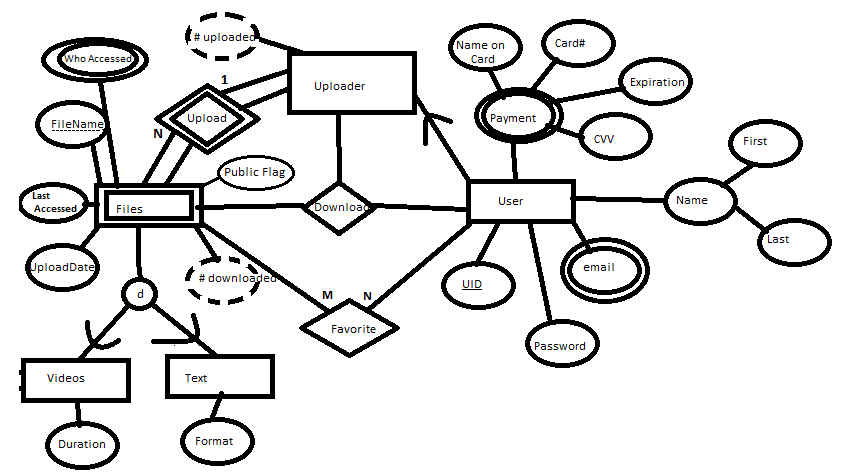
Brenden Barlow

Will Thurman

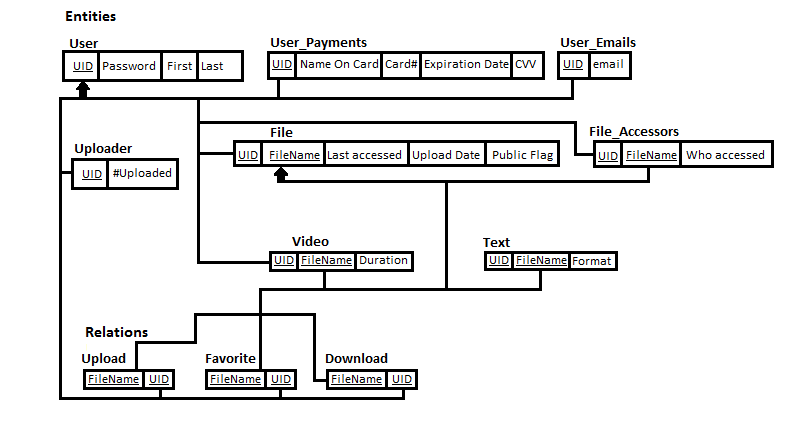
CS2300 Project Phase 2

**Problem Statement**

For those looking to organize and quickly distribute files amongst others, an online file storage system is a great solution. Allowing other people to view and edit your files has many applications, in business and personal affairs. The database system allows for people to share files only to those they want, and organize and store the files for later use. Without a database, it would be extremely difficult to allow only the requested users access to your files, and also difficult to store more than one file.

**EER Model**

**Logical Database Design**



**Attribute types**

**User**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| UID | Variable length string | User logon identifier |
| Password | Variable length string (8-16 characters) | User password |
| Email | Variable length string (list) | List of emails used to contact user |
| First | Variable length string | User’s first name |
| Last | Variable length string | User’s last name |
| Name on Card | Variable length string | Name on users form of payment |
| Card# | 16 characters | Number on users card |
| Expiration | Date (int) | Card expiration date |
| CVV | 3-4 characters | Verification code for card |

**Uploader (subclass of User)**

|  |  |  |
| --- | --- | --- |
| # of files uploaded | Integer | Number of files the user has uploaded |

**File**

|  |  |  |
| --- | --- | --- |
| Who Accessed | Variable length string (list) | A list of all usernames that have accessed the file |
| FileName | Variable length string | Name of the file |
| Last Accessed | Date (int) | Timestamp of the last time the file was accessed |
| Upload Date | Date (int) | When the file was uploaded |

**Video (sub class of File)**

|  |  |  |
| --- | --- | --- |
| Duration | Integer | Length of video |

**Text (sub class of File)**

|  |  |  |
| --- | --- | --- |
| Format | String | File format |

**Application Design**

Function 1: Search Files

This function will search for files using search text. It accesses the “File” table

Input: File name or Tag

Steps:

1. For each word input
   1. For each FileName in the File table that is Public
      1. Check if the input word is a substring of the FileName or is a keyword
         1. If so, retrieve FileName from the File table and UID from Uploader table
2. Display all FileNames with UID

Function 2: Sort Files

This function will sort files for the user. It does not access a table.

Input: Attribute to Sort

Steps:

1. If the attribute is a string (such as FileName) sort by number first, character second, from lowest number to highest, and a to z, uppercase before lowercase.
2. If the attribute is purely numbers (such as the number of files uploaded), sort from lowest to highest.
3. If sorting makes no changes (resulting list == original list), reverse the result.

Function 3: Share Files

This function will give other users the ability to see the file, or generate a direct link to it. It accesses the “File” table and the “User” Table

Input: Public or not, or request for generated URL

Steps:

1. If requesting a generated URL,
   1. Get FileName and UID of who uploaded the file
   2. Combine and convert to ASCII values
   3. Return as the generated URL
2. Otherwise, set Public value for the file to the input value
3. Get the current Public value and update webpage (regardless of if Public Value changed)

Function 4: Download/View Files

This function will download a file. If the file is public, the UID and FileName are required. If the file is not public, the ASCII value of UID + FileName will be required. It accesses the “File” table. Input should be proceeded by a 1 or a 0 (1 if the file is public, 0 if not)

Input: 0/1 + UID + 000 + FileName

1. Remove the first character from the input and store its value
2. If the stored value is 0
   1. Attempt to convert each set of 3 numbers into a character
      1. If any part fails, reject the request
      2. If 000 encountered, what has been converted so far is UID. Store the rest as FileName
3. If the stored value is 0, search for 000
   1. Anything before 000 is UID, anything after is FileName
4. Search for the FileName and UID in the File table
5. Gather all information about FileName, and present it to the user

Function 5: Upload/add Files

This function will allow a user to upload a file. It will access the “File” table.

Input: File Name, UID, inputPublic, Date

1. For each FileName in the File table that has the same UID
   1. If File Name == FileName
      1. Reject upload with this name
2. Insert tuple into File table with UID = UID, FileName = File Name, Public = inputPublic, Last Accessed = Upload Date = Date
3. Store file as UID + 000 + FileName

Function 6. Delete File – removes files from the database

Input – filename, UID

Steps:

If(UID is owner of file)

{

If(filetype == video)

Delete all records containing filename and UID from Video and File Tables

else if(filetype == text)

Delete all records containing filename and UID from Text and File Tables

Else

Delete all records containing filename and UID from File Table

}

Function 7. View Ownership – allows another user to see the ownership of a file

Input – filename

Steps:

1. Retrieve UID from File Table where filename == input filename
2. Display to page

Function 8. Recent Files – view files that the user owns in the order of most recently accessed(by owner)

Input - filename, UID

Steps:

1. Retrieve all files associated with UID from File Table
2. Sort files by most recent lastAccessed
3. Display filename, and lastAccessed date

Function 9. Favorite – adds a file to a user’s “favorite” list

Input – filename, UID

Steps:

1. Add UID and filename to the Favorite relation
2. Retrieve all records where UID = current UID from Favorite relation
3. Display to screen

Function 10. View File Activity – See all of the users who recently accessed the file

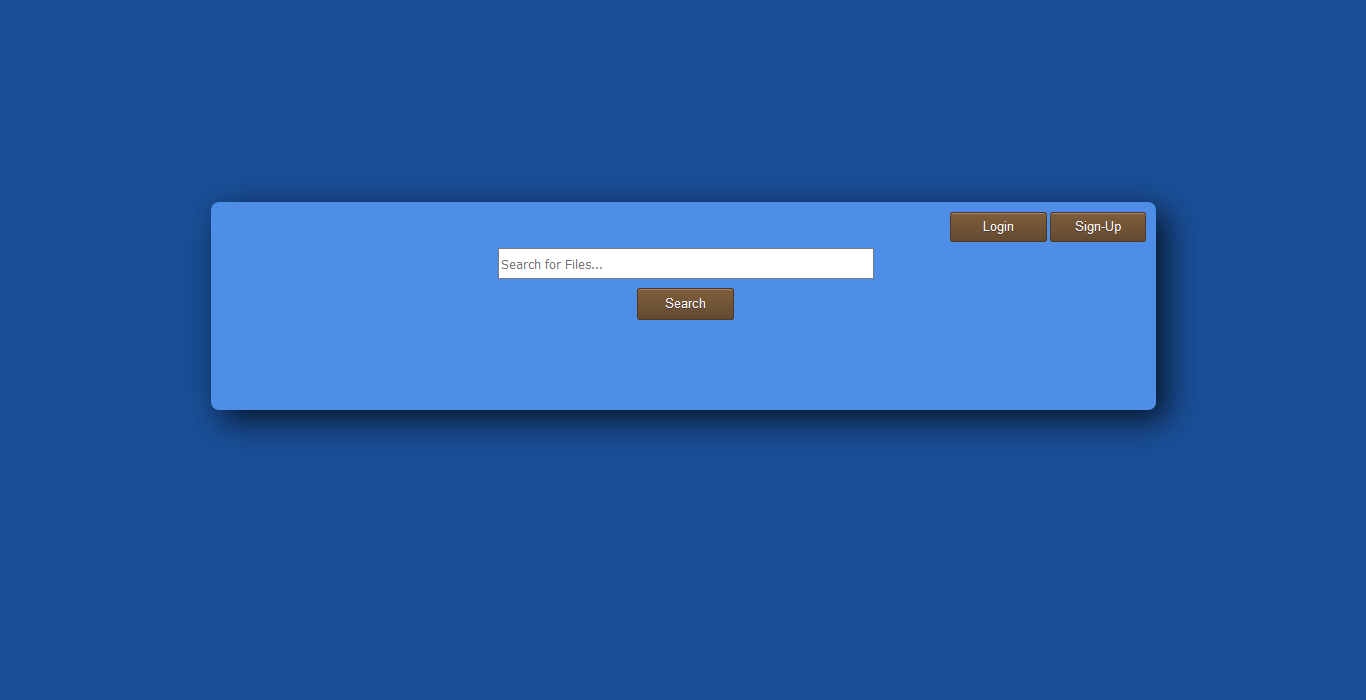
Input – filename

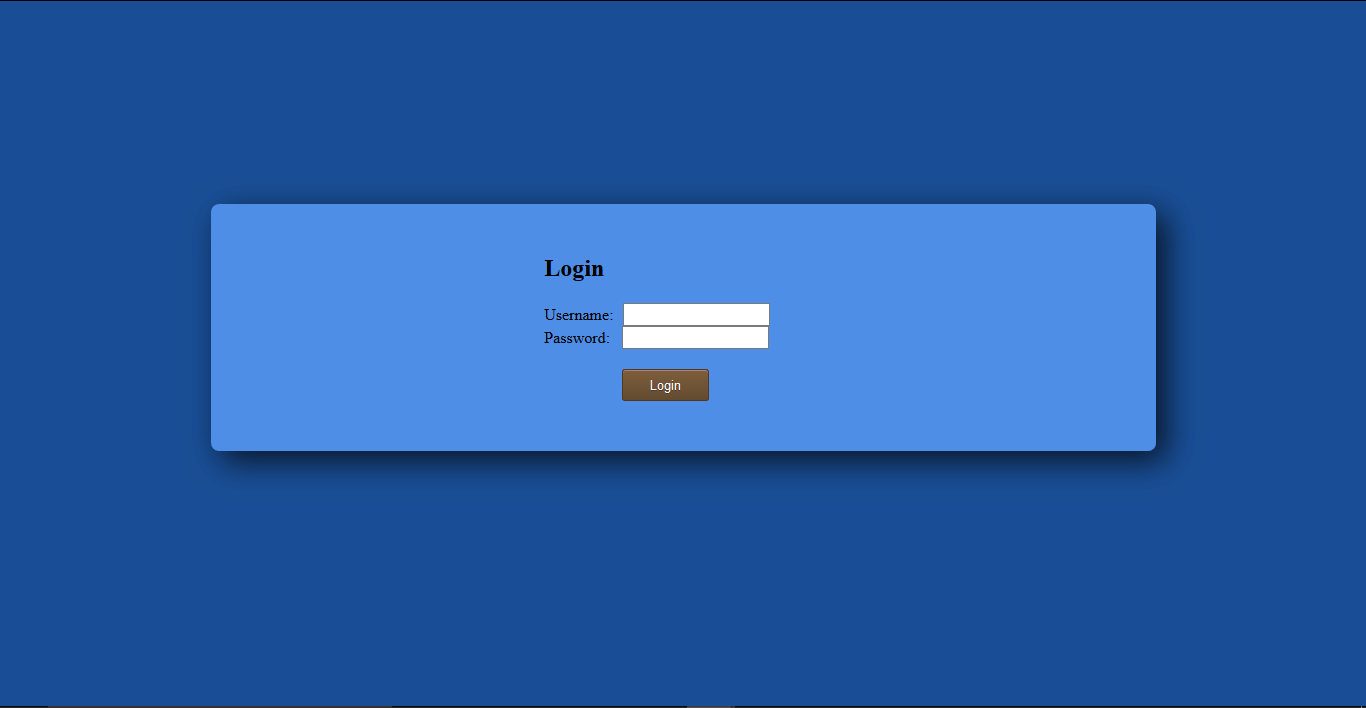
Steps:

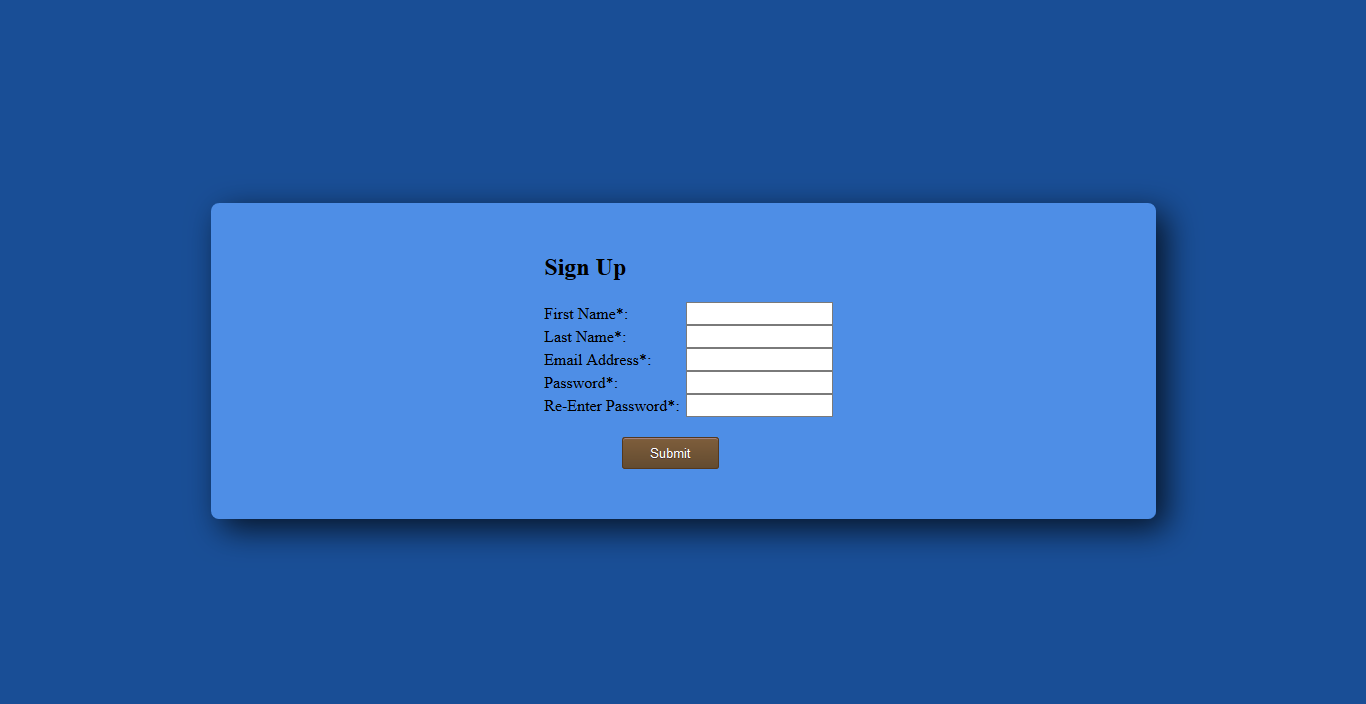
1. Retrieve the records from the File\_Accessors table where filename = input filename
2. Display all of the unique users that have accessed the file

**Screenshots**

Main Page



Login

Sign-Up

**USER MANUAL**

1. **Registration**

In order to upload files under your account, you will have to register. Registration is fairly simple, as you only need your first name, last name, user id, and a password.