# UAT Test Plan for Ethereal Cloud Storage

## Group 25

[Written by Eli Bowen]

**1. Scope** ..... **3**
1.1. Objectives and business requirements ..... 3
1.2. Scope ..... 3
**2. Testing team** ..... **4**
**3. Milestones and deliverables** ..... **5**
3.1. Design & testing process ..... 5
3.2. Staging environment ..... 5
3.3. Training ..... 5
3.4. UAT Execution ..... 6
3.5. Reporting & data analysis ..... 6
**4. Environmental requirements** ..... **7**
4.1. Hardware requirements ..... 7
4.2. Software requirements ..... 7
**5. Features to be tested** ..... **8**
5.1. Uploading a file ..... 8
5.1.1. Pass/fail criteria ..... 8
5.1.2. Test cases ..... 8
5.2. Restore from bin ..... 8
5.2.1. Pass/fail criteria ..... 8
5.2.2. Test cases ..... 8
5.3. Features to avoid testing ..... 9
5.3.1. The azure database ..... 9
5.3.2. Docker building scripts ..... 9

# 1. Scope

## 1.1. Objectives and business requirements

Our main goal is to provide an elegant, easy to use, user friendly solution; it is maintainable, scalable, and easy to expand upon in the future, in the process of doing so, gain more experience to provide better performance for the final product.

We are hoping to accomplish a fully functioning cloud storage server that will let others take the docker image and host it themselves for use in both personal and business use. We will measure success by the number of customers that come to us for a copy of the working solution so they can then host it.

## 1.2. Scope

We were trying to create a simple and easy storage solution for our clients to enable on premise access locally instead of using an outside cloud service.

For this UAT test, we'd like to:

- Test the upload and bin pages to ensure it accesses the database and stores the data of the files properly.
- Get feedback on our website from other users.
- Verify there are no obvious bugs, this involves checking all the pages to ensure all elements such as buttons and links work correctly.

We are not testing:

- The azure database
- Docker building scripts

# 2. Testing team

| Name | Responsibilities |
|---|---|
| Eli Bowen | UAT Coordinator - handles communication between end users and QA team.<br><br>Design test cases for the core functionality<br><br>Design test cases for the optional functionality:<br>● Password strength checker<br>● Change password function<br><br>Create a user testing form for anonymous feedback on Google Forms.<br><br>Create test data for the core and optional functionalities.<br><br>Write UAT report, including scope, milestones & deliverables and the designed tested features. |
| William Harding | |
| Benjamin Sanders-Wyat | |
| Riley Coulstock | |

# 3. Milestones and deliverables

## 3.1. Design & testing process.

Initial designs created in figma: https://www.figma.com/file/PZBpXhuQxZR9EwRCeYYxK6

The testing will occur in 4 stages:

1. Staging environment: set up by Eli, this environment should closely mirror production. It will be used for testing and creating a snapshot of the production database.
2. Training: UAT testers will be trained by Eli. We're holding UAT meetings in the first few weeks of February.
3. UAT execution: create test cases and have our testers/reporters report on said test cases.
4. Reporting: full data analysis, bug findings, and meeting on what remains to be done.

The testing will take place during the start of February once everyone has been trained on how to perform the tests. The tests will be conducted by using test data and test plans; these will be laid out beforehand and everyone will be trained on how to fill them in properly. The deadline for this will be February 26, 2024.

## 3.2. Staging environment

Our staging environment will be available for all UAT testers on EtherealCloudTesting@outlook.com

We will create a copy of the production database and onboard users via their usual profile. This will be done using the localhost copy of the docker containers so that there isn't any actual data getting submitted to the real website. Making it both a safe and good testing environment. The deadline for the staging environment is March 4, 2024.

## 3.3. Training

We will be holding UAT meetings during the final weeks of February leading into March. We will have Riley set up those meetings as he is responsible for documenting the meeting plans. Then he will walk them through what the new feature does and how it works.

The first meeting will be 45 minutes long and present new features and business objectives. Here we will discuss our scope and the final product including the looks, and how it will all integrate.

Second meeting will be 1 hour, talking about how to log into the staging environment, and talk about new features and how to go about testing them.

The third meeting will be around an hour for a debrief on how to report the test cases and where to log them.

The deadline for this training will be March 12, 2024.

## 3.4. UAT Execution

The execution will take 3 days. During this, everyone will be trained on how to perform the tests.

The steps involved in executing these tests are:

1. Onboarding. Getting each of our testers individually and helping them with staging the environment to get it ready for testing.
2. Test case execution. Each tester will be given specific tests to carry out. These are specified under the reporting and data analysis section.
3. Once done, there will be a meeting to report back on any issues found, and to get feedback from the testers on what they've discovered.

Deadline for UAT execution: March 18, 2024.

## 3.5. Reporting & data analysis

Here we'll go through the individual test cases to understand what the testers struggled with and to get some general feedback. The analysis will help to improve our website design and give us valuable feedback on things such as accessibility issues or buttons not functioning correctly.

The deadline for reporting & data analysis is March 25, 2024.

# 4. Environmental requirements

## 4.1. Hardware requirements

The hardware requirements for simply loading the website on a device:

| | **Minimum** | **Recommended** |
|---|---|---|
| **CPU** | Dual-core 1.5GHz | Quad-core 2.1GHz |
| **RAM** | 2GB RAM | 8GB RAM |
| **Storage** | 2GB | 10GB (for uploading large files) |

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:

| | **Minimum** |
|---|---|
| **CPU** | Quad-core 2.1GHz |
| **RAM** | 8GB RAM |
| **Storage** | 100Gb (2 Gb per user) |
| **Network Speed** | 100 mb/s |

This is specifically for hosting the cloud on a server within a business or in a home environment. The reason for 100GB of storage is to allow extra storage when users upload large amounts of data. We considered the 8GB of RAM as minimum because when we ran our tests it used around 6GB and that was for around 10 users using the website consistently for 10 minutes.

## 4.2. Software requirements

The only software that needs to be installed is Docker for hosting our website using the containers and a suitable web browser (this can be any).

For larger business they may need to pay for a docker subscription

# 5. Features to be tested

## 5.1. Uploading a file

### 5.1.1. Pass/fail criteria

**Pass:** The user can upload a file with no issues and can upload more than one file if they wish to.

**Fail:** The user can't upload a file or multiple files, it crashes or responds with an error message.

### 5.1.2. Test cases

The features will be tested as follows:

- Building docker
- Logging into the website
- Selecting the upload box on the upload page
- Selecting multiple files
- Confirming the chosen files
- **Here it will either crash or upload the files successfully**

## 5.2. Restore from bin

### 5.2.1. Pass/fail criteria

**Pass:** The user can restore a deleted file from bin and it contains the same data as it did before delete it.

**Fail:** The user can't restore the file from bin or it restore but with the wrong file data.

### 5.2.2. Test cases

The features will be tested as follows:

- Loading docker
- Logging into the website
- Clicking on a file in the upload page
- Selecting delete
- Going to bin page
- Restoring that file by clicking on it
- Re-downloading the file to check its contents.
- **Here it will either restore properly or it won't**

### 5.3. Features to avoid testing

#### 5.3.1. The azure database

#### 5.3.2. Docker building scripts

## Signoff

I hereby accept this final product. (Yes/[redacted])

(Signature)

Client Name: Anthony Edwards

Date: 30/04/2024