over using the general Ethereal Cloud web application as it would allow for enhanced control and security of the files and information shared and stored with the service. Depending on the scale of the business the requirements to support our product would be subject to change.

## **System Requirements**

Based on our preliminary testing of our product we deduced that for a beta pool of around 50 concurrent users the requirements on the host system would be:

	<b>Minimum</b>
CPU	Quad-core 2.1GHz

RAM	8GB RAM
Storage	100Gb (2 Gb per user)
Network Speed	100 mb/s

While the user would only require access to the internet and a device capable of interacting with the web application.

# Project Management (Riley)

## Communication (*Riley, Ben*)

Whilst working on the project we would have regular meetups on Wednesdays which involved catching up on personal work as well as a strategy for the day's group work, all of which was documented by Riley labelled in "Meeting Minutes" within the repository. We organised this during the first month of working on the project together.

Outside of weekly meetings, we communicated with Tony via our group's Microsoft Teams channel every 2 weeks, allowing us to update him whenever significant progress was made, keeping him up to date on project development so that he could ensure we were on track and doing everything we needed to be doing as well as give pointers for when we ran into issues. Outside of meeting up, we communicated via a group chat, this was so we kept each other up to date our progress as well as notify each other of any problems that came up.

We utilised Trello, which contained separate boards for user stories, product backlog and bugs. We used a Kanban methodology and at every meetup on Wednesday, Riley would update the live documents, Trello and Miro, to reflect our progress towards main objectives/components.

To handle our sprints a Monday board was adopted in which Ben and Riley worked together to keep the plan updated with relevant objectives. Each entry contained information on who was in charge of completing it, the timespan allocated to the task, its priority and risk in the project as well as a description of the objective and any relevant information linked towards completing it. This allowed our group to tick off objectives, plan and keep track of the overall project as a whole.

## Risk plan (Ben)

### During Development:

This is the risk plan for development of the project meaning how we would deal with issues or concerns that occur during the development phase.

ID	Risk	Probability (1-10)	Impact (1-10)	Severity Score (%)
1	Loss of Work	2	10	20
2	Data Corruption	1	10	10
3	Exceeding time frame for tasks	5	5	25
4	Bugs that are hard to track down	2	5	10
5	Miscommunication	1	7	7
6	Technical malfunctions	3	10	30
7	Lack of knowledge	7	2	14

ID	Risk Mitigation Strategy
1	We used github for version control and stored our work so that even if we personally lose work it will be stored remotely on git
2	Ensuring that data is processed properly to avoid corruption
3	Allowing for a leeway/overflow for the project. For example the last sprint is solely to account for a backlog of tasks to be completed
4	Searching for the bugs as a group, debugging code using our combined pool of knowledge. Ensuring that working versions are pushed to the git and using good coding practice so it is easy to read
5	Keep in regular contact, and when designating tasks, ensure that all parties involved are fully aware of what is required of them
6	This is an external factor that could affect our work, however we can utilise university resources in the event that a group member faces this
7	When facing a task that a group member doesn't know how to tackle, they will consult the group to decide if another member is more suited to the task - if not, the group will dedicate time from the sprint in order to learn it

## Final Product:

This is the risk plan for the final project meaning how we would deal with issues or concerns that occur during or after the deployment of the project.

ID	Risk	Probability (1-10)	Impact (1-10)	Severity Score (%)
1	Broken access control: Unauthorised access to data	1	10	10
2	Server Downtime: System isn't running affecting service availability.	2	7	14
3	Api Vulnerability: Api exploits.	1	8	8
4	SQL Injections: Injection of malicious SQL queries leading to unauthorised access of data	1	10	10
5	Data Loss: Removal or corruption of data.	1	7	7
6	Compliance: Lack of compliance to data protection laws	1	10	10
7	Resource Scaling: Failure to scale resources based on the needs.	2	6	12

ID	Risk Mitigation Strategy
1	Thorough and continuous testing of all available data to users to ensure only those authorised users can access protected resources, ensure the access control is tested by multiple developers.
2	Redundant file storage, easy to relaunch project incase of downtime, minimise crashes through appropriate error handling
3	All API endpoints are thoroughly tested with normal, erroneous and boundary data to ensure it only accepts the correct data and can't be exploited into allowing access without proper authorization.
4	Through the use of prepared statements and the like to stop SQL queries being interpreted literally using user input.

5	By having redundant versions of files in different buckets to ensure that if a bucket is corrupted, the file can be replaced through the redundant file.
6	By ensuring that the data is held as securely as possible and can only be accessed by the users themselves and no one else, also design the software with GDPR / Data Protection act in mind as to ensure compliance.
7	By designing the software to be scalable, i.e. Using docker, allows the software to be customised to work with different sets of data so that multiple containers of the same image can be launched on different machines, allowing load balancing, ensuring scalability.

### Mitigation Plan

**During development:** Before the development of our project our team created plans to mitigate the effect of any unforeseen issues.

Unforeseen Issue	Contingency Plan
Member illness	Discussion as to how to account for the ill members responsibilities, ie. splitting up their responsibilities between the other team members.
Anything under EC's	Working as a team to finish the work and supporting each other as much as possible to ensure the work gets completed in a timely manner.
Technical issues	Other team members account for anyone experiencing technical issues, Eg. a broken laptop, until the team member can get their technical issue fixed.

**After development:** These are the plans our team came up with to mitigate any issues that might occur while the final product is running

Unforeseen Issue	Contingency Plan
Maintenance	Ensure clients know in advance that there is planned maintenance, so they have enough time to get any files off the cloud if they need.
Unscheduled web application downtime	Have user feedback to show the clients the issue is being solved or worked on.
DDOS attacks	Use stronger firewalls that rely on strict policy measures, this way any suspicious packages will be rejected [5].
SQL Injection	We have data sanitisation which involves the secure and permanent erasure of sensitive data. This will guarantee that no residual data can be recovered.

## **Legal (Ben)**

Legally, Ethereal Cloud has many responsibilities, one such responsibility being both legally and ethically, to protect our user's data, and to follow the General Data Protection Regulation (GDPR, 2018), which dictates how we can collect and use user data. This regulation states that collecting user data must be limited to data relevant to the purpose it is being collected for, the minimum amount of data required should be collected, and that data must only be held for the period of time of which it is relevant, to name a few. Due to the principles stated in the legislation, we have made sure that data is only being used for its intended purpose and that only necessary data is collected.

## **Social (Ben)**

As for social implications of this solution, they are minimal as there are few vectors of socialisation through our product, the biggest concern would be with sharing a file with someone. Considering the file is only between the parties it's shared with there is no issue concerning problems such as hate speech. Taking this into account it would be difficult for malicious actors to use Ethereal Cloud as a platform, however our platform can be used to target specific individuals still without much difficulty.

## **Ethical (Ben)**

In addition to our legal and social responsibilities we have maintained a commitment to upholding ethical practices continuously through this project. The protection of users data is essential and our team has implemented many security features to ensure this, including, two factor authentication, JWT tokens, session timeouts, HTTPS, SSL and password hashing and salting [10].

To foster trust with our users we have ensured that their data is only used for the purpose they have provided the data for; to ensure that users are given a terms of service and privacy policy before registration which provides them with full transparency about how their data will be used. Within the terms of service we have also included our right to remove files and data which violate our ethical or legal standards.

During our design phase we endeavoured to make our interface as inclusive and accessible as possible to allow for any user no matter their needs to use our cloud service. By keeping all of this in mind our team has created a product we believe to be ethical, inclusive and trustworthy.

## **Development Methodology (Riley)**

We wanted to focus on creating working prototypes, so decided on the scrum methodology as described by Schwaber, K., 1997 [1]. We started by defining our scope, and outlined the project and its features. Once this was done, we then used Trello (as specified in the Communication section) to note down features and ideas we wished to implement, before coming to a decision and starting an initial design for the final product.

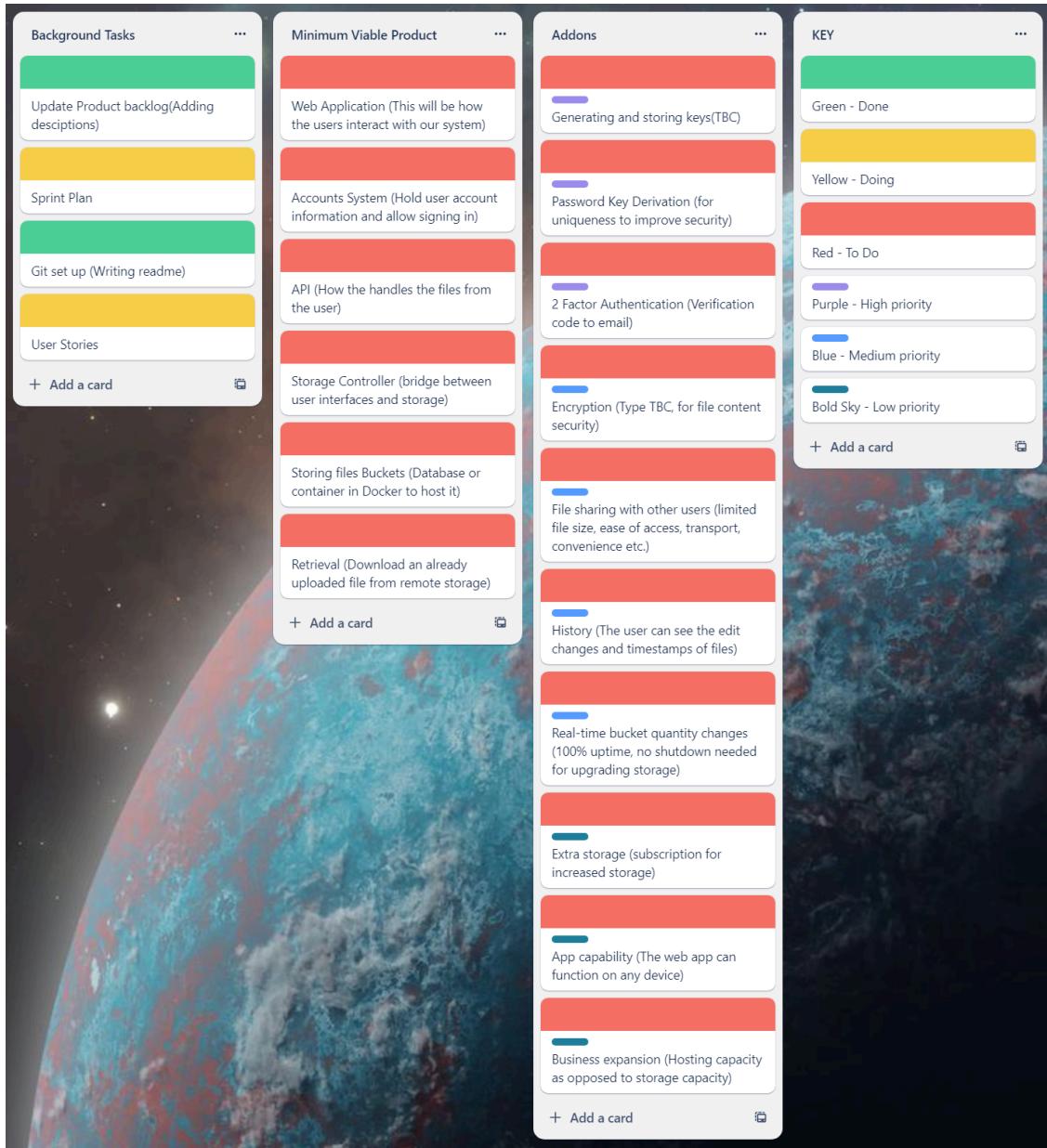
At this point, we then agreed on sprint lengths and with that the breakdown of tasks that needed to be performed at each sprint (see Sprint Plan below). We kept regular contact to keep track of progress in the sprint, with weekly meetings (mentioned above) to mark definitive progress as well as tackle tasks as a group (in person). Then at the end of the sprint, which was held every other week, we discussed our progress and design choices with our client/supervisor and asked for guidance when needed. This process was repeated over both semesters, providing steady and efficient progress of our project.

The use of scrum methodology proved extremely effective given our group was a start-up, which meant that there needed to be lots of room for error, alongside needing flexibility on requirements and needs; scrum was best suited for our needs.

## Product Backlog (Riley)

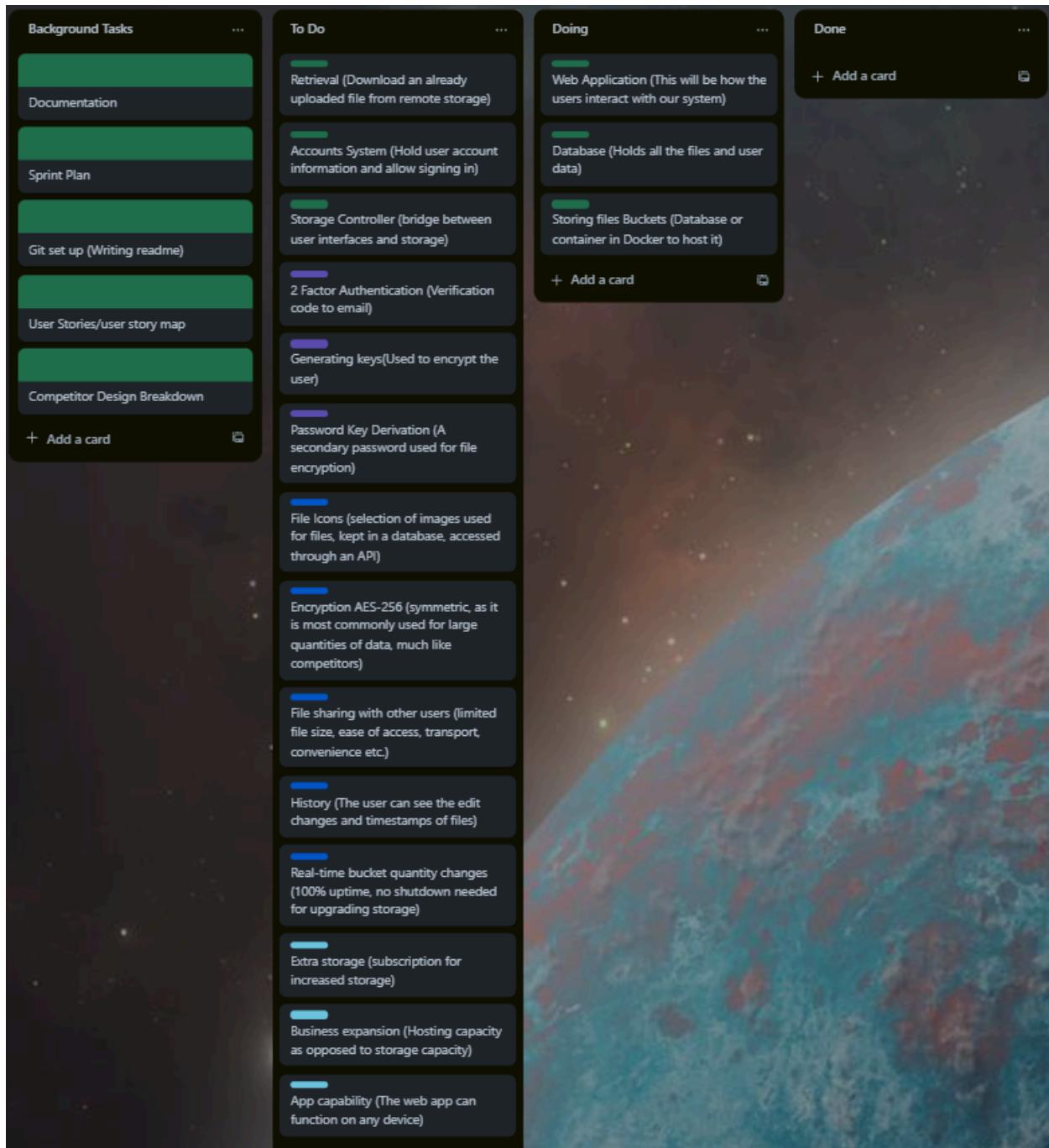
At the start of the project, we outlined the main features and ideas that we wanted to include in the system.

01-11-2023



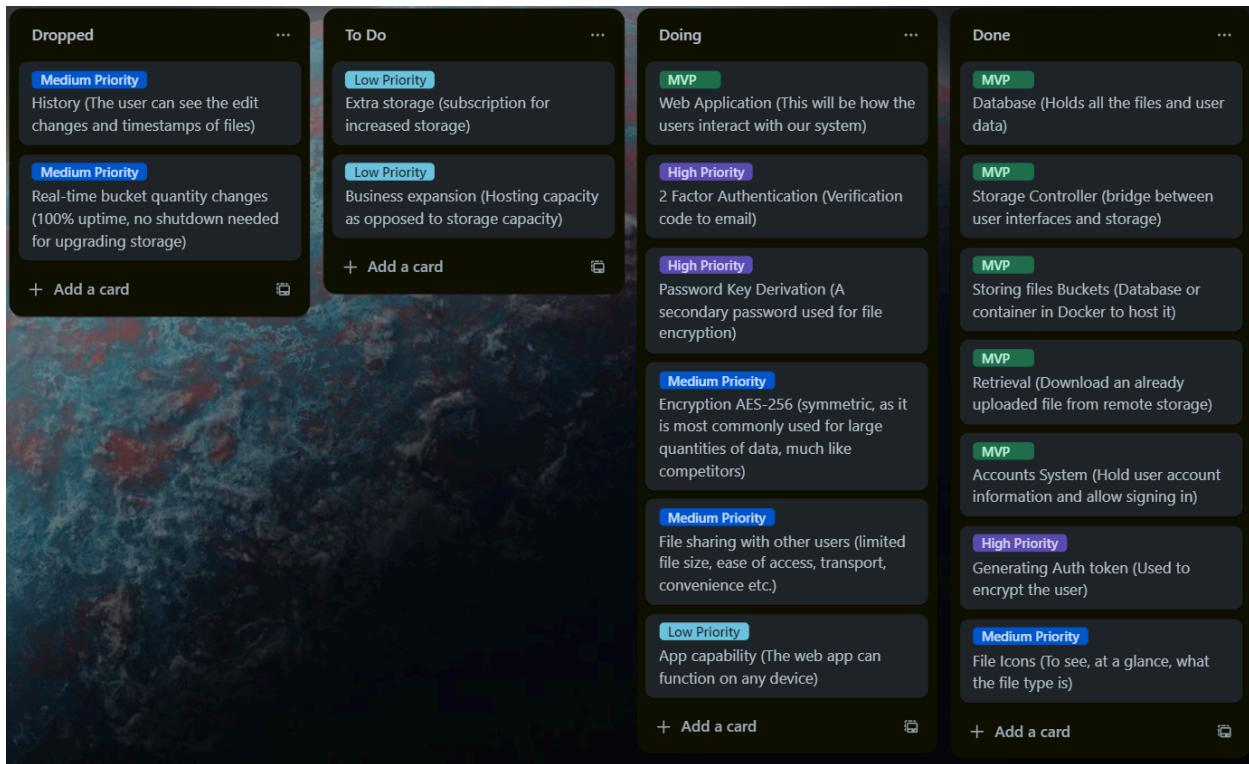
Our first priority was the Minimum Viable Product (MVP), which at its core, was to allow the user to log in, upload a file, and then be able to download the file.

17-01-2024



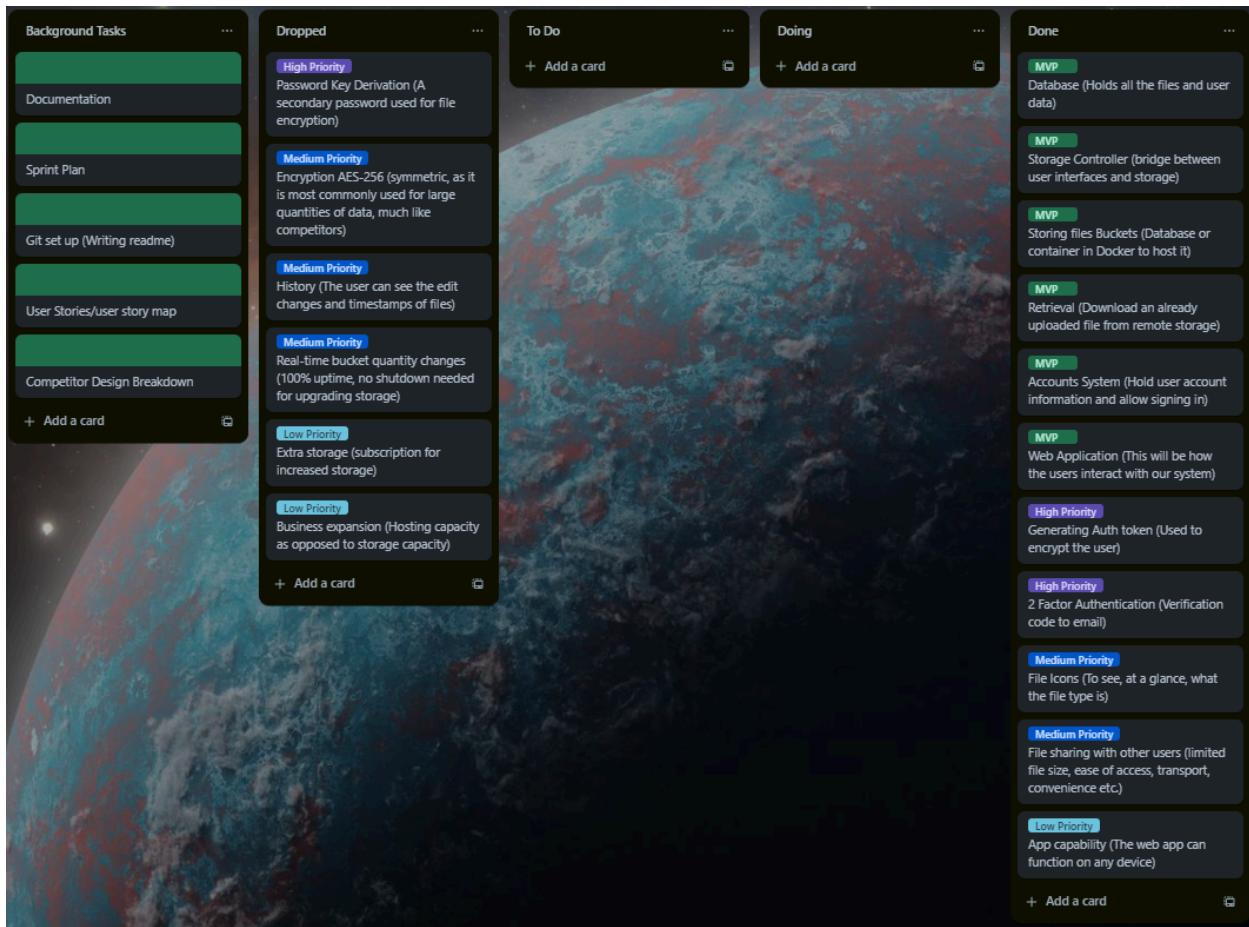
Here is the 1st version of the new product backlog format. After a discussion with Tony, we decided to incorporate a kanban methodology, where the green signifies the MVP, the purple signifies high priority (features that we really want in our end product), the dark blue (features that are also good, but not first choice in development of features), and light blue (signifies features that are nice but most likely wouldn't make it to the final product). With this, we had a more simple layout that tracked our progress as well as assigned clear importance as to what tasks we prioritise above others.

29-03-2024



It was at this point in the project that we had to introduce a “Dropped” section. We were nearing the end of our project and with how development was going, there were certain features that were either out of reach given the combination of our depth of knowledge and time left (Real-time bucket quantity changes), or simply didn’t fit right with the project (History). This gave us insight into the necessity of ensuring that our project plan and what we were developing lined up.

17-04-2024



Here is the final update for the product backlog, which has noticeable differences compared to the previously mentioned timestamp. Our project was officially completed in terms of minimum required functionality of each component. However, we had to drop more features (we didn't officially drop them until we had notified our client at a later date, but as of the timestamp these features weren't going to be completed) as they were either unrealistic, or they would not be implemented to the standard we wanted to, as it was either not have them, or have an unfunctional, problematic component in our system (which obviously wasn't an option this late in the project).

After this point in time, we performed multiple iterations of whitebox and blackbox testing in order to locate as many bugs as possible which were then debugged by the team member most suitable for the bug which was mainly the person who coded the specific section.

## Sprint Plan (Riley, Ben)

The sprint plan was used for keeping a record of what everyone in the group was doing and setting tasks for each sprint (every two weeks). It contained all of the actions, processes and components that needed to be carried out/assembled over the course of each sprint.

We chose to do our sprint plan using a Monday board, as its software was specifically suited for our needs; It allowed us to create “groups” which we termed our sprints. While the Trello board was used for broad goals the Monday board would go into detail about the specifics of the tasks.

Here is an example of how the monday board was laid out.

Sprint 1							
	Task	Owner	Status	Timeline	Priority	Risk	Long Text
	Make Documents 4	BS	Done	Oct 25, '23 - ..			
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text
	Progressing Architecture Documents - Co...	WH	Done	Oct 25, '23 - ..	High	Low	The architecture documents will show the planned archite...
	Create Product Backlog	RC BS	Done	Oct 25, '23 - ..	Medium	Low	The product backlog is a prioritized list of work for the de...
	Create user Stories	RC BS	Done	Oct 25, '23 - ..	Low	Low	Outlines the users needs and wants for the system.
	Create sprint plan	BS RC	Done	Oct 25, '23 - ..	High	Low	This monday board is the sprint plan.
	+ Add subitem						
	Initialise Github	EB	Done	Oct 25, '23 - ..	High	Low	Setting up how the github will be structured a...
	+ Add task						

Each linked task within a sprint was grouped together in a parent task and at the end of the sprint the average priority and risk is shown. For the individual tasks they contained the following information:

- **Task** - The task is the title of the objective which gives a broad understanding of the sprint.
- **Owner** - The owner is assigned to one or more of the members of our group, indicating that they will be in charge of completing the objective. BS = Ben Sanders-Wyatt, EB = Eli Bowen, RC = Riley Coulstock and WH = William Harding.
- **Status** - The status is an easy way for our team to see what objectives are being worked on, what needs doing and what has been completed.
- **Timeline** - The timeline is the time allocated to this task. This couldn't always be strictly adhered to in the case that an issue occurred but to mitigate this we ensured that these problems wouldn't affect other tasks and in turn the rest of the timeline.
- **Priority** - The priority of a task is set to indicate how impactful to the overall project the objective is, the options being Low, Medium, High and Critical. Each member of the team could then take this information and plan their assigned tasks accordingly.
- **Risk** - The risk similarly to priority has the allocations of Low, Medium, High and Critical. These indicate the likelihood and impact of issues occurring during the task and helped our team to easily plan around potential issues.

- **Long Text** - The long text column was used to describe the goal of the sprint and what was required to complete it.

Here is the final sprint plan for the project.

## Sprint 1

### ▼ Sprint 1

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	Make Documents 4		Done	Oct 25, '23 - ...				
<input type="checkbox"/>								
<input type="checkbox"/>	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
<input type="checkbox"/>	Progressing Architecture Documents - Co...		Done	Oct 25, '23 - ...	High	Low	The architecture documents will show the planned archite...	
<input type="checkbox"/>	Create Product Backlog		Done	Oct 25, '23 - ...	Medium	Low	The product backlog is a prioritized list of work for the de...	
<input type="checkbox"/>	Create user Stories		Done	Oct 25, '23 - ...	Low	Low	Outlines the users needs and wants for the system.	
<input type="checkbox"/>	Create sprint plan		Done	Oct 25, '23 - ...	High	Low	This monday board is the sprint plan.	
<input type="checkbox"/>	+ Add subitem							
<input type="checkbox"/>	Initialise Github		EB	Done	Oct 25, '23 - ...	High	Low	Setting up how the github will be structured a...
<input type="checkbox"/>	+ Add task							
<div style="display: flex; justify-content: space-around; align-items: center;"> <span>Oct 25, '23 - Nov 3, '23</span> <span></span> <span></span> <span></span> <span></span> </div>								

## Sprint 2

### ▼ Sprint 2

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	Website Design 3		Done	Nov 4, '23 - N...				
<input type="checkbox"/>								
<input type="checkbox"/>	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
<input type="checkbox"/>	Gather Competitor Research		Done	-	Low	Low	Gathering information and screenshots from other already...	
<input type="checkbox"/>	Create UI prototype		Done	-	Low	Medium	Using the competitor research create a prototype design ...	
<input type="checkbox"/>	Research Bootstrap		Done	-	Low	Low	Gain understanding to how to use the tool.	
<input type="checkbox"/>	+ Add subitem							
<input type="checkbox"/>	Research API		BS +3	Done	Nov 4, '23 - N...	High	Low	Gain understanding of how to create & operat...
<input type="checkbox"/>	Research Encryption		RC +3	Done	Nov 4, '23 - N...	High	Low	Gain understanding of how to encrypt and han...
<input type="checkbox"/>	Create user story map		EB WH	Done	Nov 4, '23 - N...	Medium	Low	Create a further breakdown of the user stories.
<input type="checkbox"/>	+ Add task							
<div style="display: flex; justify-content: space-around; align-items: center;"> <span>Nov 4, '23 - Nov 17, '23</span> <span></span> <span></span> <span></span> <span></span> </div>								

## Sprint 3

### Sprint 3

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	Research 2		Done	Nov 18, '23 - ...	Medium	Low		
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
<input type="checkbox"/>	Research API		BS +3	Done	Nov 18, '23 - ...	Medium	Low	Gain understanding of how to create & create API's.
<input type="checkbox"/>	Research Encryption		RC +3	Done	Nov 18, '23 - ...	Medium	Low	Gain understanding of how to encrypt and handle data.
	+ Add subitem							
<input type="checkbox"/>	Database Design & Development 5			Done	Nov 18, '23 - ...			
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
<input type="checkbox"/>	Specify database structure		WH	Done	-	High	Medium	
<input type="checkbox"/>	Create ERD		RU WH	Done	-	Medium	Low	The Entity Relationship Diagram will show how the tables ...
<input type="checkbox"/>	Normalisation of Entities/Attributes		RC	Done	-	Medium	Low	Outlining the general structure of the database.
<input type="checkbox"/>	Create Field Definition Grid		RC	Done	-	Low	Low	Highlights the attributes within the tables. Showing data...
<input type="checkbox"/>	Create Database		RC	Done	-	High	High	This is where the database SQL will be created to implem...
	+ Add subitem							
<input type="checkbox"/>	Set up Docker		EB WH	Done	Nov 18, '23 - ...	Critical	High	Creating and running docker containers which...
<input type="checkbox"/>	Website Development 2			Done	Nov 18, '23 - ...			
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
<input type="checkbox"/>	Highlight HCI principles		BS	Done	Nov 18, '23 - ...	Low	Low	(Human computer interactions) This will show how our de...
<input type="checkbox"/>	Implement Design		BS	Done	Nov 18, '23 - ...	Medium	Low	This is where the design will be implemented onto the we...
	+ Add subitem							
<input type="checkbox"/>	+ Add task							
				Nov 18, '23 - Dec 1, '23				

## Sprint 4

### Sprint 4

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
	Docker 2	EB	Done	Dec 2, '23 - Dec 15, '23				
	Subitem						Long Text	
	Setup Docker Containers	EB WH	Done	Dec 2, '23 - Dec 15, '23	Critical	High	Creating and running docker containers which will hold a...	
	Setup Docker Vm	EB	Done	Dec 2, '23 - Dec 15, '23	High	Medium	The docker Virtual Machine will be a security solution.	
	+ Add subitem							
	Database 2	EB	Done	Dec 2, '23 - Dec 15, '23				
	Subitem						Long Text	
	Implement Database into Docker	RC	Done	Dec 2, '23 - Dec 15, '23	Critical	Medium	This is where the database SQL will be created to implem...	
	Enable Communication between storage co...	WB BS	Done	Dec 2, '23 - Dec 15, '23	Critical	Medium	This is so that the website can access the API's and the d...	
	+ Add subitem							
	Website 2	EB	Done	Dec 2, '23 - Dec 15, '23				
	Subitem						Long Text	
	Add website pages	BS EB	Done	Dec 2, '23 - Dec 15, '23	High	Low	The website pages will be added, e.g. home page, account...	
	Finalize website design	BS	Done	Dec 2, '23 - Dec 15, '23	Medium	Low	Create a final prototype which will be used for the website.	
	+ Add subitem							
	Web API's 5	EB	Done	Dec 2, '23 - Dec 15, '23				
	Subitem						Long Text	
	Plan/Design Storage Controller	WH	Done	Dec 2, '23 - Dec 15, '23	High	Low	The storage controller will directly communicate with bot...	
	Create Storage Controller	WH	Done	Dec 2, '23 - Dec 15, '23	Medium	Medium	Create the planned storage controller.	
	Plan/Design File Explorer	BS	Done	Dec 2, '23 - Dec 15, '23	Low	Low	The file explorer will gather all the user files and show the...	
	Create Validation Requirements	RC	Done	Dec 2, '23 - Dec 15, '23	Low	Low	The validation requirements can be gathered from the field...	
	Create Accounts API	BS EB	Done	Dec 2, '23 - Dec 15, '23	Medium	Medium	Create the API which will handle the user accounts.	
	+ Add subitem							
	+ Add task							
				Dec 2, '23 - Dec 15, '23				

## Sprint 5

### Sprint 5

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
	Implement website design	BS EB	Done	Dec 16, '23 - Dec 29, '23	Low	Low	Fully implement our design onto the website.	
	Implement Storage Controller	WH	Done	Dec 16, '23 - Dec 29, '23	Medium	Medium	Implement the storage controller onto the website.	
	Implement Folder Creation	BS	Done	Dec 16, '23 - Dec 29, '23	Low	Medium	Implement the storage controller onto the website.	
	+ Add task							
				Dec 16, '23 - Dec 29, '23				

## Sprint 6

Sprint 6 was used as a time buffer where we could look over the project as a whole. This allowed us to fix any bugs and ensure that the project continued to run smoothly.

### Sprint 6 (Time Buffer)

	Task	Owner	Status	Timeline	Priority	Risk	Long Text
	Buffer	⊕	Done	Dec 30, '23 - ...			Bug fixes if necessary
	+ Add task						

## Sprint 7

### Sprint 7

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
	>Create File Explorer 4	⊕	Done	13 - 26 Jan			Create the panned file explorer.	
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	-
	Create file paths	⊕ BS	Done	13 - 26 Jan	Critical ⚡	High	Create the ability to go into and out of folders	
	Create Folder naming function	⊕ BS	Done	13 - 26 Jan	Low	Low	Create function to allow the naming of folders	
	Implement Folder naming	⊕ BS	Done	13 - 26 Jan	Critical ⚡	High	Implement the function into the project to work with the ..	
	Implement subfolders	⊕ BS	Done	13 - 26 Jan	High	Medium	Implement the ability to create folders inside of other fol...	
	+ Add subitem							
	>Password Hashing 5	⊕	Done	13 - 26 Jan			Hashing the passwords for secure storage	
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	-
	Research suitable methods	⊕ WH	Done	13 - 26 Jan	Low	Low	Gain knowledge about industry standard methods of pass...	
	Incorporate salt on password	⊕ WH	Done	13 - 26 Jan	Medium	Low	Utilizing salting of password to ensure security	
	Implemented hash	⊕ WH	Done	13 - 26 Jan	Medium	Low	Hashing of the password used so that the password isn't s...	
	Reworked database	⊕ WH	Done	13 - 26 Jan	Medium	Low	Including salt storage into database	
	verifying hash on login	⊕ WH	Done	13 - 26 Jan	Medium	Medium	Password is now authenticated by comparing	
	+ Add subitem							
	Perform white box testing	⊕ RC	Done	13 - 26 Jan	High	Medium	Ensuring the flow of data through the system ...	
	Attend to bugs 2	⊕	Done	13 - 26 Jan				
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	-
	Authtoken Issues	⊕ BS WH	Done	13 - 26 Jan	Medium	Medium	Authtoken error for each user.	
	File leak issues	⊕ WI BS	Done	13 - 26 Jan	Critical ⚡	Medium	Fixed issue where all users could see all files.	
	+ Add subitem							
	Add folder table	⊕	Done	13 - 26 Jan			Separate folders and files into separate tables..	
	+ Add task							

## Sprint 8

### Sprint 8

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
	Implement file encoding	⊕ WH	Done	27 Jan - 9 Feb	Medium	Low	File encoding is used to ensure that the file int...	
	Rework File upload process	⊕ BS WH	Done	27 Jan - 9 Feb	Low	Low	Changed encoding from base64 to hex	
	Implement error pop up message	⊕ EB BS	Done	27 Jan - 9 Feb	Low	Low	Letting the user know when something has do...	
	Bootstrap 2	⊕	Done	27 Jan - 9 Feb				
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
	Migrate log in page	⊕ EB	Done	27 Jan - 9 Feb	Low	Low	transferred log in page from html to bootstrap	
	Migrate sign up page	⊕ EB	Done	27 Jan - 9 Feb	Low	Low	transferred sign up page from html to bootstrap	
	+ Add subitem							
	+ Add task							
				27 Jan - 9 Feb				

## Sprint 9

### Sprint 9

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
	Filepath 3	⊕	Done	10 - 23 Feb				
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
	Develop display	⊕ EB	Done	10 - 23 Feb	Low	Low	The filepath location on the display as well as how the use...	
	Incorporate images	⊕ BS	Done	10 - 23 Feb	Low	Low	Vector images were added for home and back button	
	Develop layout	⊕ EB BS	Done	10 - 23 Feb	Low	Low	Improved design and functionality of customer side pages	
	+ Add subitem							
	Improve filepath display	⊕ BS	Done	10 - 23 Feb	Low	Low	Improved visuals for how the file path is shown	
	Storage controller 4	⊕	Done	10 - 23 Feb				
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
	Optimised save endpoints	⊕ WH	Done	10 - 23 Feb	Low	Low	Broke down the file endpoint to 1 save endpoint	
	optimised get endpoints	⊕ WH	Done	10 - 23 Feb	Low	Low	Broke down the file endpoint to 1 get endpoint	
	Generalise authorisation	⊕ WH	Done	10 - 23 Feb	Low	Low	Instead of copy and pasting user authentication from end...	
	Create helper class for api request	⊕ WH	Done	10 - 23 Feb	Low	Low	Class created to help communicate between API's	
	+ Add subitem							
	Ensure bucket and database sync	⊕ WH	Done	10 - 23 Feb	High	Medium	Make sure that the bucket and database com...	
	Compile bug list	⊕ RC	Done	10 - 23 Feb	Medium	High	Take note of any bugs found in order to resolv...	
	+ Add task							
				10 - 23 Feb				

## Sprint 10

### Sprint 10

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
	Add Models 2	BS	Done	24 Feb - 8 Mar				
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
	Create an optimised input forms	BS	Done	24 Feb - 8 Mar	Low	Low	User inputs are able to be validated by the models and ea...	
	Integrate file path to model	BS	Done	24 Feb - 8 Mar	Low	Low	The file path is changed to work with the new models	
	+ Add subitem							
	Revise model constraints	RC	Done	24 Feb - 8 Mar	Low	Low	The constraints are used to ensure that the us...	
	Research terms and conditions	EB	Done	24 Feb - 8 Mar	Low	Low	Gain knowledge of general terms and condition...	
	Research privacy policy	EB	Done	24 Feb - 8 Mar	Low	Low	Gain knowledge of general privacy policies for ...	
	Research improvements for api communication	WH	Done	24 Feb - 8 Mar	Low	Low	Understand how the api communication can b...	
	+ Add task							

## Sprint 11

### Sprint 11

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
	Revise database to match constraints	RC	Done	9 - 22 Mar	Medium	Low	Making sure that the limits and data types for ...	
	Model Constraints 2	RC	Done	9 - 22 Mar				
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
	Create password validation	RC	Done	9 - 22 Mar	Medium	Low	Create function for the validation of passwords on user si...	
	Implement password validation	RC	Done	9 - 22 Mar	Medium	Low	Implement function into register page	
	+ Add subitem							
	Create 2 Factor authentication	RC EB	Done	9 - 22 Mar	Low	Low	Create function(s) in order to increase securit...	
	Implement 2FA	RC EB	Done	9 - 22 Mar	Medium	Medium	Implement the function into the necessary pa...	
	Implement HTTPS	BS	Done	9 - 22 Mar	Critical	Medium	Setup the website with a self signed certificat...	
	+ Add task							

## Sprint 12

### Sprint 12

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	Test Project 2		Done	23 Mar - 5 Apr				
<input type="checkbox"/>	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	User Testing		Done	23 Mar - 5 Apr	High	Medium	Give our project to testers to get feedback and report on ...	
<input type="checkbox"/>	Response to testing		Done	23 Mar - 5 Apr	High	Medium	Given the results of the tests we will make the necessary...	
<input type="checkbox"/>	+ Add subitem							
<input type="checkbox"/>	Auth Page 4		Done	23 Mar - 5 Apr				
<input type="checkbox"/>	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	Create Auth page UI		Done	23 Mar - 5 Apr	Low	Low	Create the looks of how the user would interact with the ...	
<input type="checkbox"/>	Implement Email code generation		Done	23 Mar - 5 Apr	Medium	Low	For two factor authentication a 6 digit code is emailed to ...	
<input type="checkbox"/>	Implement auth code check		Done	23 Mar - 5 Apr	Medium	Low	Implement the logic to ensure that the user has entered t...	
<input type="checkbox"/>	Create Testing Emails		Done	23 Mar - 5 Apr	Low	Low	Create two emails. One is to send the 2 factor authentica...	
<input type="checkbox"/>	+ Add subitem							
<input type="checkbox"/>	Implement restore from bin		Done	23 Mar - 5 Apr	High	Medium		
<input type="checkbox"/>	Upload page 2		Done	23 Mar - 5 Apr				
<input type="checkbox"/>	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	Implement renaming		Done	23 Mar - 5 Apr	High	Medium	Implement the renaming function so that folders and files...	
<input type="checkbox"/>	Implement sorting		Done	23 Mar - 5 Apr	Low	Low	Make it so that the user can press a button to sort the dis...	
<input type="checkbox"/>	+ Add subitem							
<input type="checkbox"/>	Enable compatibility with browsers		Done	23 Mar - 5 Apr	High	High	All browsers need to be able to display the we...	
<input type="checkbox"/>	+ Add task							

## Sprint 13

### Sprint 13

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	Share Page 5		Done	6 - 19 Apr				
<input type="checkbox"/>	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	+
<input type="checkbox"/>	Create page looks		Done	6 - 19 Apr	Low	Low	Create how the page looks. Doing the CSS.	
<input type="checkbox"/>	Create popups		Done	6 - 19 Apr	Low	Low	Create the popups to input data such as the username to ...	
<input type="checkbox"/>	Implement popups		Done	6 - 19 Apr	Low	Low	Implement the functionality behind the popups.	
<input type="checkbox"/>	Create Share endpoints		Done	6 - 19 Apr	High	Medium	Create the endpoints that can be accessed by the webap...	
<input type="checkbox"/>	Implement endpoints on webapi		Done	6 - 19 Apr	High	Medium	Implement on the webapi when the endpoints will be call...	
<input type="checkbox"/>	+ Add subitem							
<input type="checkbox"/>	Complete documentation		Done	6 - 19 Apr	Medium	Low	Complete the required documentation for the...	
<input type="checkbox"/>	+ Add task							

## Sprint 14

### Sprint 14 (Time Buffer)

	Task	Owner	Status	Timeline	Priority	Risk	Long Text	+
	Test Project 2	⊕	Done	20 Apr - 2 May				
	Subitem	Owner	Status	Timeline	Priority	Risk	Long Text	
	User Testing	⊕ BS RC	Done	20 Apr - 2 May	High	Low	Give project to users to get feedback and discover bugs.	
	Response to testing	⊕ BS +3	Done	20 Apr - 2 May	High	Low	Respond to feedback from the testing.	
	+ Add subitem							
	Complete documentation	⊕ BS +3	Done	20 Apr - 2 May	High	Low	Complete the required documentation for the ...	
	+ Add task							

# Detailed System Design *(William)*

For this product, we came up with a set amount of design layers that would enable us to expand the solution to more dedicated platforms down the line and that would allow hosting the product to be scaled up. The dedicated layers of the solution are as follows:

- Web site using an ASP.NET Web API and Razor pages.
- Storage controller ASP.NET API for interacting with the database and buckets.
- The database for storing the user and file data.
- Bucket ASP.NET API for storing the file contents.

## Website + Web API *(William, Eli)*

The first layer of the solution is the website and web API, which are baked into the same layer as they are using Razor pages, a framework with ASP.NET that allows C# to be mixed directly into HTML files and automates a lot of the gap bridging between the HTML / Javascript and the C# API, allowing us to more effectively develop pages without having to manually manage sending requests from the javascript to the API.

We have also included HTTPS in our project. The certificate is attached to the docker image but for it to work the certificate must be installed on the local computer or server. With this certificate it then verifies that the data is secure, this means any files sent to or from the website are encrypted in transit. Also, when entering data into the login or register forms this will be encrypted when sent to and from the database.

We are also using bootstrap for the front end of the website, making it easier for the front end development to apply to a variety of screen sizes and to make styling the website in general easier and faster due to bootstrap's built in styles for different types of areas of websites. Bootstrap also allows us to cater for all types of web browsers as it has built in css with the styles you can add to the HTML.

## Storage Controller *(William)*

Leading on from the web API, the storage controller is another ASP.NET API that is the bridge between the platform specific APIs and the database and file storage buckets, this application has no website and only answers requests to specific endpoints in order to add, remove, edit or view rows in the database pertaining to the user's data, file data, folder data and so on. The storage controller also decides where files will be saved (in which bucket), and would gain the ability to decide where redundant file stores would be held when saving new files. The storage controller also utilises entity framework core that accelerates the querying of the database and makes creating and managing the database entities a lot easier while developing endpoints.

We decided to utilise this storage controller as it allows us to develop extra methods of accessing the data, such as android or IOS apps, by accessing this central API for the database

which also handles authentication and authorization tokens for requests. By having this central point for data access, we also ensure that if any exploits are found for data access, we only have 1 point of failure to fix, ensuring that we don't end up having to fix multiple exploits across multiple platforms.

### Database(*William*)

The database is handled purely by the storage controller, and is where all of the data regarding the users, files and folders are stored, the database is an MSSQL database from microsoft and is run in docker alongside all the other components of the system. The database ended up with 8 used tables, which are detailed in the table below.

Table	Columns
Users	UserID Username Email Password (hashed) PasswordSalt Administrator
UserFolders	FolderID UserID Privilege
UserFiles	FileID OwnerID UserID Privilege
Folders	FolderID FolderName ParentID
FolderBin	DeleteID FolderID
Files	FileID FileType FileName FilePassword CreationDate RedundantID BucketID FolderID
FileBin	DeleteID FileID

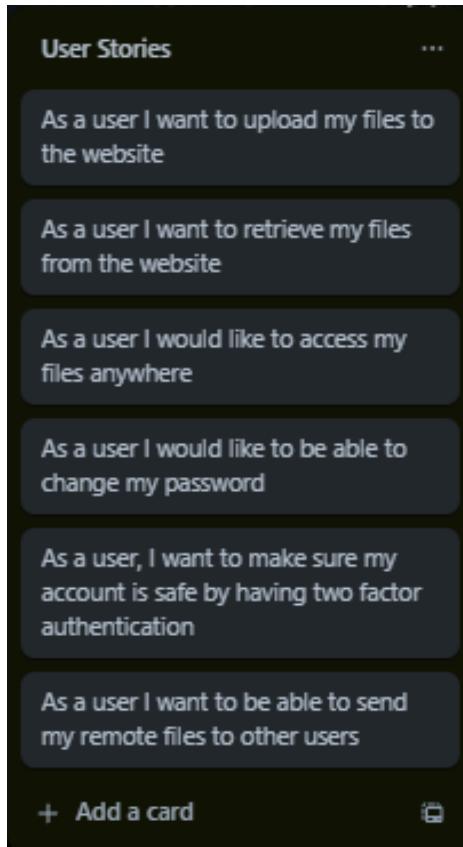
Buckets	BucketID BucketIP BucketPort
---------	------------------------------------

## Bucket (*William*)

The bucket's only purpose is to store the content of files, it also connects to a database, creates a table of the files it's storing and their file path in its local file system and then stores the content under a random file name. This works using a docker volume to persist the bucket contents through restarts of the container and answers requests using an ASP.NET API with purpose built endpoints for saving, deleting and retrieving a file's contents [9].

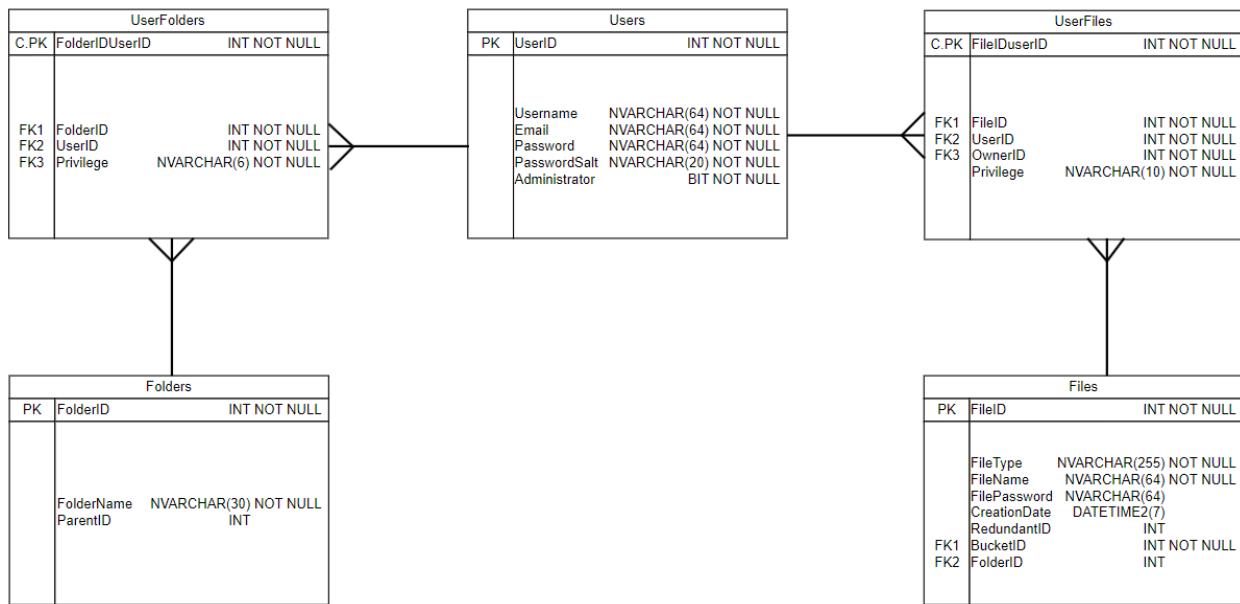
By handling file contents this way, we can scale the storage capacity of the website by adding more buckets into the mix, allowing the storage controller to select from more areas to save files, also enabling data redundancy in case the main file content save is inaccessible.

## User Story (*Riley, Eli*)

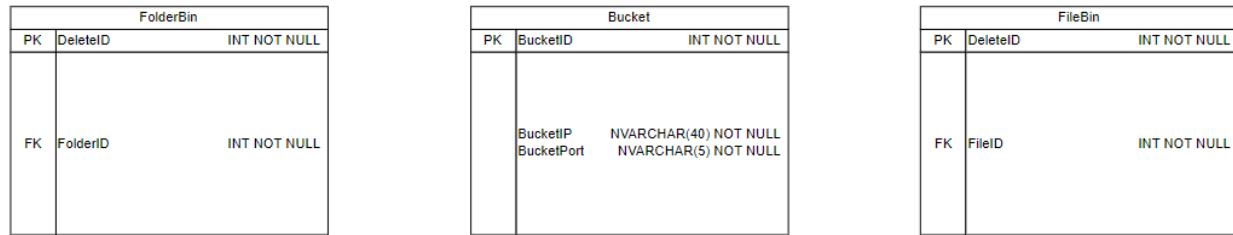


Here we simply laid out everything the user would want on our cloud storage website. This included things such as being able to upload files, changing their password, and being able to access their files from anywhere.

## ERD - (Riley)

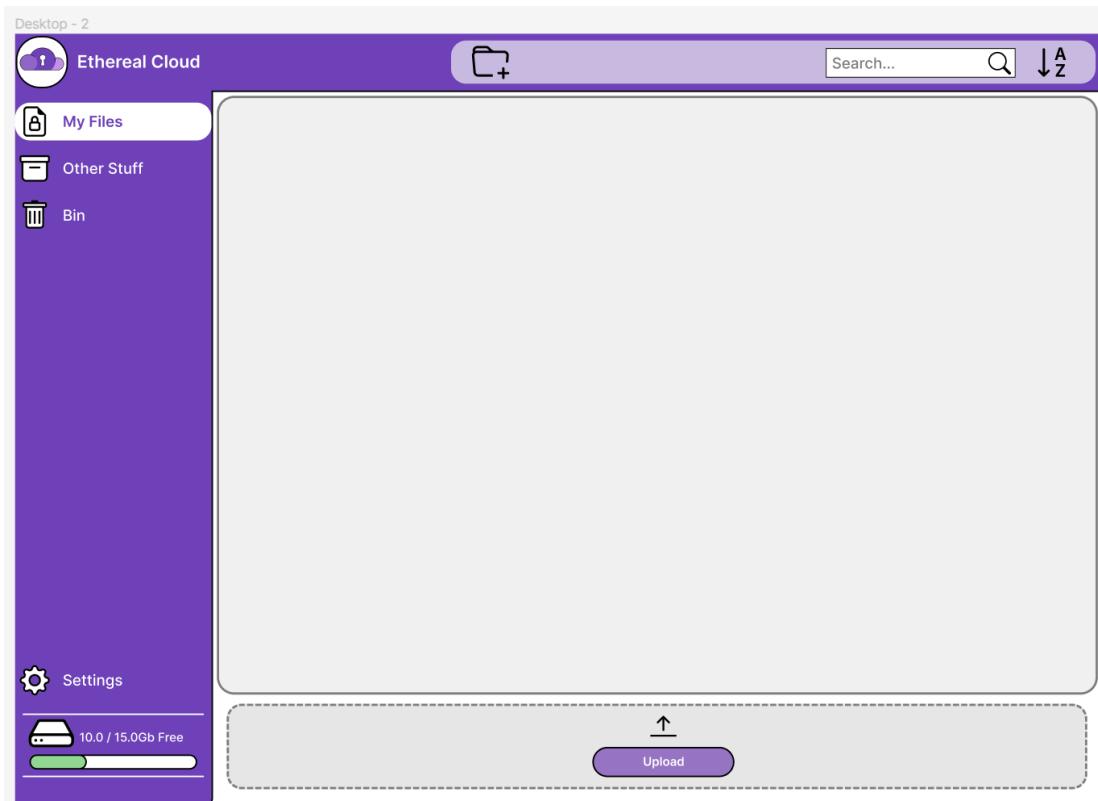


Here is our ERD diagram showing the relationships of our project and how each works together.



These are entities that don't have a many to one relationship with other entities but they still serve a purpose. The Bins are responsible for holding the folder or file ID of deleted folders or files. The Bucket's role is the same as mentioned above.

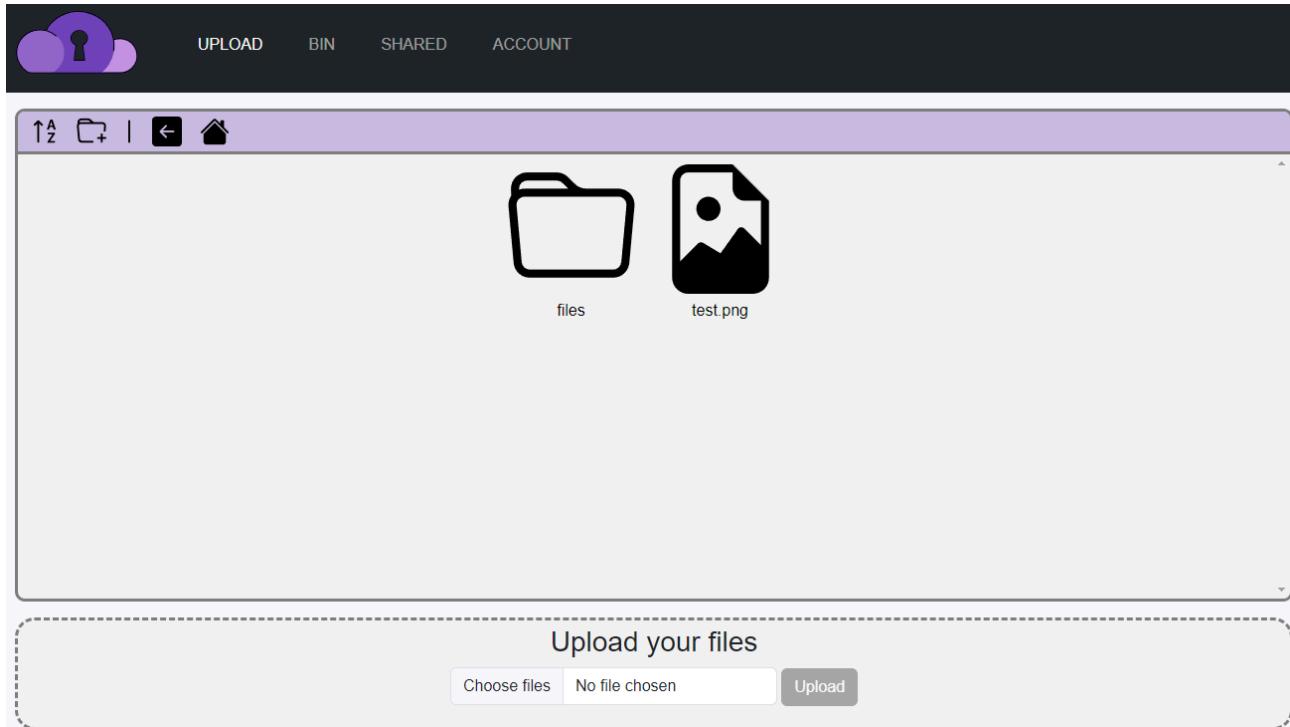
## Prototype Snippets (*Ben*)



Above shows how we wanted the upload page to look. We included the navigation bar which had the search and sort (in the top right) along with our logo (in the top left). Overall it was a very simplistic design which is what we were going for as it only needed to have an upload button with the files displayed.

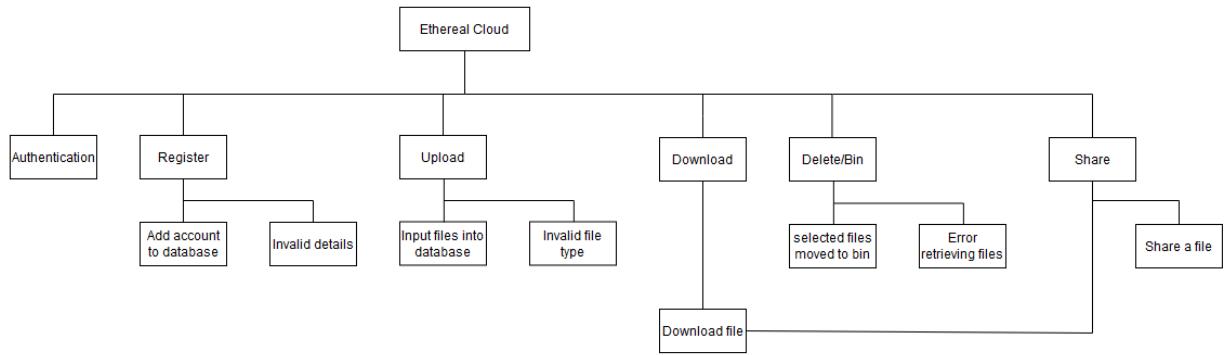
### Final:

As you can see from the below screenshot of the final product it looks somewhat similar in terms of the overall layout:



We decided to remove the sidebar as we considered it to be in the way when we could have everything on the navigation bar at the top. The upload page is somewhat similar with the file upload box and the file/folders being displayed.

## HIPO Chart (*Riley, Ben*)

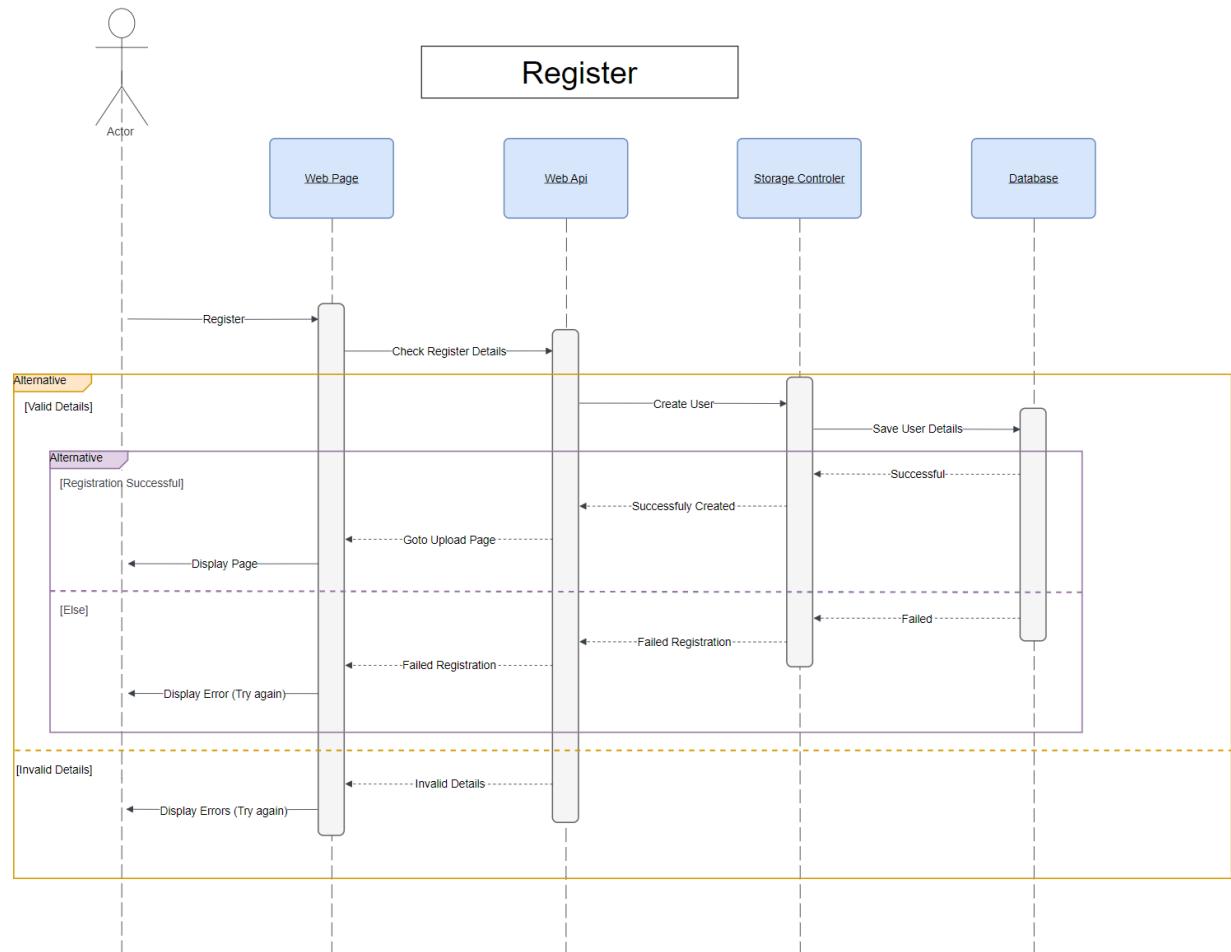


# Sequence Diagrams (*Ben*)

The sequence diagrams show how the different parts of our product interact in each specific scenario.

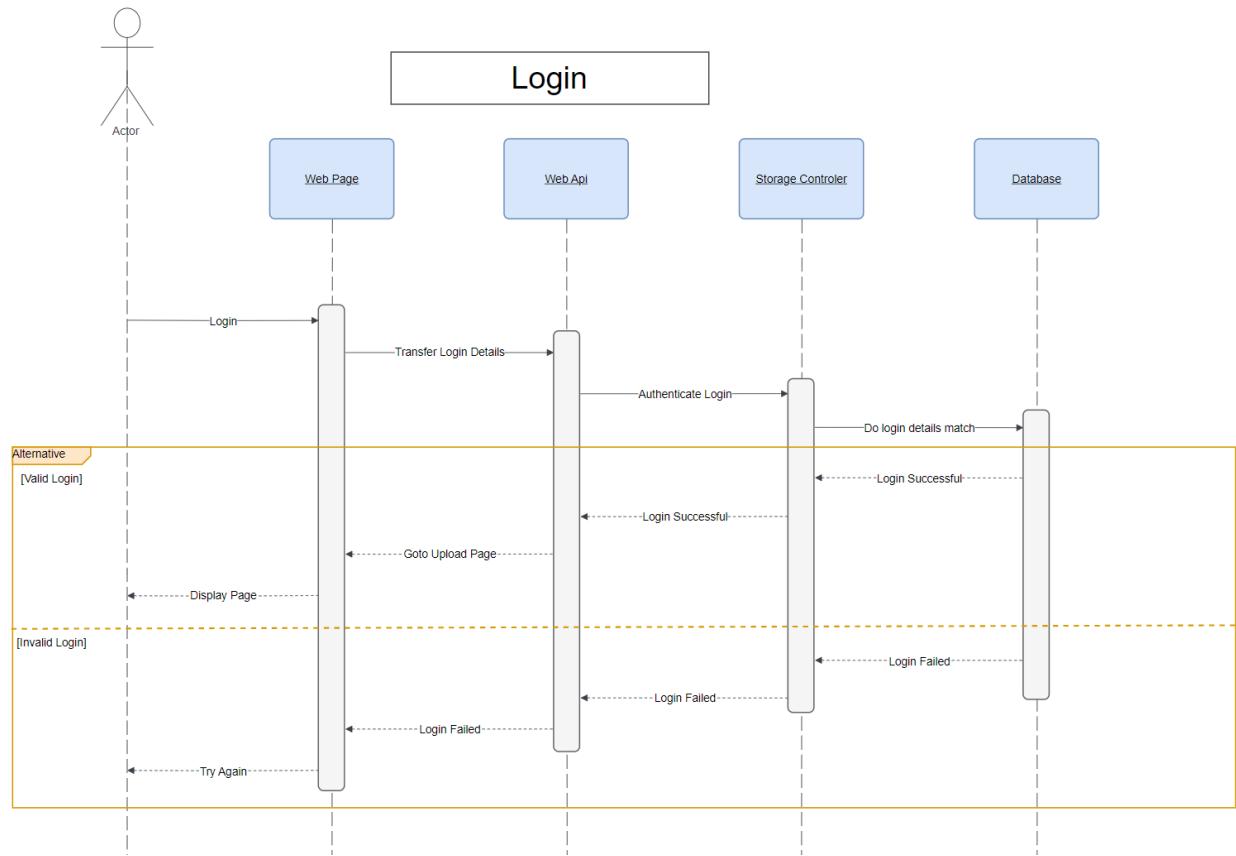
## Register

This diagram indicates how the system handles a user creating an account.



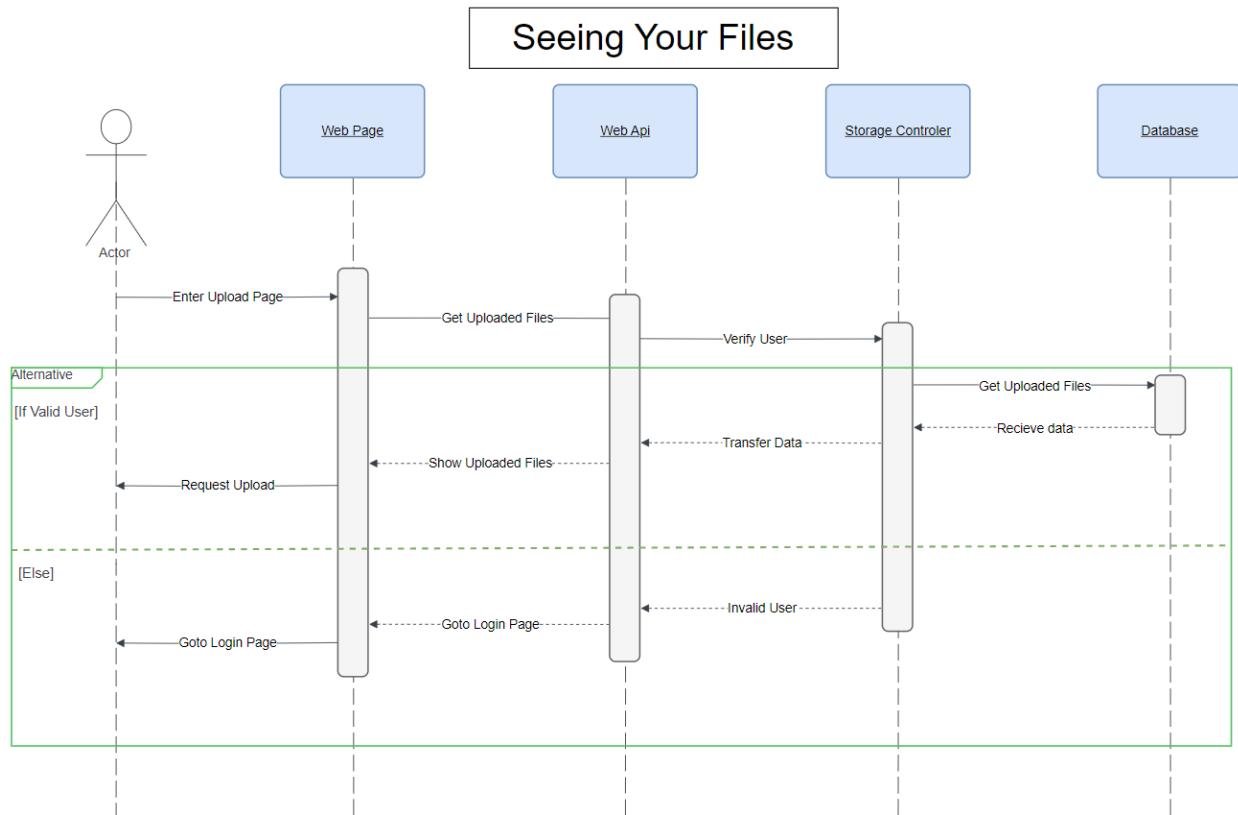
## Login

Below shows the login system.



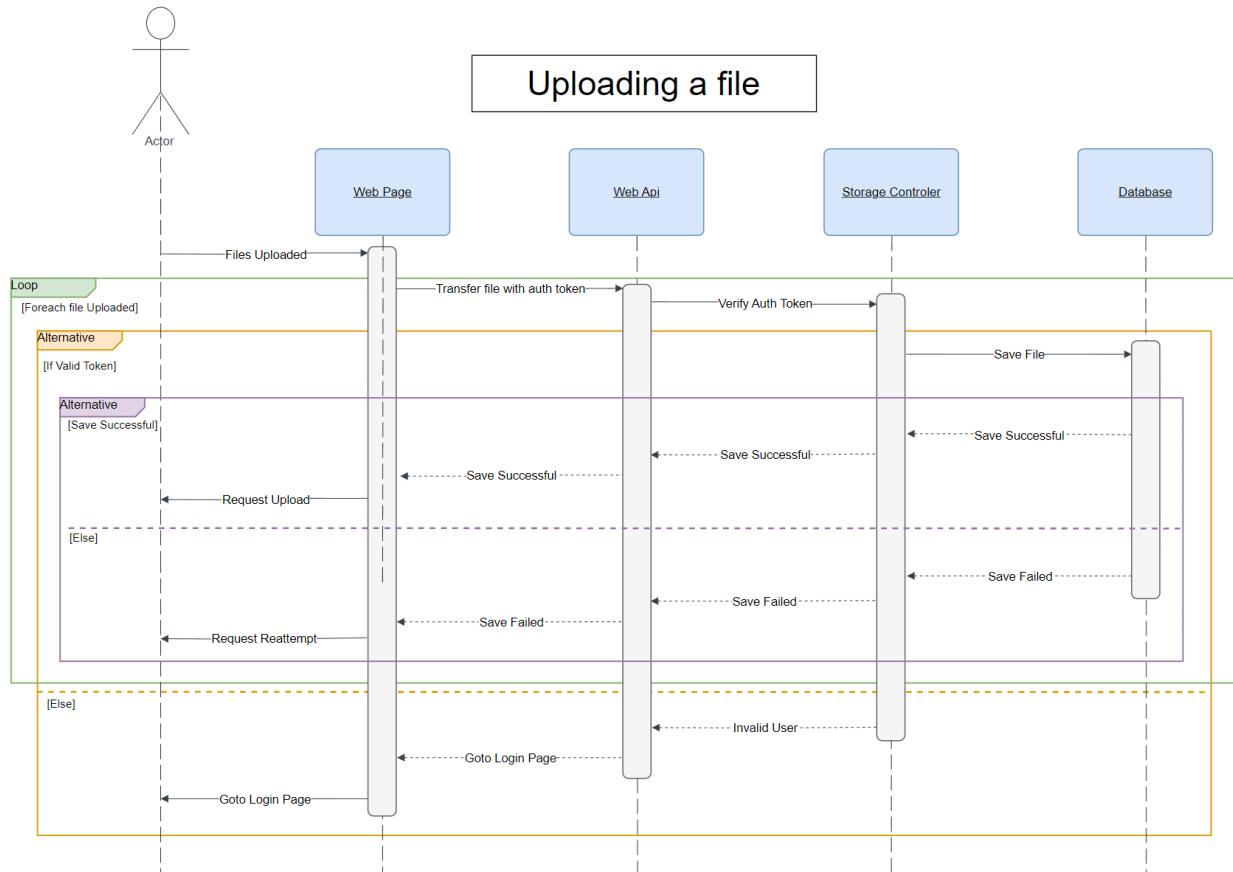
## Seeing your files

When the user enters a page a request is sent to retrieve the required files and display them. The diagram below demonstrates this process for the upload page however the process is identical for the bin and shared pages.



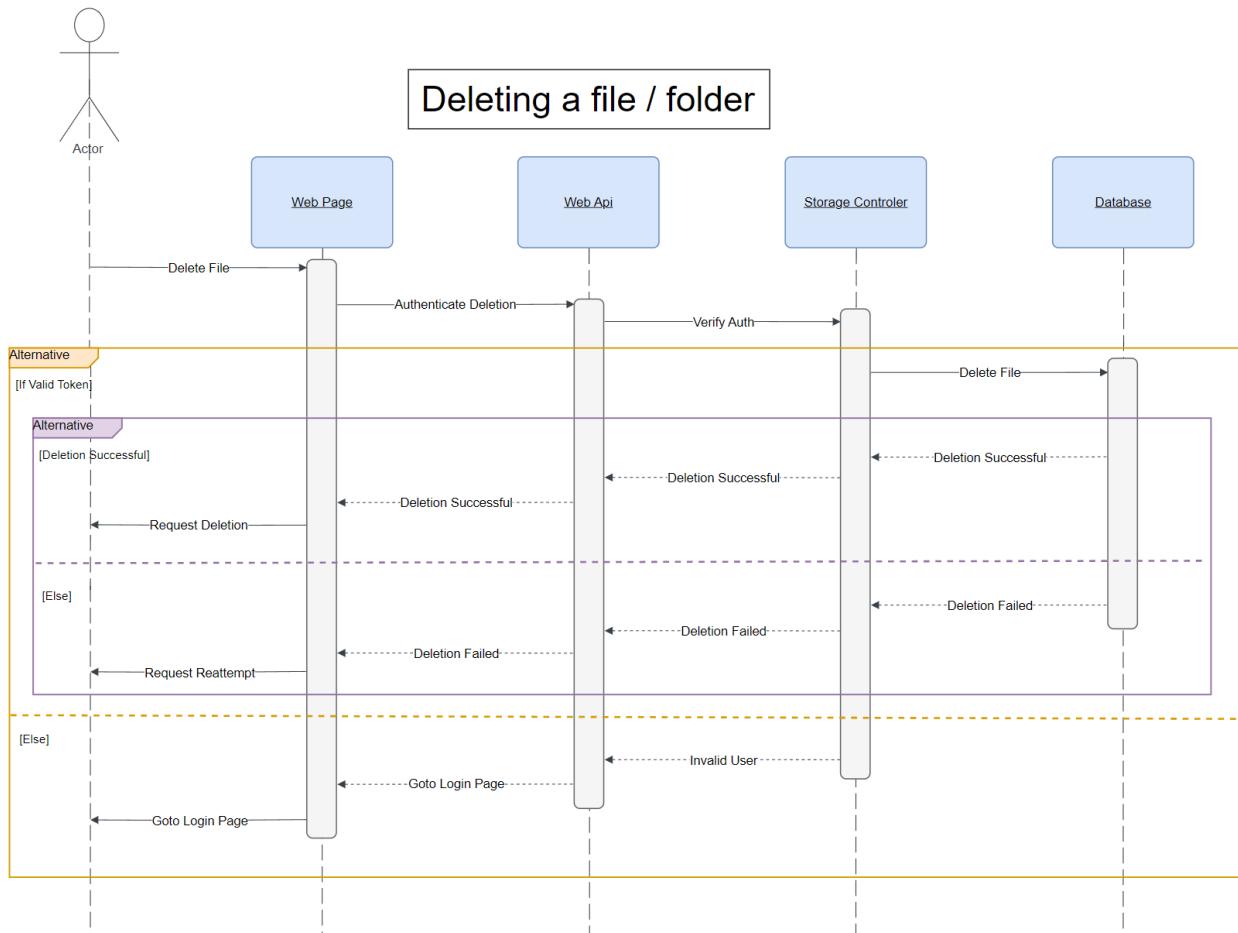
## Uploading a file

Files when uploaded by the user can be done in bulk or one at a time. The diagram below shows how this would be handled.



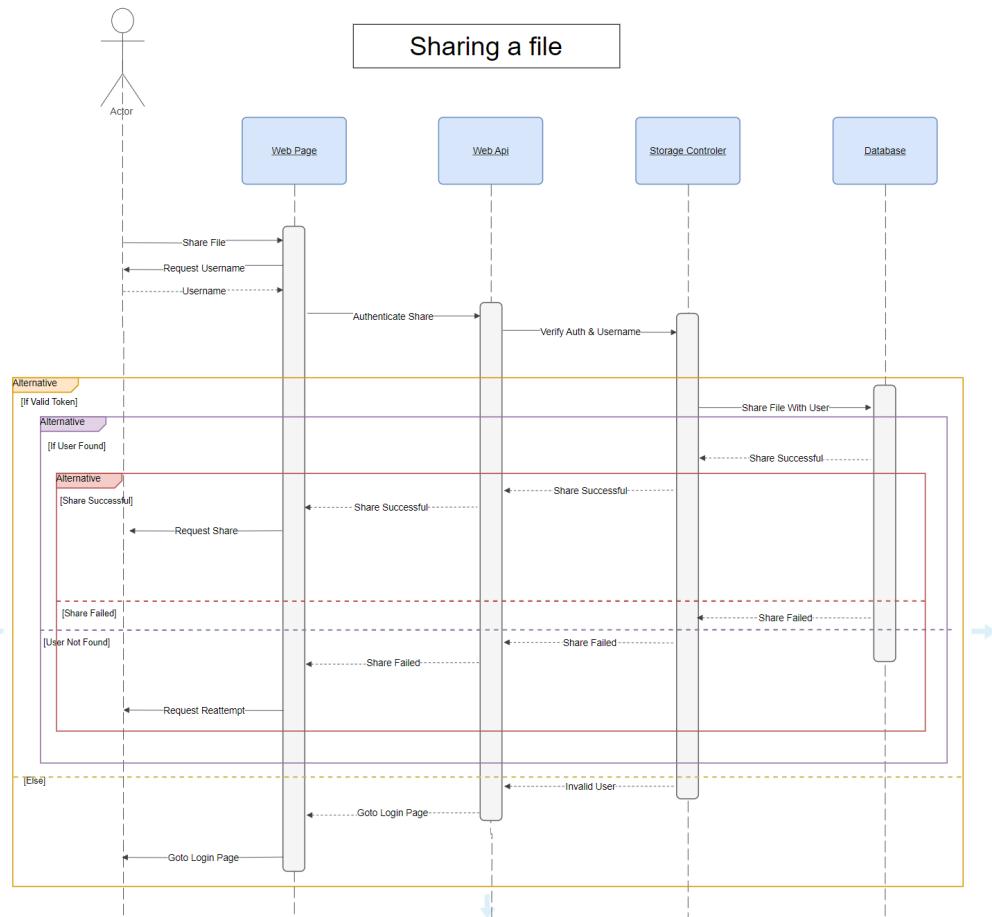
## Deleting a file / folder

The process of deleting a file or a folder is identical.



## Sharing a file

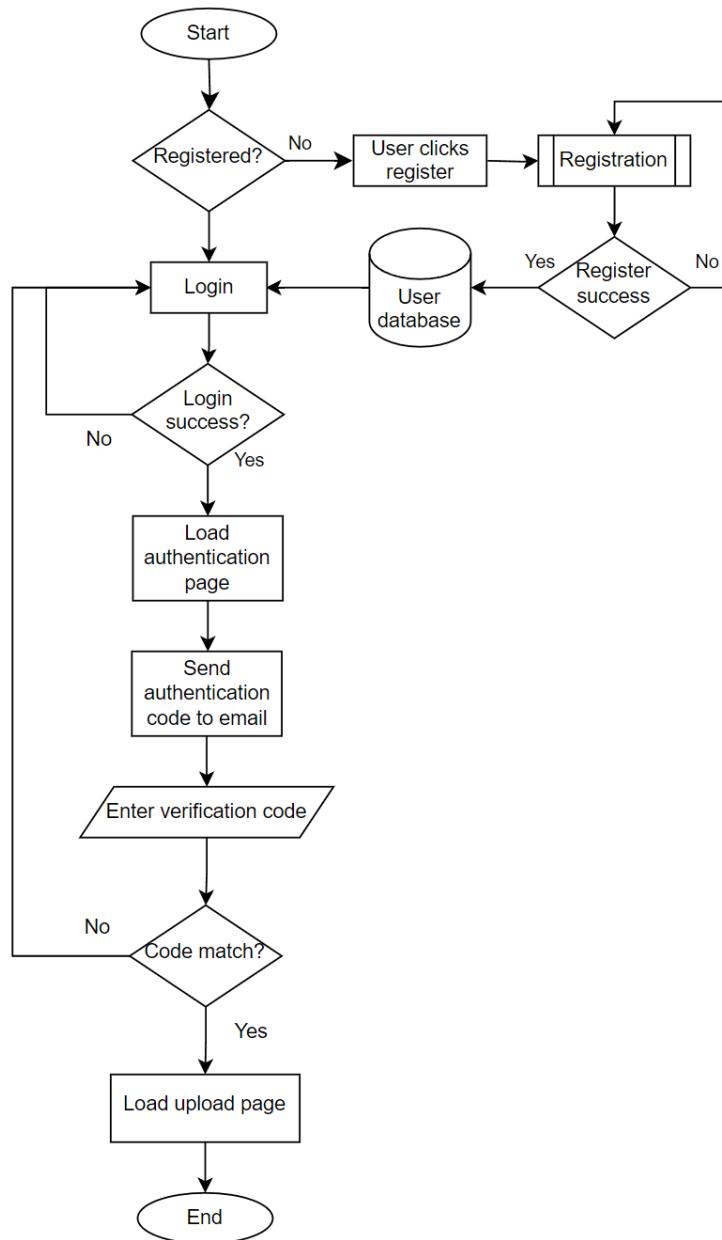
Below shows what occurs when a user tries to share a file with another user.



## Flowcharts (Ben)

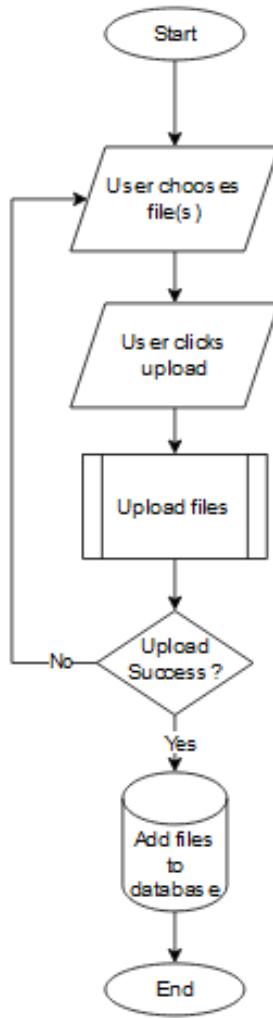
### Login and register page

The login and register shows the process in which the user would create a new account and login. If they already have an account on the home page they would simply select login and that would go straight through to the authentication page.



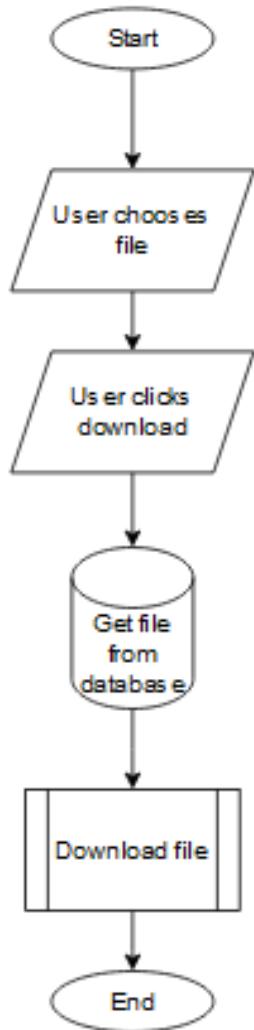
## Upload

Here you can see the process the upload page goes through in order for the user to upload files.



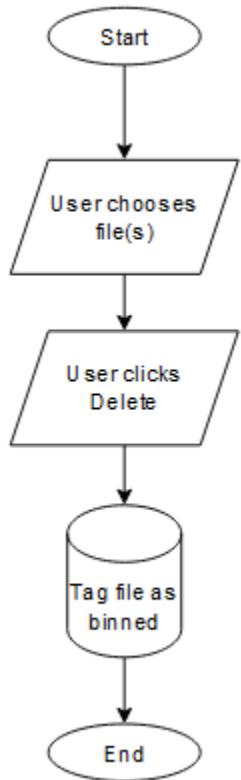
## Download

The download flowchart shows the steps taken for the user to download a file.



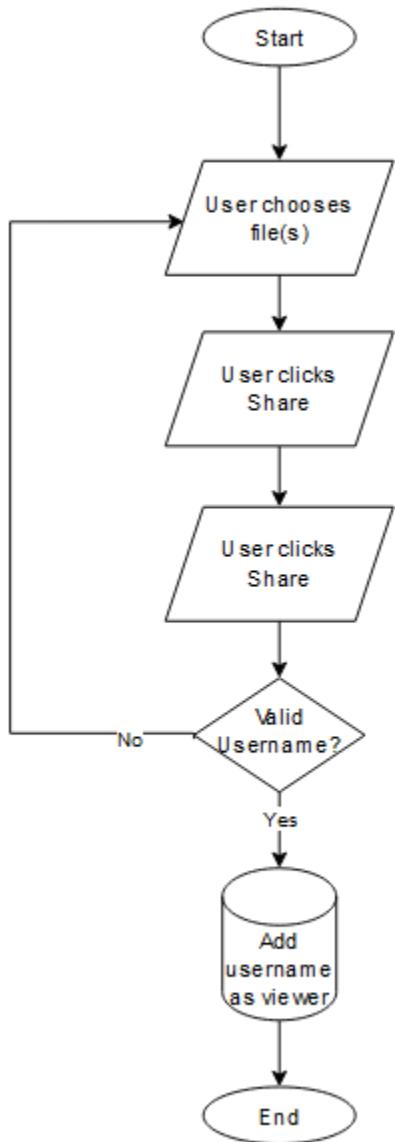
## Delete

The delete flowchart shows the steps taken for the user to delete a file or a folder.



## Share File

The share flowchart shows the steps taken for the user to share a file with another user.



## **Quality Assurance (Eli)**

After all of the necessary functions of the system had been created and integrated, we then decided to focus on the quality of the project by analysing our system against a rigorous testing plan to target both functional and non-functional aspects of the system.

When discussing functional testing, we are analysing how well the project responds to certain manipulations of data, seeing if there is anything that causes the system to go against the expected result, such as an unexpected user feedback or the outright failure of a function given its intended role against the actual result.

As for non-functional testing, it follows a similar procedure with regards to testing all features against a plan, however the focus here is for performance, usability, accessibility and security. This particular testing was performed after functional testing so as to ensure that the features were fully functional before improving on the quality of user experience and safety.

## Functional Testing (Riley)

You will find below a series of tests against the project around data being passed into the system and how it handles it [7]. There are components that do not have data passed through from the user, but functionality is still tested nonetheless. You can find a section of test files within the git repo at Test Files/Content, some of which I have used in the following tests

ID	Target Page	Test Data	Description	Expected Outcome
1	Index - Register	N/A	The user clicks the register button	User is taken to the register page
2	Index - Login	N/A	The user clicks the login button	User is taken to the login page

ID	Actual Outcome	Pass
1	Success, result is the same as expected	Yes
2	Success, result is the same as expected	Yes

3	Register	Username: "TimLernersB" Email: "testing@test.com" Password: "Testing123!" Confirm Password: "Testing123!" T&C's: True	The user creates an account with valid data	The user is taken to the login page
4	Register	Username: "" Email: "testing@test.com" Password: "Testing123!" Confirm Password: "Testing123!" T&C's: True	The user tries to create an account without a username	Error message "Username is required"
5	Register	Username: "TimLernersB" Email: "" Password: "Testing123!" Confirm Password: "Testing123!" T&C's: True	The user tries to create an account without an email	Error message "Email is required"
6	Register	Username: "TimLernersB" Email: "testing@test.com" Password: ""	The user tries to create an account without a password	Error message "Password is required"

		Confirm Password: "" T&C's: True		"Confirm Password is required"
7	Register	Username: "TimLernersB" Email: "testing@test.com" Password: "Testing123!" Confirm Password: "Testing321!" T&C's: True	The user tries to create an account where the passwords do not match	Error message "Passwords do not match"
8	Register	Username: "TimLernersB" Email: "testing@test.com" Password: "Testing123!" Confirm Password: "Testing321!" T&C's: False	The user tries to create an account without accepting the terms and conditions	Error message "You must accept the T&C's"
9	Register	Input values are larger than expected - 64 characters	A user tries to create an account with input values larger than 64 characters	Error message "The field [violated field] must be a string with a maximum length 64"

3	Success, result is the same as expected	Yes
4	Success, result is the same as expected	Yes
5	Success, result is the same as expected	Yes
6	Success, result is the same as expected	Yes
7	Success, result is the same as expected	Yes
8	No functional error occurs, but can be bypassed using inspect element and enabling the register button	No
9	Success, result is the same as expected	Yes

10	Login	Username: "TimLernersB" Password: "testing@test.com"	A user with a valid account tries to sign in	User is taken to the upload page
11	Login	Username: "" Email: "" Password: "Testing123!"	A user tries to log in with no username or email	Error message "Username or Email is required"

12	Login	Username: "TimLernersB" Password: ""	A user tries to log in with no password	Error message "Password is required"
13	Login	N/A	A user clicks the "Forgot Password?" link	The user is taken to the forgot password page

10	Success, result is the same as expected	Yes
11	Success, result is the same as expected	Yes
12	Success, result is the same as expected	Yes
13	There is no forgot password page	No

14	Authentication	N/A	A user receives a code via email	The user has a code in their inbox
15	Authentication	Code: "123456"	The user enters the valid code given via email	User is sent to Upload page
16	Authentication	Code "1234567" or Code: "12345"	Code entered is outside of expected range	User is sent back to the Login page
17	Authentication	Code: ""	No code is given by the user	Warning "Please fill in this field"
18	Authentication	Code"abcdef"	Code is not a number	User is sent to the Login Page

14	Success, result is the same as expected	Yes
15	Success, result is the same as expected	Yes
16	Success, result is the same as expected	Yes
17	Success, result is the same as expected	Yes
18	Success, result is the same as expected	Yes

19	Upload	Selected File: "PDFBig.pdf"	The user chooses a file to upload	Name of file is seen next to the choose files button
20	Upload	Selected File: ""	The user tries to upload when no files selected	Error message "Please select a file"
21	Upload	Rename "PDFBig.pdf" to "PDFBig.txt"	A user tries to rename the file extension of a file	File is renamed without affecting file type extension
22	Upload	Multiple files of choice	The user is able to select and upload multiple files	Files are uploaded, displayed in the file explorer
23	Upload	N/A	User tries to upload a selected file using the upload button	File is uploaded, displayed in the file explorer
24	Upload	N/A	User tries to download a file from the website	File is downloaded
25	Upload	New_Name: "Test"	A user wants to rename their file to something else	The new file name is as the user inputs
26	Upload	New file name is longer than 64 characters	A user tries to rename a file to something larger than 64 characters	Error message "File name should be a maximum of 64 characters"
27	Upload	A valid account name that exists	The user shares a file with another valid, existing user	Success, file can be seen in the "Shared With Others" tab within the shared page
28	Upload	N/A	User presses delete on a selected file	The file is removed from upload and can be found on the bin page

19	Success, result is the same as expected	Yes
20	No functional error, however no user feedback	No
21	Success, result is the same as expected	Yes

22	Success, result is the same as expected	Yes
23	Success, result is the same as expected	Yes
24	Success, result is the same as expected	Yes
25	Success, result is the same as expected	Yes
26	No functional error, no max length on <u>renaming</u> of file	No
27	Success, result is the same as expected	Yes
28	Success, result is the same as expected	Yes

29	Upload	Folder name: "Test"	User wants to add a folder, clicking the new folder button and entering a name	Folder added, displayed in the file explorer
30	Upload	Folder name is longer than 30 characters	A user tries to create a folder with a name longer than 30 characters	Error message "Folder name should be maximum 30 characters"
31	Upload	Folder name: ""	A user tries to create a folder with no name	Error message "Folder name required"
32	Upload	N/A	The user tries to click the back button when they are already at the main page	There is no change
33	Upload	N/A	The user wants to go from inside a folder to the main page using the back button	The user is redirected to the main page
34	Upload	N/A	The user wants to go from the current folder to the parent folder using the back button	The user is directed to the parent folder

29	Success, result is the same as expected	Yes
----	-----------------------------------------	-----

30	No functional error, however no user feedback	No
31	No functional error, however no user feedback	No
32	Success, result is the same as expected	Yes
33	Success, result is the same as expected	Yes
34	Success, result is the same as expected	Yes

35	Bin	N/A	The user wants to recover a file	The selected file is moved from bin back to upload
36	Bin	N/A	The user wants to recover a folder	The folder and its contents are moved from bin back to upload
37	Bin	N/A	The user recovers a shared file	The file is re-shared with all previous users

35	Success, result is the same as expected	Yes
36	Success, result is the same as expected	Yes
37	Success, result is the same as expected	Yes

38	Share	Target: "TimLernersB" File to share: "TxtBig.txt"	The user wants to send a file to another existing user	The file is visible
39	Share	N/A	The user wants to download a file that's been shared with others	File is successfully downloaded
40	Share	Target: "TimLernersC"	The user tries to share a file to a user that doesn't exist	Error message "User not found"
41	Share	Target: ""	The user tries to share a file with no	Error message "No username given"

			target	
42	Share	N/A	The user renames a file that's been shared with others	The file's name is changed for all shared users
43	Share	N/A	The user shares a file with more than 1 user	The list of users shared with shows all users
44	Share	N/A	The user deletes the file that is shared with others	The file is no longer visible on any shared pages of all users
45	Share	N/A	The user wishes to unshare a file from a specific user by pressing the "X" button next to the username	The target user can no longer see the file in the "Shared With Me" section of the shared page

38	Success, result is the same as expected	Yes
39	Success, result is the same as expected	Yes
40	No functional error, however no user feedback	No
41	No functional error, however no user feedback	No
42	Success, result is the same as expected	Yes
43	Success, result is the same as expected	Yes
44	Success, result is the same as expected	Yes
45	Success, result is the same as expected	Yes

46	Share	N/A	The target user can see the file on the shared page	Shared file is visible in the "Shared With Me" tab
47	Share	N/A	The user wants to download a file that's been shared with them	File is successfully downloaded

46	Success, result is the same as expected	Yes
47	Success, result is the same as expected	Yes

48	Account	New password: "Testing321!"	The user wants to change their password to a valid password	The Password is changed and a notification "Password has been changed"
49	Account	New password: "Testing"	The new password doesn't fit the minimum requirements	Error message "Invalid password"
50	Account	New password: ""	User clicks change password without entering a new password	Error message "Password required"
51	Account	N/A	The user wants to log out by pressing "Logout here!"	User is sent to login page and session cookie

48	Success, result is the same as expected	Yes
49	No constraints, no error message, meaning password can be under minimum requirements	No
50	Success, result is the same as expected	Yes
51	Success, result is the same as expected	Yes

It is important to note that we also performed the above tests on other devices, using a mixture of the "inspect element" tool on chrome to replicate different device models and shapes, as well as utilising the remote version of the website and using our phones. The test results were equally successful. See below for the function-oriented device tests:

ID	Target Feature	Test Data	Description	Expected Outcome
1	Touchscreen	N/A	The user wants to navigate using the touchscreen	The page moves according to the users input
2	Touchscreen	N/A	The user wants to	The page responds

			interact using the touchscreen	according to user input
3	Touchscreen - Keyboard	N/A	The user types using their built in keyboard	The page is compatible with this input method

1	Success, result is the same as expected	Yes
2	Success, result is the same as expected	Yes
3	Success, result is the same as expected	Yes

### Non-Functional Testing (*Ben, Eli*)

ID	Target Page	Test Data	Description	Expected Outcome
1	Index	Change screen size	Use “Inspect Element” to change the screen size	Screen displays well on all screen sizes

ID	Actual Outcome	Pass
1	Success, result is the same as expected	Yes

2	Register	Change screen size	Use “Inspect Element” to change the screen size	Screen displays well on all screen sizes
3	Register	Show hidden password	Enter password and press show	The hidden password is now shown
4	Register	Show hidden confirm password	Enter password in confirm password and press show	The confirm password is now shown
5	Register	password	Enter into password the test data and check the indicator	Indicator shows red
6	Register	Password	Enter into password the test data and check the indicator	Indicator shows orange

7	Register	Password1	Enter into password the test data and check the indicator	Indicator shows yellow
8	Register	Password1!	Enter into password the test data and check the indicator	Indicator shows green
9	Register	N / A	Click the terms and conditions to see the result	The terms and conditions pops up and can be clearly seen
10	Register	N / A	Click the privacy policy to see the result	The terms and conditions pops up and can be clearly seen

2	Success, result is the same as expected	Yes
3	Success, result is the same as expected	Yes
4	Success, result is the same as expected	Yes
5	Success, result is the same as expected	Yes
6	Success, result is the same as expected	Yes
7	Success, result is the same as expected	Yes
8	Success, result is the same as expected	Yes
9	Success, result is the same as expected	Yes
10	Success, result is the same as expected	Yes

11	Login	Change screen size	Use "Inspect Element" to change the screen size	Screen displays well on all screen sizes
12	Login	Show hidden password	Enter password and press show	The hidden password is now shown

11	Success, result is the same as expected	Yes
----	-----------------------------------------	-----

12	Success, result is the same as expected		Yes
----	-----------------------------------------	--	-----

13	Auth	Change screen size	Use “Inspect Element” to change the screen size	Screen displays well on all screen sizes
----	------	--------------------	-------------------------------------------------	------------------------------------------

13	Success, result is the same as expected		Yes
----	-----------------------------------------	--	-----

14	Upload	Change screen size	Use “Inspect Element” to change the screen size	Screen displays well on all screen sizes
15	Upload	Press sort button	Add folders and files to see if they are sorted correctly	The files should be sorted in accordance to the icon
16	Upload	Press Home	Press Home from root directory	No result should occur
17	Upload	Press Home	Press Home from inside a folder	The file path should go back to the root
18	Upload	Double click a folder	Double click a folder to enter the folder	The file path should change and the user should go into the folder
19	Upload	Double click a file	Double click a file to enter the file	The file should be downloaded correctly
20	Upload	Hover over a folder	Move the mouse over the folder icon	The menu appears button appears
21	Upload	Hover over a file	Move the mouse over the file icon	The menu & download appears button appears
22	Upload	Click download button on file	The download button should appear after hovering over the icon	The file downloads correctly
23	Upload	Click folder menu button	The menu shows	The menu appears

			more options of how to interact with the folder	correctly and contains rename and delete options
24	Upload	Click file menu button	The menu shows more options of how to interact with the file	The menu appears correctly and contains download, rename, share and delete options
25	Upload - folder menu	Click rename option	A popup is used to allow users to enter a new name	A popup should show up to enter the new name
26	Upload - rename popup	TestName	The rename popup is used to enter a new name for the folder or file	The folder should be renamed to TestName
27	Upload - folder menu	Click delete option	The folder and all its content inside should be deleted (moved to the bin)	The folder should now be in bin
28	Upload - file menu	Click download	After the menu button is clicked on a file the menu option download is clicked	The file should be downloaded correctly
29	Upload - file menu	Click rename option	A popup is used to allow users to enter a new name	A popup should show up to enter the new name
30	Upload - rename popup	TestName	The rename popup is used to enter a new name for the folder or file	The file should be renamed to TestName
31	Upload - file menu	Click delete	The file should be deleted (moved to the bin)	The file should now be in bin
32	Upload - file menu	Click Share	A popup is used to allow users to enter the username to share the file with	A popup should show up to enter the Username
33	Upload - share popup	TestUsername	A user with TestUsername is available	The file should be shared with the user

14	On pages with small height the upload box gets cut off by the bottom bar	No
15	Success, result is the same as expected	Yes
16	Success, result is the same as expected	Yes
17	Success, result is the same as expected	Yes
18	Success, result is the same as expected	Yes
19	Success, result is the same as expected	Yes
20	Success, result is the same as expected	Yes
21	Success, result is the same as expected	Yes
22	Success, result is the same as expected	Yes
23	Success, result is the same as expected	Yes
24	Success, result is the same as expected	Yes
25	Success, result is the same as expected	Yes
26	Success, result is the same as expected	Yes
28	Success, result is the same as expected	Yes
28	Success, result is the same as expected	Yes
29	Success, result is the same as expected	Yes
30	Success, result is the same as expected	Yes
31	Success, result is the same as expected	Yes
32	Success, result is the same as expected	Yes
33	Success, result is the same as expected	Yes

34	Bin	Change screen size	Use "Inspect Element" to change the screen size	Screen displays well on all screen sizes
35	Bin	Press sort button	Add folders and files to see if they are sorted correctly	The files should be sorted in accordance to the icon

36	Bin	Press restore button	The folders and all its content inside should be restored (moved to the upload)	The folder should be in upload
37	Bin	Press restore button	The files and all its content inside should be restored (moved to the upload)	The files should be in upload

34	Success, result is the same as expected	Yes
35	Success, result is the same as expected	Yes
36	Success, result is the same as expected	Yes
37	Success, result is the same as expected	Yes

38	Shared with others	Change screen size	Use "Inspect Element" to change the screen size	Screen displays well on all screen sizes
39	Shared with others	Press sort button	Add folders and files to see if they are sorted correctly	The files should be sorted in accordance to the icon
40	Shared with others	Double click a file	Double click a file to enter the file	The file should be downloaded correctly
41	Shared with others	Hover over a file	Move the mouse over the file icon	The menu & download appears button appears
42	Shared with others	Click download button on file	The download button should appear after hovering over the icon	The file downloads correctly
43	Shared with others	Click file menu button	The menu shows more options of how to interact with the file	The menu appears correctly and contains download, rename, share and delete options
44	Shared with others - file menu	Click download	After the menu button is clicked on a file the menu option download is clicked	The file should be downloaded correctly

45	Shared with others - file menu	Click rename option	A popup is used to allow users to enter a new name	A popup should show up to enter the new name
46	Shared with others - rename popup	TestName	The rename popup is used to enter a new name for the folder or file	The file should be renamed to TestName
47	Shared with others - file menu	Click delete	The file should be deleted (moved to the bin)	The file should now be in bin
48	Shared with others - file menu	Click Share	A popup is used to allow users to enter the username to share the file with	A popup should show up to enter the Username
49	Shared with others - share popup	TestUsername	A user with TestUsername is available	The file should be shared with the user

38	Success, result is the same as expected	Yes
39	Success, result is the same as expected	Yes
40	Success, result is the same as expected	Yes
41	Success, result is the same as expected	Yes
42	Success, result is the same as expected	Yes
43	Success, result is the same as expected	Yes
44	Success, result is the same as expected	Yes
45	Success, result is the same as expected	Yes
46	Success, result is the same as expected	Yes
47	Success, result is the same as expected	Yes
48	Success, result is the same as expected	Yes
49	Success, result is the same as expected	Yes

50	Account	Change screen size	Use "Inspect Element" to change the screen	Screen displays well on all screen sizes
----	---------	--------------------	--------------------------------------------	------------------------------------------

			size	
51	Account	Show hidden password	Enter password and press show	The hidden password is now shown

50	On mobile devices the logout here gets squished and its internal text escapes	No
51	Success, result is the same as expected	Yes

## User testing form (*Eli*)

Below is our user testing form, which gave us valuable feedback on our project. We gave this out to a few people to get some feedback on our design and interface. Here were some of their responses:

Login page:

What do you think of the colour scheme?

7 responses

seems ok

As someone with poor eyesight the background colour needs to be more defined/darker.  
(Note to devs: background colour could be the same as the middle colour of the logo)

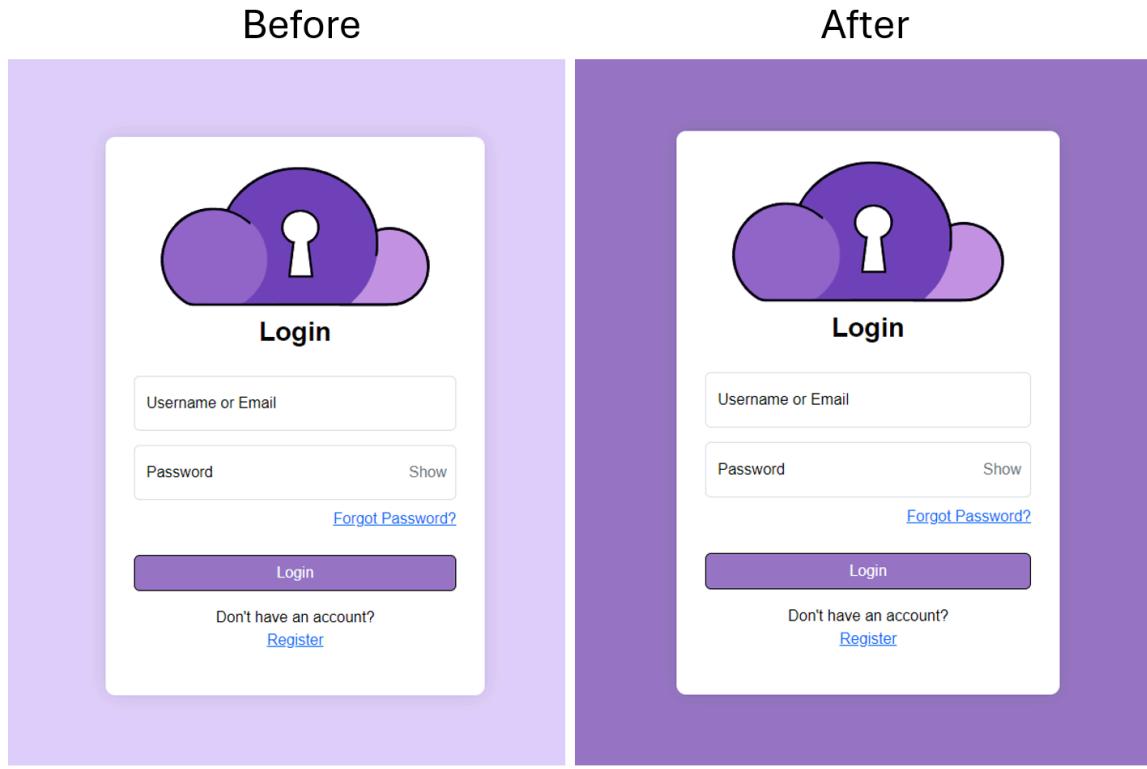
Appealing, not too overstimulating.

i like the colour scheme. it a relaxing colour.

Here they said they liked the purple colour scheme but didn't like how similar the background colour was to the content colour. So, we changed the background colour to be a much darker purple we fixed this in this commit:

<https://github.com/Plymouth-University/comp2003-2023-25/commit/5f7d065c216970930f4a00fad8fc7ea2eb886b80>

Here shows the before and after of the colour changes:



In regards to the register page submission:

Any other comments?

2 responses

got confused when finished registering, didnt realise i had to log in.

This was saying that the register button didn't redirect from the register page once they created an account. We fixed this in this commit:

<https://github.com/Plymouth-University/comp2003-2023-25/commit/167b5bc3230161dc4ecd4e281f097a47f6b0df8c>

So now when they create an account instead of finding the login button themselves it just automatically redirects on successful account creation.

Upload page:

Originally we didn't have specific icons for each file type which meant it was hard to distinguish between them. So, to fix this, we added icons for each of the common file types, for example PDF, Docx or HTML.

Can you easily distinguish between 2 files that are successfully uploaded?

7 responses

yes

Yes

By title yes, but not by the icon type.

yes because it tells you.

The excel file didn't have an identifiable icon

Before and after changes:

Before



After



When uploading a file a few of our testers clicked on the upload button, which caused an error as they didn't have a file selected to upload.

## Uploading a file

Were there any issues with uploading the file?

7 responses

no issues

no, very straightforward

Initially clicked the wrong button (Upload instead of choose file.) Was able to upload once I had selected the correct button.

I initially clicked on upload and not choose files.

We fixed this issue by making the upload button greyed out until a file has been selected or dropped into the upload box. This was the before and after of our upload button:

## Before

Upload your files

Choose files

No file chosen

Upload

## After

Upload your files

Choose files

No file chosen

Upload

By doing this it made it impossible for a user to press upload with no files selected.

There were no issues with the uploading of files, everyone found that easy enough to do.

What file type did you try to upload? What was the end result?

7 responses

text file. successful

Text. Successful upload

Word document.

word and image file.

image

PDF, it uploaded correctly

image (jpg) double files uploaded

Creating folders also didn't pose an issue as it is clearly labelled as a new folder when you hover over the icon.

### Create a folder

Is it clear on how to add a new folder on the website? If not, how could this be improved?

7 responses

yes very clear

yes

Yes

Yes very simple.

it took a while to find the add a file folder but thats only because its been a while since i have done it.

no, icon needs a hover description

Overall on the user testing they only found issues with the icons looking too similar, accidentally uploading nothing, and the colour scheme. The rest of the feedback was very positive showing our user interface is simple and understandable to other users. They all agreed that the login and register was clear and easy to navigate

## Conclusion *(Will)*

Overall, we achieved most of the objectives we set out to complete, having most of the core features we wanted available on the platform functional to a good degree, furthermore, our process for this project worked quite well for our group with each of us splitting off into different areas of the project to take responsibility for. This meant that there was always someone working on each part of the project, ensuring that all parts of the project were kept up to date with each other and none fell behind in terms of features or polish as well as that each of us knew exactly who to talk to in order to bring issues to light with the different parts of the solution. Consequently, having the areas split up like this ensured that we could all specialise into our areas of the project instead of having to balance learning multiple new technologies at once, leading to faster and higher quality work on the solution.

There were several hurdles we had to overcome in order to bring this project to where it is today, such as learning about Docker and how to use it, how it works and how to utilise it to make hosting the solution easier. On top of that, we had to learn about other tools that would assist us during development, such as entity framework core for database interaction and Razor pages for the website that would improve development speed.

Even with those tools to assist us and speed up development, we had to drop certain features from the final product, such as file encryption, as we did not have time to implement them to a degree that we would deem suitable to be included in the solution. This is potentially due to overestimating how much we could get done in the time given, however, estimating this will only get better with experience. Therefore, if we were to do this project again, we would lower the amount of objectives we wanted to complete by a decent margin in order to ensure that we could get the features done within the given timeframe, as well as ensuring that the implemented features are of a high quality.

We initially set out to produce an easy to use and accessible cloud storage platform that can be viewed and utilised on any device at any time (provided an internet connection is available, of course). Looking back at the project, we have achieved these goals, as our website works on computers, across different browsers, and on a variety of mobile devices. On top of this accessibility, all of the core features are in place for a cloud storage platform, such as uploading files, downloading files, deleting files and sharing files, encompassing the basic foundation for a full cloud storage website, only needing extra features developed on top of this foundation to be feature complete.

Furthermore, the solution that we have built suits what our client envisioned for the project to a high degree, being a cloud storage solution with a certain set of essential features that people would expect from such a website, such as uploading and downloading files as well as deleting and sharing files. However, not all features are present as we had to drop some in order to ensure the quality of the product, therefore features such as file encryption, upgrading storage capacity and more were not included.

## References (Ben, Riley, Will, Eli)

- [1] Schwaber, K., 1997. Scrum development process. In *Business Object Design and Implementation: OOPSLA'95 Workshop Proceedings 16 October 1995, Austin, Texas* (pp. 117-134). Springer London.
- [2] GDPR (2018). *General data protection regulation (GDPR)*. [online] General Data Protection Regulation (GDPR). Available at: <https://gdpr-info.eu/>.
- [3] Agile methodology Kumar, G. and Bhatia, P.K., 2012. Impact of agile methodology on software development process. International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2(4), pp.46-50.  
[https://www.researchgate.net/profile/Gaurav-Kumar-175/publication/255707851\\_Impact\\_of\\_Agile\\_Methodology\\_on\\_Software\\_Development\\_Process/links/00b49520489442e12d000000/Impact-of-Agile-Methodology-on-Software-Development-Process.pdf](https://www.researchgate.net/profile/Gaurav-Kumar-175/publication/255707851_Impact_of_Agile_Methodology_on_Software_Development_Process/links/00b49520489442e12d000000/Impact-of-Agile-Methodology-on-Software-Development-Process.pdf)
- [4] Scrum methodology Srivastava, A., Bhardwaj, S. and Saraswat, S., 2017, May. SCRUM model for agile methodology. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 864-869). IEEE.  
[https://eresearch.stikom-bali.ac.id/admin/files/publikasi\\_files/406b9aa48840ed8d357af87f8bb62530.pdf](https://eresearch.stikom-bali.ac.id/admin/files/publikasi_files/406b9aa48840ed8d357af87f8bb62530.pdf)
- [5] How to prevent DDoS attacks (N.D). [online] Available at:  
<https://www.cloudflare.com/en-gb/learning/ddos/how-to-prevent-ddos-attacks>
- [6] Awati, R. (2022) What is Hypertext Transfer Protocol Secure (HTTPS)?, Software Quality. TechTarget. [online] Available at:  
<https://www.techtarget.com/searchsoftwarequality/definition/HTTPS>
- [7] How To Create A Test Plan (Steps, Examples, & Template) (2023) TestRail. [online] Available at: <https://www.testrail.com/blog/create-a-test-plan>
- [8] Docker Documentation. (2021). *Reference documentation*. [online] Available at:  
<https://docs.docker.com/reference/>.
- [9] jamesmontemagno (2022). *Docker containers, images, and registries*. [online] learn.microsoft.com. Available at:  
<https://learn.microsoft.com/en-us/dotnet/architecture/microservices/container-docker/introduction/docker-containers-images-registries>.
- [10] tdykstra (2023). *Enforce HTTPS in ASP.NET Core*. [online] learn.microsoft.com. Available at:  
<https://learn.microsoft.com/en-us/aspnet/core/security/enforcing-ssl?view=aspnetcore-6.0&tabs=visual-studio%2Clinux-ubuntu> [Accessed 1 May 2024].

[11] Rick-Anderson (2023). *Generate tokens with dotnet user-jwts*. [online] learn.microsoft.com. Available at:

<https://learn.microsoft.com/en-us/aspnet/core/security/authentication/jwt-authn?view=aspnetcore-8.0&tabs=windows>.

[12] IEvangelist (2024). *Containerize an app with Docker tutorial - .NET*. [online] learn.microsoft.com. Available at:

<https://learn.microsoft.com/en-us/dotnet/core/docker/build-container?tabs=windows&pivots=dotnet-8-0> [Accessed 1 May 2024].

[13] Docker (2021). *Docker security*. [online] Docker Documentation. Available at:

<https://docs.docker.com/engine/security/>.

# Appendix

## Eli Bowen

### **Technical skills achieved:**

I have achieved the ability to program in HTML and JavaScript. The project has allowed me to expand my knowledge and understand it better. As I am the frontend developer, I have had the ability to play around with various CSS styles and HTML code to ensure the website runs smoothly. I also used some JavaScript that allows popup boxes to appear to rename and create folders.

I learned about Docker and its use in hosting containers that can serve as web servers for database administration and file hosting during this assignment. With the help of C# and Razor, I actively coded the website, enabling smooth front-end and back-end interaction. In addition, this project has helped me hone my already strong HTML and CSS coding abilities. Notably, prior to working on this project, I had very little experience with Razor; hence, this experience gave me important insights into how it interacts with website code. I want to continue using my programming abilities to further improve my work.

### **The soft skills I achieved:**

Teamwork was a major part of the group project both with meeting up on Wednesday every week and attending the meetings with Tony on Fridays. Teamwork was an integral part of this project without it the project would not have succeeded. As I was the frontend developer, I relied on my team to do the backend of the website so that I could solely work on the frontend to make the website look good for future users of the site.

Time management was also a crucial skill where I allocated enough time to fix bugs, and strategically planned which parts of the website I was going to fix before working on it. This allowed me to spend more time on the issues and make it a fully functional page before moving onto other features.

Communication again is another important skill, I improved on without effective communication of what needs adding. For example, when creating the login page, I had to communicate to my team that I need the backend for the login and register buttons to work and submit the login form to the database.

I showed leadership when outlining the bugs within the project and instructing my team to fix them and what needs doing to fix them. Also, towards the end of the project, I took it upon myself to say what needs to be done. This included me making to do lists and ensuring that everyone knew what they were doing.

**Challenges overcome:** I didn't realise that picking a cloud computing project would require so much work. When we first thought of the idea of doing a cloud storage website, I didn't anticipate all the various elements within the scope of the project. For example, the encryption needed both HTTPS, and file encryption, then session cookies for users. Also, the upload/download functionality with the ability to delete and restore files. There was a lot we needed to decide including figuring out which aspects are important and should be included within our project, and which can be forgotten about due to time constraints and extra complexity.

**Experience with group mates:** Overall, everyone in my group had a role to play. We all decided to stick to the same roles throughout the project mine being the frontend developer. There were no issues with distractions or anyone lacking behind we all kept up to the project demand depending on what was needed either to be coded or discussed.

The organisation within the group was good, we had consistent communication, this helped to enable us to understand where we all were and what needed doing. This collaborative approach led to a good team dynamic, with no conflicts, and ensuring smooth workflow without disruptions.

Since we all had roughly the same level of knowledge on creating a cloud service, we decided to divide the project into manageable chunks. I chose to get our GitHub page set up, and I oversaw organising GitHub and creating a functional file system. However, as I was the developer who created and set up the website, Docker containers, and GitHub, I didn't get to spend much time on the Trello board or the documentation side of things.

**Things learned from dealing with clients:** My initial impression of the meetings were that they needed to be extremely professional and formal but then realised that the meetings were fairly relaxed and casual. Interacting with Anthony Edwards (Tony) was good, he gave us ideas and improvements to make to our project. For example, he told me to include normalisation for the css, this helped to fix the issues we had with some browsers looking different than others. It also provided an insight into the real-world application of their work and helps tailor solutions to meet client requirements effectively.

**Incidents that gave valuable experience:** Incidents like overcoming technical obstacles, resolving conflicts within the team, successfully meeting client expectations, and delivering high-quality results under pressure. These experiences contributed to my personal and professional growth, enhancing skills and resilience. In terms of disagreements there weren't many, and the ones we had were more about which features we should implement and which we should drop. So the experience with my team overall was really good.

**How I will continue to grow after this module:** After completing the module, I will continue to grow by pursuing further education or certifications, participating in workshops, or training sessions. I want to engage in self-directed learning opportunities to apply my skills in real-world projects. Additionally, seeking feedback from peers and mentors and reflecting on past experiences will help me to identify areas for improvement and set goals for future development.

This project has given me these extra skills needed to fulfil projects in the future and I will look back on this and use the experience learned to improve on upcoming projects.

## **William Harding**

### **Technical Skills**

In terms of the technical skills that I achieved or grew throughout this project, I would like to bring mention to my new acquired knowledge regarding technologies such as Entity Framework Core, Docker and furthering my experience with ASP.NET and C#. More specifically, I learnt how to utilise EFC in order to effectively connect to a database and create tables and relationships and use it to query the database to add new data or retrieve data from the different entities. In regards to docker, I learnt roughly how containers work, able to manipulate auto generated docker files to enable features such as persistent file storage in docker containers, how docker volumes work to enable this persistent storage and how to build docker images and create containers from them for hosting. My skill development with ASP.NET and C# included few new skills overall, however, I became a lot more comfortable working in and around C# based projects, being able to read through code to find bugs and errors faster and solve them.

### **Soft Skills**

The soft skills I developed further during this project mainly were my ability to pick up new technologies faster by applying previous experience and knowledge to learn about the new tools in a faster and more sure fire way for understanding, allowing me to be more adaptable with my workload and what I can do in short periods of time. I also improved my teamwork as working in a team such as I did gave me a new perspective regarding working on projects in this vein, improving my communication and teaching me to work together more closely with others to fulfil our goals more efficiently.

### **Challenges**

Before this project, I had barely any experience with entity framework core, and thus had to pick it up quickly in order to build a solid foundation for the project, allowing me to speed up development when adding endpoints and generally progress my knowledge and experience. I also had to learn to deal with the pressure that came with being the main developer of the backbone of the solution. Without the storage controller, the website would not be able to add any features as they would not have access to the database or buckets to save or retrieve files.

### **Group Experience**

Overall I had an extremely positive experience with my group mates, we worked together very well, and, after a couple of hurdles, our communication improved greatly and we all got along well. Furthermore, working with my group has been a pleasure and an eye opening experience for working together with others towards a shared goal that we could only achieve as a collective. We also built mutual trust in one another so that we could rely on each other for help if we can't solve a problem by ourselves, or just require assistance to finish off a piece of work.

Each group member had a part to play in the project, with all 4 of us covering all of the responsibilities that needed to be shouldered in order to allow a smooth development experience. We all had our own areas of the project that we mainly worked in, mine being the storage controller for example, constructing an environment where we each knew who to go to for support for any manner of bugs or errors.

## Experience with Client

Through my involvement with the client for this project, I gained extremely valuable experience and learnt a lot about communicating with external individuals who are involved in the project, keeping them up to date on progress and developing my ability to effectively convey information to others. Furthermore, I learnt how important it is to maintain steady updates to the client in order to keep them in the loop regarding progress of the solution, as they could have important insight for the project, and may have suggested changes or recommendations for moving forward with development.

## Growing with Experience

I would like to bring attention to a particular incident including me and Ben, the Bucket Incident where me and Ben had a roughly 15 minute debate on how the buckets were to work in the project, as we had a misunderstanding of us both misinterpreting what the other was saying, leading to a time where we both meant the same thing, but with different wording. My main take away from this experience is that high quality communication is important for a smooth experience when deciding on the specifics of a project, ensuring that no miscommunications will occur and solidifying that we are all on the same page when it comes to what we mean and can agree on the details of what is being discussed.

## Moving Forward

Moving on from this project, I will be carrying forward the experience and skills gained, hoping to further progress them and increase the breadth of my knowledge of as many subjects as possible to become both more competent, experienced and knowledgeable, allowing me to share this experience to help bring up others. I will be seeking out more complicated and bigger projects that will test myself in various skill sets, allowing me to develop myself further as both a programmer and computer scientist.

Continuing on, I plan to seek out more shared projects to heighten my communication skills as well as learn new skills, new technologies and utilise more tools to become more well rounded in my skill sets and be able to approach problems from new angles and give more insightful commentary regarding problem solving and decision making on large projects.

# Riley Coulstock

## Technical skills

Over the course of both semester 1 (MVP) and semester 2 (Final Product), I had to polish and, in some cases, relearn or even start from scratch, HTML, CSS and JavaScript. From this I then had to learn alongside the group how to work with cshtml files (an integration of C# and HTML) also known as RAZOR pages.

Specifically in semester two, I was tasked with implementing the back end for the password strength checker (bar underneath the password on the register page) and email verification code (the thing that sends the code to the user's email). For the strength checker, I learnt how to utilise RegularExpressions (or regex), which were integrated into the models (which were refined by me). With the email verification, I had to learn how to utilise a client (outlook) to send a randomly generated code. Both of these were successful and I am proud of what I have learned in this module.

## Soft skills

My main development in this module were my soft skills, as I took a leadership and documentation role. During semester 1, I mainly kept track of the live documents (Monday, Miro, Trello) but rarely extended beyond that, only providing analysis and breakdown of the requirements for the report (which was completed evenly between the group) and providing points of improvement and proofreading.

During semester 2, upon receiving feedback, I took a slightly different approach. On top of the roles from semester 1, I also got more involved in the project, which helped me make more informed and useful decisions about features and processes by testing each stage of development which proved extremely helpful in discovering several fatal bugs. By being more involved and taking charge of the testing, it also helped improve team coordination and camaraderie as I was constantly exchanging ideas and bugs with the whole team, some of which were very amusing (at one point, all users' files were visible to everyone).

## Challenges

The main challenge I had (which was discussed above) was that I was not as involved in the coding of the project as I should've been, which led to some pitfalls in the MVP report as I had to enquire frequently about the system.

I overcame this by trying to take on lower priority programming tasks (which in summary are: DB and model constraints, password strength checker, 2 factor authentication helper) which required me to understand the project to some extent if I were to integrate by parts into the project. With the use of testing, I was able to modify and interact with the system in a different light than the other group members which allowed me to gain a unique insight which helped with the development process.

## **Group experience**

As a group, we performed exceedingly well. We had lots of chemistry and even though 3 of the 4 members study Cyber Security, we still had a diverse skillset which covered each other's strengths and weaknesses.

Eli was an excellent front-end developer, whenever we needed some user interfaces, he would create them with efficiency and style. Will was the backbone of the back-end, with his skillset naturally leaning towards the storage controller, he proved vital to the creation of data management and storage. Ben was the jack-of-all-trades, effectively tying the front end and back end together via endpoints (and where necessary notifying Will for specific endpoints).

I am very happy that I got to work with such an effective team, as I believe our chemistry and skills blended together to make a challenging project come to life.

## **Client experience**

We were a start-up, which meant that we had fortnightly meetings with Anthony Edwards (Tony). My initial impression was that the meetings were to be extremely professional and of utmost importance to notify of every tiny piece of progress (like in those videos about nightmare software dev jobs).

After the first meeting, this impression was flipped upside down, it was a casual 15-minute meeting to see how we were coming along and if we were stuck on anything, which was wonderful, with me keeping track of everyone's progress, the meetings were very smooth as we all knew what we had done in the past sprint before the call. He was more than happy to review drafts and answer questions outside of "business hours" and it wasn't frowned upon if one of us ended up asking a question at 4 in the morning, as he would reply as soon as he could.

Overall, I have learnt how to interact with a client and their needs, as well as how to notify them of any change in plans.

## **Incidents**

There were very few clashes within the group, with the exception of mild disagreements about concepts, but otherwise no disagreements on implemented features (as they would be brought up for discussion before a final decision was made as a group).

## **The future**

This module has taught me a lot in regards to technical skills, soft skills, group work, and handling expectations of a self-set and client-oriented nature. In later projects of my own making or assignment, I will take these lessons into account and ensure that project progress is smooth and that I do not fear getting my hands dirty and asking for help when I feel that things are in over my head.

## **Ben Sanders-Wyatt**

### **Soft skills I have gained from this project**

My work with ASP.NET core has greatly improved my skills with HTML, CSS and javascript; the work has improved my frontend website development abilities by being able to work first hand and in depth.

Learning how to use Razor and incorporate it effectively into our project has gained me the experience as a full stack developer to understand how the frontend and backend of our websites can communicate and how to ensure they are doing so effectively. The work with Razor pages has also increased my knowledge of API's communication as I constantly made requests between the Web-Api and storage controller.

Familiarity with docker and how it functions has been a valuable learning experience. My experience with setting up a HTTPS environment with docker was a great source of this experience and gave me a greater understanding of how docker functions and how I could create and use images, containers and volumes in more projects in the future.

Database wise my skills have improved as I was required to interact constantly with the data retrieved from the tables. While working with Will to use this data has gained me a greater understanding of how to effectively create a database and interact with it.

### **Soft skills I have gained from this project**

In regard to soft skills, teamwork would be one of my most valuable learning experiences as I was able to gain understanding of how to communicate clearly with my teammates in order to work together smoothly.

As the “middleman” between frontend and backend I gained a greater knowledge of the overall project. This naturally led to me leading the direction of the project as I had the most understanding of the project as a whole. This allowed me the opportunity to greatly improve my leadership skills by organising Eli at the frontend and Will at the backend to complete related tasks at similar times.

My work with Riley on the Monday sprints gained me the skills to organise and plan out effective objectives, risks and time spans for each step of the project as well as ensure that I maintain my own time management by sticking to the sprints. These skills in organisation and leading what a team's tasks are will only become more important as I work on more projects in the future.

### **Challenges I overcame**

The main challenge I overcame was learning how to use and implement Razor effectively into our project as I had no previous knowledge about the tool. This was a challenge as I needed to be able to perform my part in the team; my solution for this was to allocate myself more time to

research and practice, in addition to my working time, in order to gain more experience in the subject.

### **How well did my team work together?**

As a team I believe we communicated well and were able to come up with solutions and compromises to issues quickly. I believe we worked so effectively together because we were comfortable with sharing ideas and talking through them to work towards the best solution.

At the beginning of the project we discussed our strengths and weaknesses and this allowed us to take on the best roles fitting to our areas and skillset. However the reason behind this success is that no one on our team was afraid to help each other out with their parts and that we were all each happy to take on more roles as the need arises.

Overall I think that we worked together as a team extremely well and that this was a major reason behind the success of our project.

### **Client interactions and what I gained**

I learnt that I shouldn't be afraid to ask Tony (our client) for feedback on specific parts of the project or the overall whole.

I also discovered the importance of informing your client of the processes the team will be taking to create the desired product. While I believe we had no major conflicts with Tony the consistent meetups lead me to understand how important communication between the project team and the client can be.

### **How I learnt from incidents**

An incident I constantly look back on is what our team now calls the "Bucket Incident". This started as a discussion between myself and Will about how we would implement the buckets to store files. During this discussion we kept trying to understand what each other was saying, however the base misunderstanding between the two of us just kept building into more and more confusion, when in the end we were both trying to explain the same thing. The reason our team still brings this up is because I would continue to bring it up to remind my teammates that we should ensure we are in complete understanding about topics discussed. This was a good learning experience for me as I have the experience of how to communicate more effectively within my team.

Apart from this incident we were able to communicate effectively and prevent any major disagreements.

### **What does my future hold?**

I plan on further enhancing my experiences with Docker, Razor and databases by creating my own web application. My goal with this new project is to use the skills I learnt from this module

and transfer them to the Blazor framework so I can further grow my skills in full-stack development.

I believe the experience from this project will be very useful going forward in my future work with teammates and clients and I would like to use the skills I have learnt in preparation for gaining Microsoft's Azure development certifications.