

BTech450 Project Paper - End of Semester Report: Digital Asset Management System

Dan Kane

Abstract. The need for a single location for storage of digital assets that can be made readily available from any where, has been the greatest push behind the number of different digital asset management (DAM) systems coming out in the previous few years. This paper covers the current stage of my project, in creating a DAM system. It will go over DAM systems in general, and what the specific type of DAM system is that I am creating. The methodology chosen and various reasons for it, comparing and contrasting different approaches to development and implementation will all be explained, as well as what processes I have gone through at each stage to get to where I am at the moment. The project will be completed by the end of the second semester so the end aims of the project and the processes I will be using to get these aims completed will be included in the report also.

1 Introduction

Digital Asset Management Systems are not a new idea, having been around for a least 10years already, but the evolving state of the Internet and line speeds has meant that they are much improved on what they used to be. In simple terms they are web-based systems, with an underlying storage system, normally a database. The web base system is usually displayed to the user using a thin browser based client, however some approaches have used a software based client that the user installs on their home machine and this then connects to the web based system.



Fig. 1. System consists of a user interface connected to a web system with an underlying database

1.1 Reasons for the project

The idea for the project's development comes from the company sponsoring the project, Digital Stream. Their customers had come to them asking for this kind of system, because they were happy with their current solution for storing their data. Digital Streams also are looking for more constant revenue, and have a DAM system will allow them to charge a monthly storage cost to those users uploading their digital assets.

1.2 How the project will be made

The project has been asked by Digital Streams to be developed using ASP.Net on a SQL Server Database. This would be the traditional way of doing this, but however there have been new developments in this area, and there are new technologies coming out. These technologies such as, Windows Communication Foundation, and Windows Presentation Foundation are newer, more improved versions of ASP.Net, so I will be researching into these new technologies before any actual implementation is started.

2 Project Methodology

There are multiple ways of developing systems, such as iterative, waterfall, agile etc, that all encompass their good points and bad, and they all have their own situations where they should be used. For this project I have tried to choose a methodology that fits with a single person project development and fits with this type of project, as this project doesn't involve a lot of coding and class development, but more interface development and database design.

2.1 The Waterfall Methodology

The waterfall methodology is one in which systems are designed and developed in stages. The commonly used stages are Requirements specification, Design, Implementation, Integration, Testing and debugging, Installation, and finally Maintenance. The waterfall method states that each stage should be completed before moving to the next one. However this is where problems begin, as it is generally hard to perfect each stage before moving to the next one, and often when developing a system there is a lot of back and forth between stages.

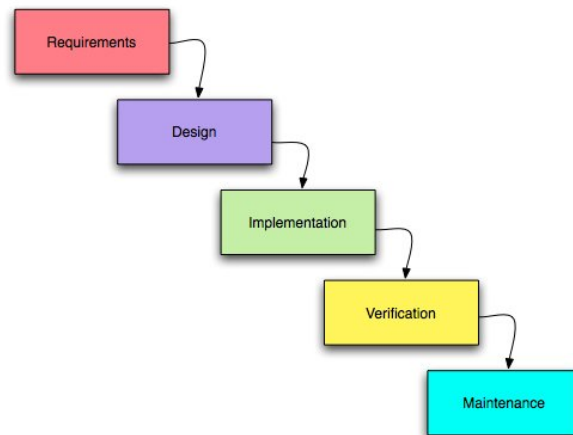


Fig. 2. The Waterfall Methodology

2.2 Agile Software Development

Agile methodologies develop software and systems by way of time boxes, which are short periods of time, typically around 4 weeks that includes the development of a new functionality for the system. The time box contains all the processes needed to develop the new functionality, planning, requirements analysis, design, coding, testing, and documentation. Developing in this way, creating the system piece by piece (or functionality by functionality) means that risk is reduced, as if the development is disbanded then the functionalities developed can still be reused in other developments. Agile methodologies use face-to-face communication rather than the production of documents. The development team (including customers) is usually located in one place where it is easy for communication. Instead of documentation at early stages, agile methodologies use working software to display progress. This results in a lack of documentation for these methodologies. Since this project is a single-person development I have decided this is not a good methodology to follow, and as a university paper as well, I need the documentation for handing in reports.

2.3 The Chosen Methodology

For the methodology that I chose to use for this project, I took the main points of the waterfall methodology but made allowances to interchange between stages. While designing the system, when coming across new requirements, or changes to current ones, I will still return to initial stages to make updates. At the implementation stage too, if I find parts of my design not suitable I will redesign them, rather than just continuing on, and if the redesign requires redefining requirements that's ok too. However as this project involves developing a system that is similar to those made before, I don't imagine that the requirement will change much, making this methodology save to use.

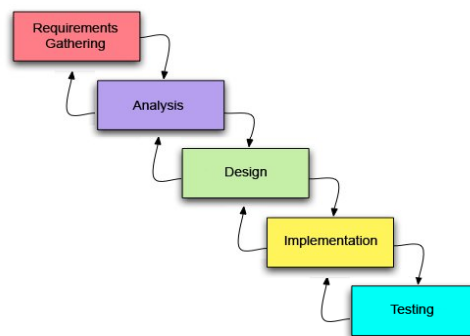


Fig. 3. The Chosen Methodology

3 Research

My research process began by first looking at other companies that had attempted the same sort of project that I am working on. I looked at mostly New Zealand companies but also a couple from overseas. New Zealand companies were more relevant to the requirements of this project. For each system examined, I created a list of the functionality provided, and what made the system stand out.

3.1 E-See[1]

One of the systems I researched into was E-see. It was developed by a New Zealand based company that has been around since 1997. The E-See system is mainly used for brand data management, where companies and advertisers are the main clients. The companies can upload their logos and product images and advertisers can search for them and download. They divided their clients into two categories, Brand Owners and Brand Users. Brand Owners store their brand data by uploading it to the system and Brand Users download it. The system is free it use for Brand Users to download and

search, but for Brand Owners they need to purchase a monthly plan for data space usage. They do have an option for a free Brand Owner account but this is only able to upload and store up to two items, so is only suitable for testing the system. This is one of the good features of E-See. Brand Owners also have the ability to restrict who can access their brand assets which is done via a permission lock on each asset. When uploading an asset the Brand Owner sets if the assets are available to general use or Brand Users must request permission before using it. So when Brand Users search for a particular asset, they will have to option to either download it directly or request permission first. Another one of E-See's strong points is that they do all picture processing for the Brand Owners, any image uploaded is tidied up and then converted into multiple formats for the Brand User to download.

3.2 Original Image[2]

Original Image is also a New Zealand company that created a Digital Asset Management System. However they have developed it in a different direction than that of E-See, where E-See was directed at companies brand data, Original Image has directed theirs towards storing all of a companies digital assets in one convenient location. There is still the option to make the assets available to others, but there is no free user account option, permission is given out from the owner of the account, and they can define what restrictions are put on the account they give out. Original Image also aims to store all sorts of digital assets, not just images, assets like video, audio, documents, etc are all supported.

3.3 Research Summary

Over all the system's looked at they had many parts in common, and only a few specific things each. The systems don't look to be overly complex, Just a user interface connecting to a storage facility, that gives the clients options to search, upload, download, and restrict. I have taken the finding from the research into the analysis stage where the aim was to reduce all the systems finding into what I thought a Digital Asset Management System was.

4 Analysis

From the lists created in the research stage I began to form the basis of my initial requirements document by comparing and contrasting each system that was researched. The end goal of all this was a final requirements document that would contain all the information I about every task my end system would need to perform. I defined four actors in the system Customers, Users, Owners, and Administration. Customers were those that came to the sight without having signed up previously, they would need to be able to signup – either as owners or users – and view product information which

would probably be done by way of a tour, where they would see screenshots of the major functions of the system.

Creating a Use-Case diagram, and subsequent documentation, helped with not only requirements but also later with the design too. The requirements of the systems functionality directly mapped on to the use-cases in the diagram.

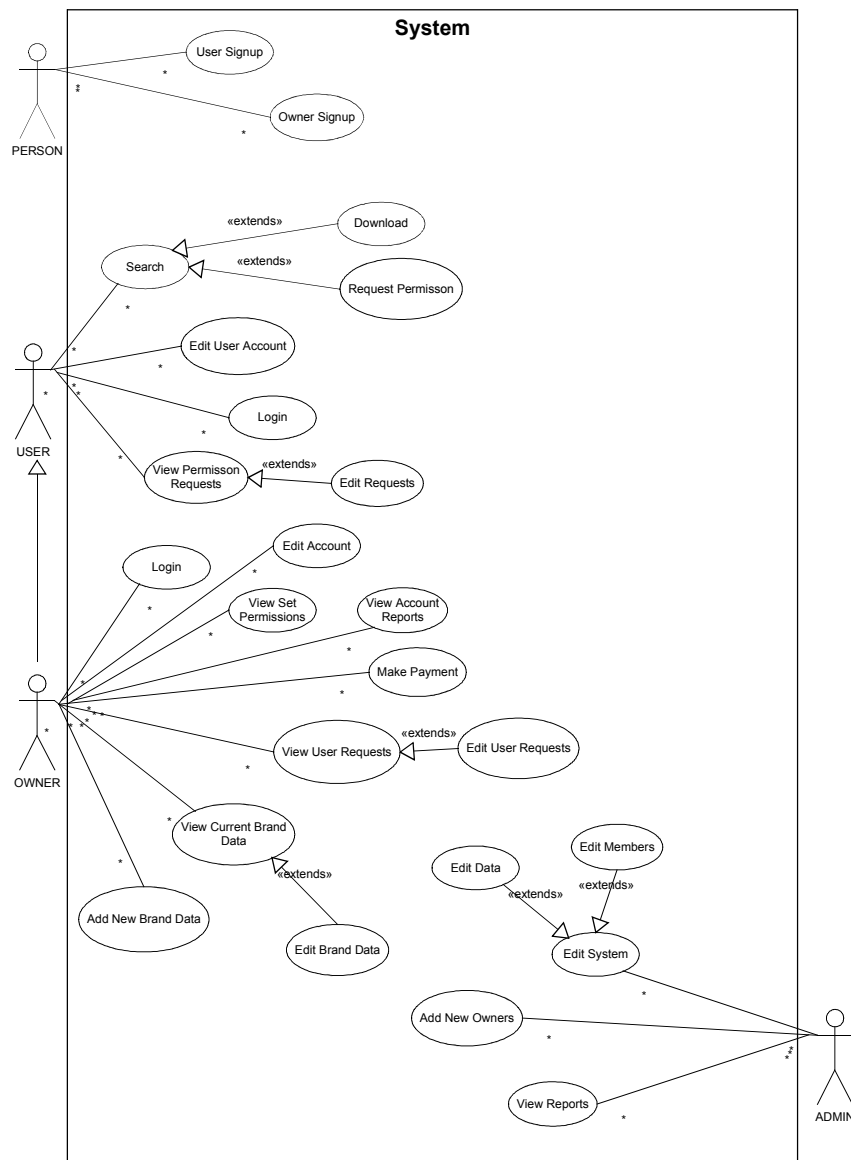


Fig.4. Use-case diagram for the system

4.1 Users

Users are those that signed up to download assets. These accounts would be free to sign up to, and come with no monthly fees. Users would need to perform more tasks than a customer but less than an owner. The user needs to be able to login to the site using their chosen ID and password. They need to be able to initiate a search for the asset they want, this will be split into two categories, simple and advanced. The simple search will be able to be accessed from the user's main page and just consist of a single text box input, which will search through asset names, companies, and descriptions. The advanced search will have its own page, and give the searcher multiple choices on what to search under. They will be able to specify what terms they wish to search and what type of search they want to do. For example they can do an advanced search on the tag type of ProductID under the category of Plasma Television with the search term PTV19944. This will search all assets that are in a category called Plasma Television and that have a meta tag called ProductID. Search results will be displayed to the user and they will give the option to download, add to basket, or if the asset is locked, request permission. The add to basket option is similar to a shopping basket in online stores, but instead of paying for multiple goods together you can download them in a compressed form together. The download options given to the user will be via email or directly using HTTP.

If the asset is locked and the user needs to request permission then they can request it by submitting a request reason message that will be sent to the owner of the asset. When the request had been granted or declined by the owner, an email will be sent to the owner informing them of the owner's response. Users will also be alerted to any new responses when they first login to the system.

Other user functions include being able to edit their accounts, and create favorite companies and assets.

4.2 Owners

The owner has many more functions than a user, so more time is required to be spent on analysis and design on their part of the system. Their main functionality adding an asset to the database needs to allow them to specify how their asset is described exactly, this will be done by having preset fields for them to fill out as well as custom defined meta tags that can create descriptions specific to a single or group of assets. The Owner will be able to upload either by sending an email with the description of the asset(s) and the asset(s) attached, or directly via HTTP upload. When they add an item they can also specify the permission status of the asset, if anyone can download – unlimited or limited time – or if they need to request permission first.

Owners need to respond to requests from users, and any requests that haven't been responded to will be shown to the owner when they login as well as sending them an email when the request was made. The response is either to decline or accept, with an optional response message to the user.

Another functionality is being able to view and edit their previously added assets, which will encompass a search tool to let them search through the assets stored in the

database, but restrict the search to the current owners assets. Here the search function from the user will be reused to save time.

Other owner functions include editing their account, changing payment plan, making payments, viewing previous payments, and viewing reports.

The viewing reports functionality is quite important and there will be multiple types of reports that can be viewed, including asset usage, asset overview, account payments etc.

4.3 Administration

The administration will be employees of Digital Stream making updates to the assets in the database and the client's accounts. If an owner sends in an asset using email then an administrator will need to login and add this new asset to the owners account. Administration will also need to be able to edit any assets in the database, including deleting them from the system, just incase anything illegal is uploaded.

Owners who haven't paid their bills will have their accounts suspended until payment is received so administration will need to be able to do this as well. Users who are causing problems with the site may need to have their accounts removed is a functionality required for an administrator.

The other main functionality is report viewing, for this I would like the administration to be able to create their own sort of report, as well as having some commonly used reports preset in. Things like owners who haven't paid, database capacity, daily or monthly system usage etc.

5 Design

The design for this project is being done in phases, user interface, database, and finally business layer. I decided to begin with the user interface as working out what controls were required on each page really helped in the design of the database, and once the interface and database designs are done, it is a much simpler task to develop the connections required between them for the business layer. To help with the design I used the use-case documentation and activity diagrams developed in the analysis stage, I found going through the steps of completing system functionalities was the best way of finding what controls should be on a page by thinking what a client would expect to see when performing a task.

5.1 User Interface

The user interface is quite complex for this system as there are many functionalities that need to be performed by the owner and user. The user interface design began in its most simple form, just working out what pages were required in the system, not thinking about what controls were to be on each page. This showed what I needed to design and how the pages would all connect together. This task was done by going

through system tasks. For example, a user searching for a plasma television image, firstly they would need to login, so a login page, leading to a user home page where they could perform their simple search. The home page where the search was done would need to link to a search results page, that linked to either a request page or download page. This tells me already five pages required of the system and how they connect to each other. Doing this using many different examples led me to the completion of the initial version of the interface navigation diagram. I say initial as the current diagram doesn't look like the first version I created.

From this diagram I began to add the controls required on each page, what textboxes would be required to be filled out, what buttons needed to be pushed etc. These came from the same way as before, going through tasks and imagining what the client would need to fill in. This diagram led to the creation of the database design.

As well as designing for owners and users, the system needed a separate interface system for the administration too. This interface wouldn't connect to the customer interface at all, but it is still an integral part of the system. As no customers will be viewing the interface, decoration is no as much a requirement as the user and owner interfaces, but it has to be useable and good enough to an extent that the administration don't mind making updates if they need to. If the administration interface were so bad that they are putting off making critical updates to accounts because of the interface then this would be a large problem. I am not creating the graphic designs for the interface, but I needed to develop the information to be displayed in a way that was easy for the designer to use and understand. I came up with a 6-section diagram showing users, controls, descriptions, actions, page links and a sketch/example of each interface. An example is below.

OWNER REGISTRATION			
Actors	Tasks		
 Person	Register as owner		
Description	Controls		
Contains the form that needs to be filled out to register as an owner. Some of the form is optional, some required. The required parts will be highlighted. When the client selects a plan from the combobox, a description of a plan will appear next to it. Also have the option to cancel the registration process and return to the home page. If clicking submit, and errors are found, either missing inputs or bad inputs, then the controls containing the errors will be highlighted.	Account Plan Email Password Re-enter Password First name Last name Title Profession Company name Company description Website Street address City Country Postcode Contact Number Register Home	combobox textbox textbox (masked *) textbox (masked *) textbox textbox radiobuttons textbox textbox textarea textbox (optional) textbox textbox combobox textbox (optional) textbox button button	
Links	Sketch		
Owner Page			

Fig. 5. Interface screen diagram for owner registration

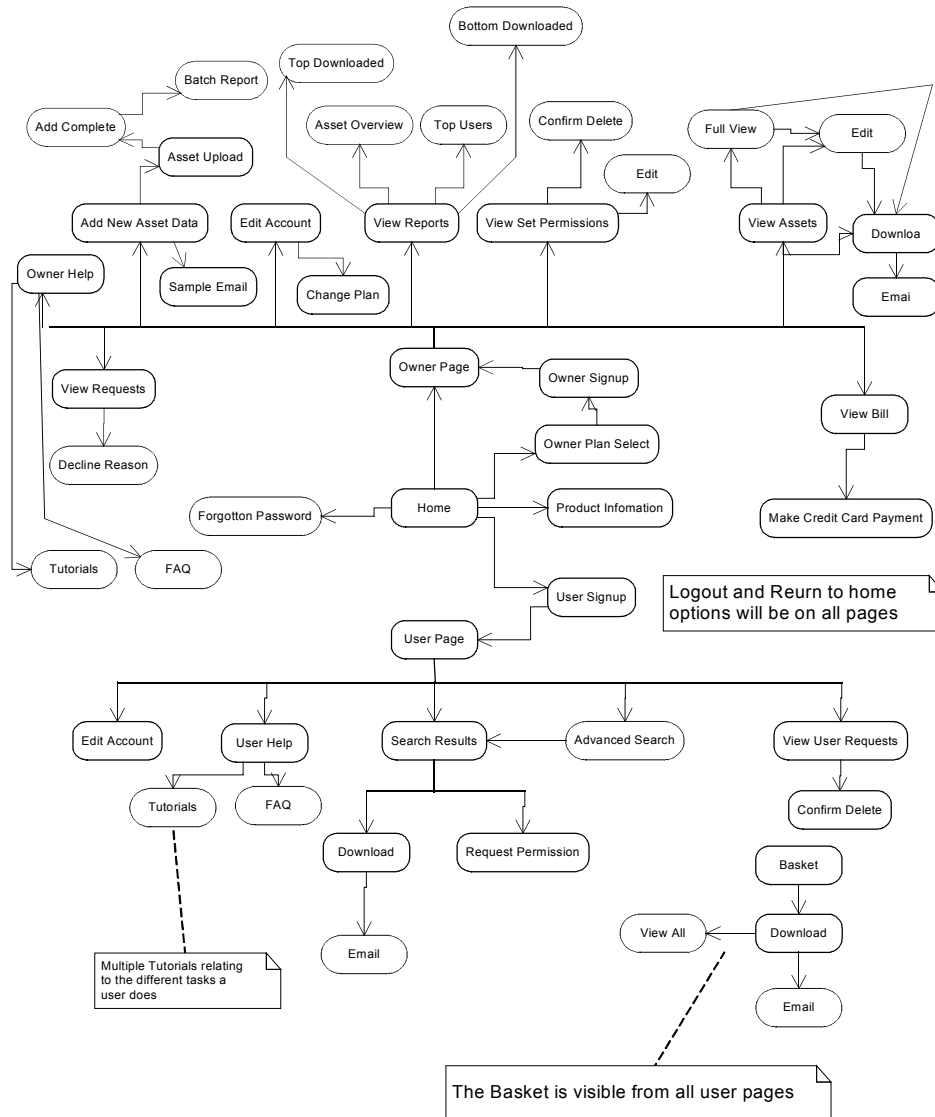


Fig. 6. Interface Navigation diagram for the users and owners

5.2 Database

The database needed to contain tables that would store all information required by the system. It is not easy to work out exactly what is required by the system and the database has been refined many times but using the user interface design makes the task a little simpler.

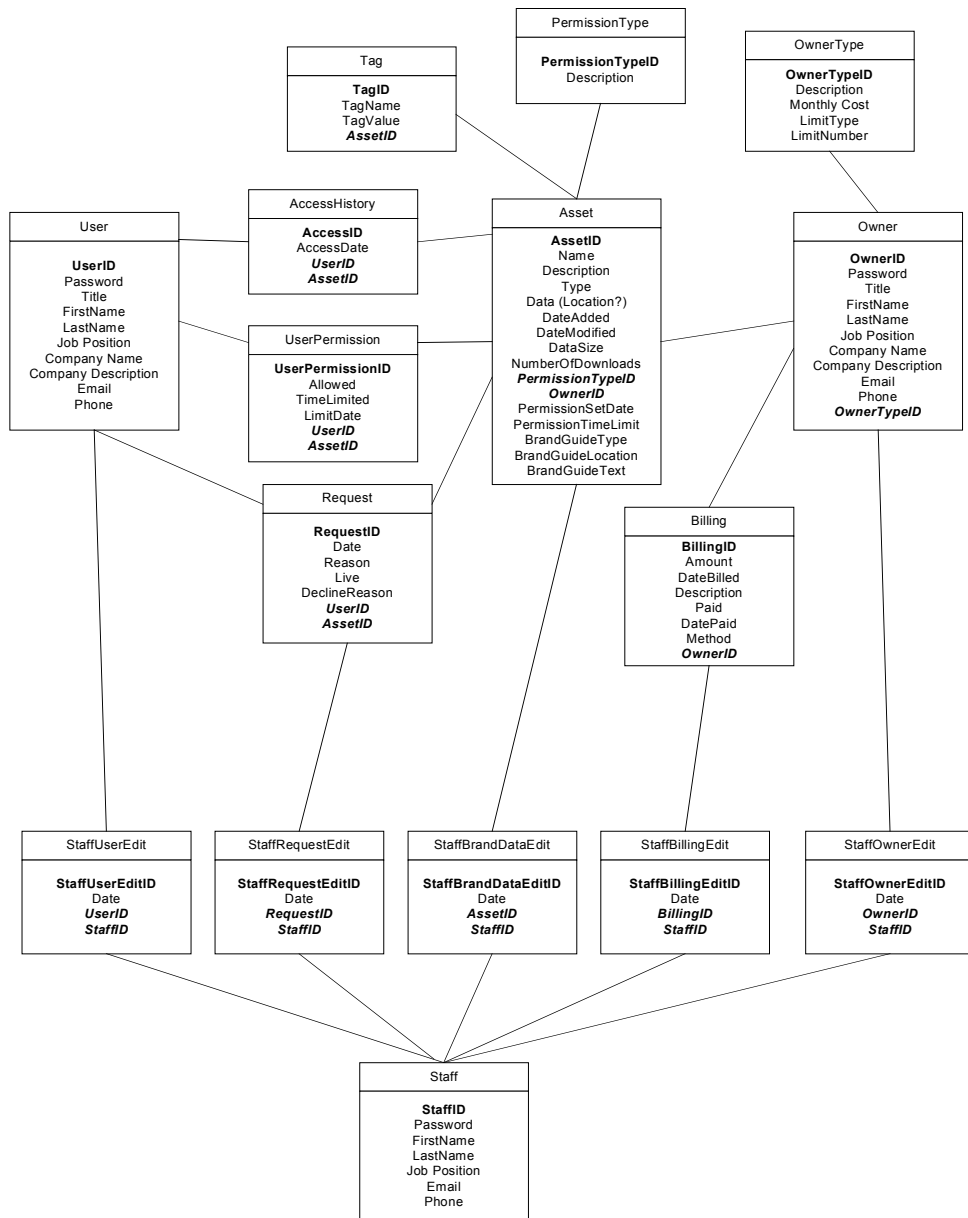


Fig. 7. Database diagram for the system

The database tables are quite self explanatory except for the staff history tables, I have decided that any edit made by a administrator should be recorded just incase a customer wants to follow up on an edit made, having the edit history table means it can be easily found who made the edit.

5.3 Business Layer

At the moment the business layer design just consists of what tasks are required of the system when the interface controls are activated, it is still at a high level as I haven't fully researched into how I will be implementing the system yet. Current design level is like this,

5.4 Storyboarding

I developed some storyboards to help with the interface design, as in previous work I have found that they are the best way of showing the flow of the interface and easily able to identify problems with it. I developed storyboards for the two main functions of the system, owners adding assets and users downloading.

5.4.1 Owner Adding New Assets

digitalstream Brand Asset Management System
THE FUTURE OF COMMUNICATION

Username: Password: Login

What is a brand asset management system?

A brand asset management system stores logos, brand guidelines, product images, design templates, PDF files, photographs, movies and music in any digital format.

These digital assets can be made available to any user that the owners allow.

The system stores digital assets in an easily searchable and secure online environment. Customer-defined categories and keywords ensure a logical storage system for future reference and retrieval. Search criteria can be customised to a level of complexity that the customer defines.

New user?

If you just need access to brand material
USER SIGNUP

If you want to upload your own
OWNER SIGNUP

Fig. 8. Interface page that a user would see when accessing the site, here they will put in their username and password and click login

Welcome Test,

Messages:

You have 3 pending requests:View Now

Bill payment overdue:View Now

Account Summary:

Assets:	
Locked	42
Unlocked	9
Total	51
Requests:	
Pending	3
Enabled	41
Declined	16
Billing:	
Last	23/4/07
Next	23/5/07

Fig. 9. The main page for owners, displaying new messages, and a summary of their account, here the owner clicked *add new asset* from the menu

Add New Assets

Add via Email:

To add an asset using email.....

View a sample email

Add via http:

To add an asset using http....

Add via HTTP

Fig. 10. First page display when adding assets, contains information on how to add using email and http, here the owner selected http

Add Using HTTP:

Asset Location: [Browse...](#)
 Asset Name:
 Asset Description:

Category: - [help](#)
 New Category: Previous Category: ☒

Permissions: - [help](#)
 Lock Asset: yes ☐ no ☐
 Time lock: yes ☐ no ☐
 Set time lock: Available for years months days hours

Custom Tags: - [help](#)
 Tag Name: Tag Value:
 Tag Name: Tag Value:
 Tag Name: Tag Value:

Brand Guidelines: - [help](#)
 Upload:
 Text:

[More Custom Tags](#)

[Add Asset!](#)

Fig. 11. The owner will be required to fill out a form describing the asset

Add Using HTTP:

Asset Location: [Browse...](#)
 Asset Name:
 Asset Description:

Category: - [help](#)
 New Category: Previous Category: ☒

Permissions: - [help](#)
 Lock Asset: yes ☐ no ☐
 Time lock: yes ☐ no ☐
 Set time lock: Available for years months days hours

Custom Tags: - [help](#)
 Tag Name: Tag Value:
 Tag Name: Tag Value:
 Tag Name: Tag Value:

Brand Guidelines: - [help](#)
 Upload:
 Text:

[More Custom Tags](#)

[Add Asset!](#)

Your asset has been successfully added.

To add assets with similar tags and descriptions,

[Add Similar](#)

To add entirely new assets,

[Add New](#)

If you have finished adding assets,

[Add Complete](#) [Add Complete](#)
 (with upload summary)

Fig. 12. After clicking *Add Asset!* The user has the option to finish adding or add more

Upload Report

Total Asset uploaded : 2

Asset Name: My House
Asset Description: My very own house.

Asset is Locked for general users.



Asset Name: A shopping cart
Asset Description: Not my very own shopping cart.

Asset is Available for general users.



Fig. 13. Here the owner has added 2 assets and chosen to view a upload report after completing their additions

5.4.2 User Downloading Assets

digitalstream Brand Asset Management System
THE FUTURE OF COMMUNICATION

Username: Password: Login

What is a brand asset management system?

A brand asset management system stores logos, brand guidelines, product images, design templates, PDF files, photographs, movies and music in any digital format.

These digital assets can be made available to any user that the owners allow.

The system stores digital assets in an easily searchable and secure online environment. Customer-defined categories and keywords ensure a logical storage system for future reference and retrieval. Search criteria can be customised to a level of complexity that the customer defines.

New user?

If you just need access to brand material

USER SIGNUP

If you want to upload your own

OWNER SIGNUP

Fig. 14. Main page, again the user will put in their details except this time the system will recognize a user and take them to the users page

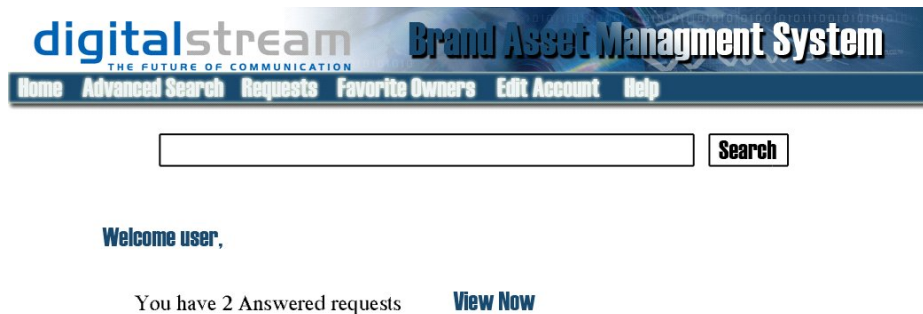


Fig. 15. Main users page, here the user is shown new messages and given a simple search option

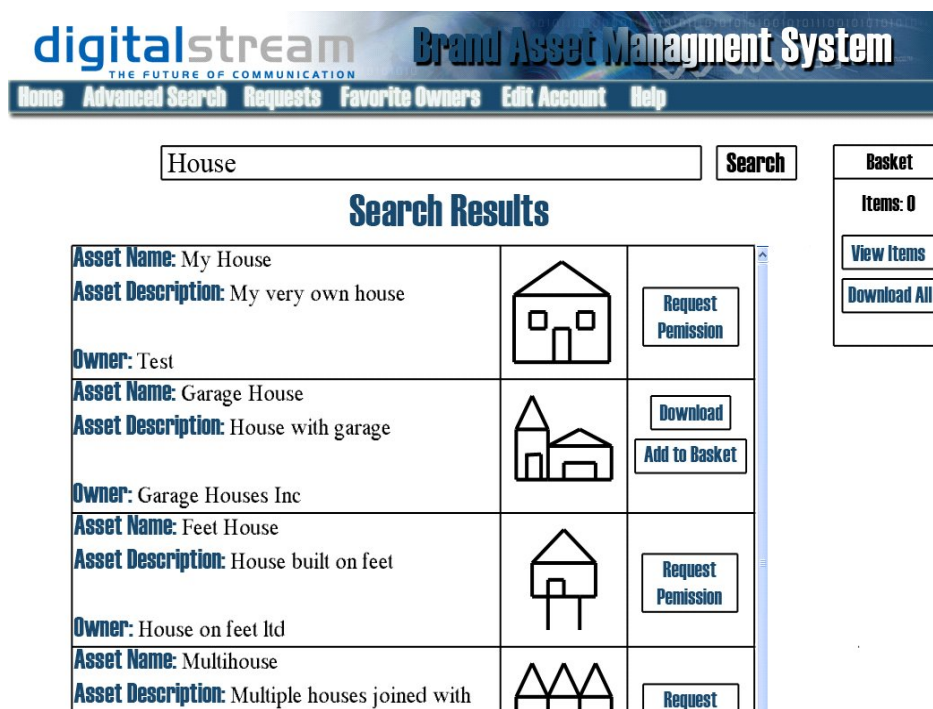


Fig. 16. Search results, this shows a summary of each of the results the search gave, here the user can download garage house, or request permission of the others

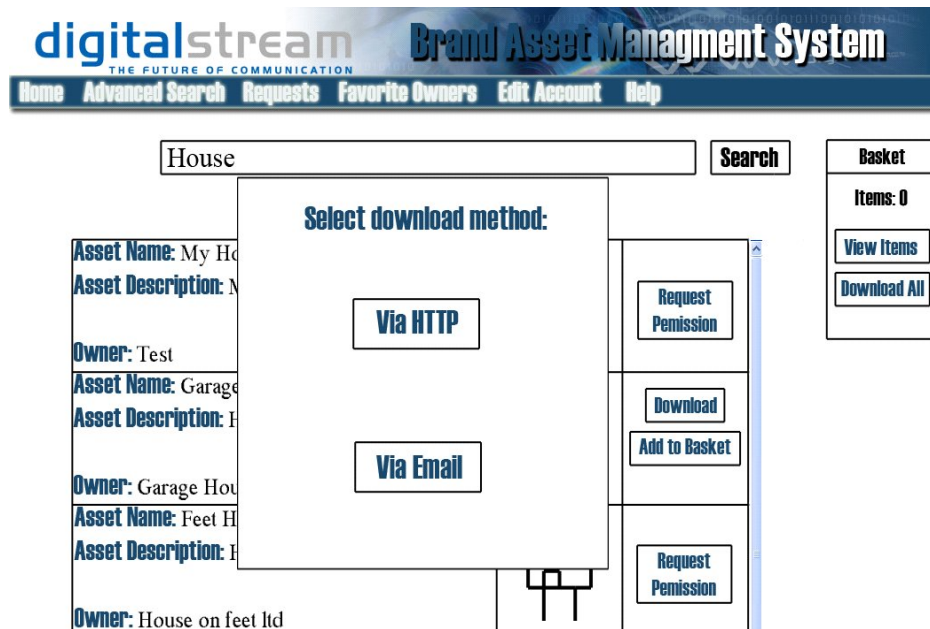


Fig. 17. If the user selects to download, then they have the option to download via HTTP or Email

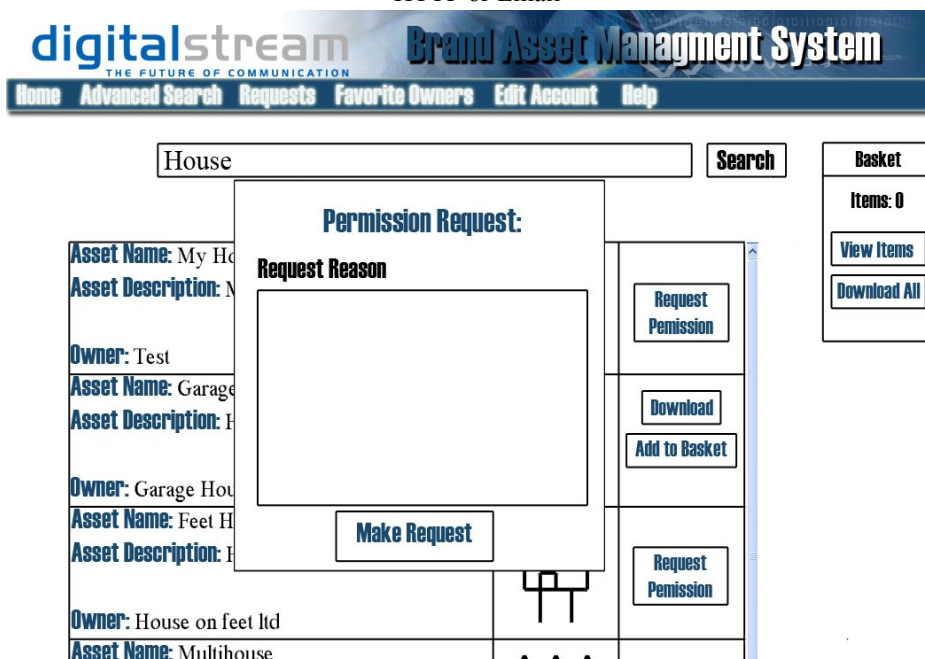


Fig. 18. If they make a request then they need to give a request reason that can be sent to the owner of the image

6 Future Work

The design needs to be completed up to a point where I can begin implementing the system; this involves researching into some new technologies and applying this to the project. I also need to research into web security, as this system will allow credit card payments onsite, security is a big factor. After the design is done, implementation will be followed by intensive testing and debugging. There will need to be stress testing done on the system to make sure it can handle multiple users. I would also like to do some user evaluation where some potential users of the system can test it out, and give their feedback on the interface and design.

References

1. E-See Company, www.e-see.com
2. OriginalImage, www.originalimage.com