Ryan Tick, James Harkins, and Kyle Kozak

Secure Upload/Download

**Purpose**

Every member of our group is currently pursuing a Cyber Security concentration within our degree of Computer Science. Because of that, we want to make a project where we can involve/expand upon our knowledge of Cyber Security.

Currently, we intend on making a website where users can download, upload, and share files securely. First, the user would have to make an account. After the account was created, the user could use their credentials to sign in to the site. From this point, the user would be able to upload text files and they would be securely encrypted on the file server. The user would have the option to share these files with other users (provided they know their user name). Additionally, they could download their files from the file server as much as they wanted.

In terms of website design, we would want the website to be very dynamic in terms of design; we want the layout of the site to look as good on a mobile phone as it does on a desktop computer.

**System Usage**

We are designing a website for our final project, so a user could access it from any web browser. We will have tested and optimized accessibility specifically for IE, Chrome and Firefox. Additionally, we want it to look good on a mobile device. Therefore, we will all test it on our phones.

End User

**Use Cases Diagram**

Delete Users

Sign In

Download  
or  
Receive Shared File

Main Page

Admin

List of User’s Files

View Delete Encrypted Files of Any Users

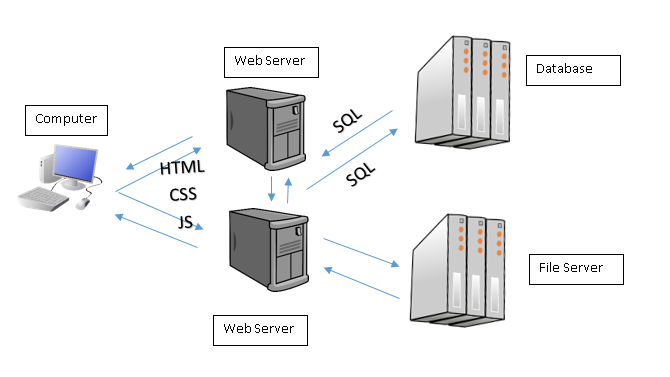
Secure File Site

Request to Upload, Download, Delete File(s)

Fulfill Request

Upload, Download, Delete Page

**System Layout Diagram**



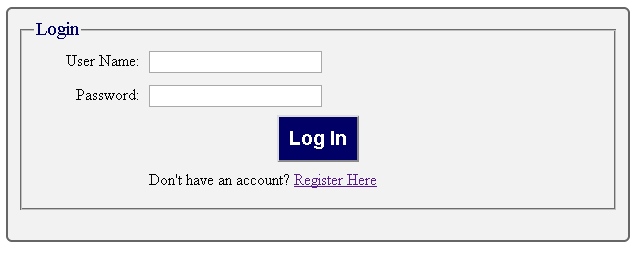
**Features**

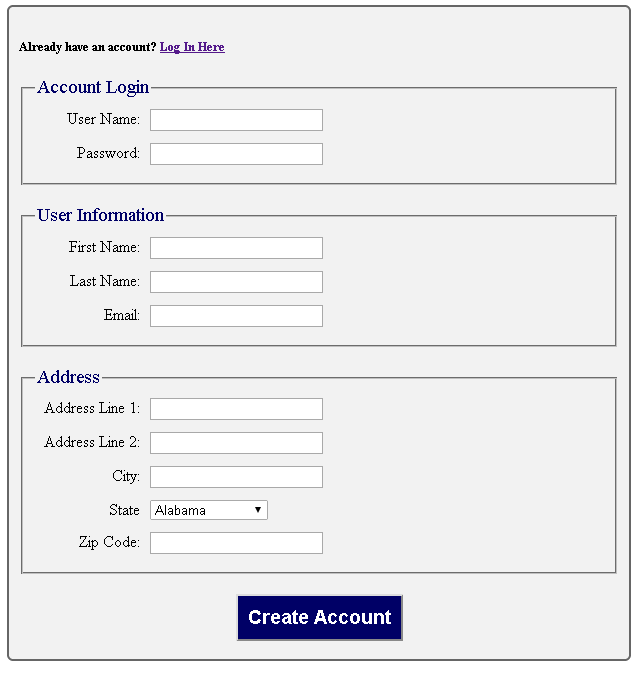
1. User Database:
   * Description: database will include first name, last name, address, email address, telephone number stored for each user.
   * Relevant Use Cases: All users must create an account and supply this information to use the sight. However, only admin users (Ryan Tick, James Harkins, and Kyle Kozak) will have the ability to see this login information that other users submit.
   * Functional Requirements: Be able to securely upload a user’s personal information to a data base.
   * Non-Functional Requirements: The account is required to access the website.
2. Login Credentials:
   * Description: Each user needs to have a secure password (at least 8 characters long with one upper case character and the use of one special character).
   * Relevant Use Cases: All users must have a username and password to access the site. However, only admin users (Ryan Tick, James Harkins, and Kyle Kozak) will have the ability to reset a password for a user. Even the admins though will not be able to see the old, or existing, passwords.
   * Functional Requirements: Be able to store a user’s username and password securely on a data base. The user needs to be able to log into their account using a password.
   * Non-Functional Requirements: The password needs to be at least 6 alphanumeric characters long.
3. File Upload/Download:
   * Description: Each user, once logged into the site should be able to upload and download files (their own files, and files shared with them). Max file size of 1 GB. Only text files are supported at this point.
   * Relevant Use Cases: All users are able to download their own files or files shared with them. No exceptions are made to this rule. Admins can see the encrypted files that other users have upload, but they will not be able to download them.
   * Functional Requirements: Be able to store a user’s uploaded files securely on a file server.
   * Non-Functional Requirements: Should be able to upload/download a text file no bigger than 1 GB.
4. Sharing of Files:
   * Description: Users should be able to select which users can download their files by sharing files with them.
   * Relevant Use Cases: All users can only share their own files. They cannot share files that are shared with them.
   * Functional Requirements: Be able to set different permission levels on the file, depending on with whom the file is shared.
   * Non-Functional Requirements: Need to know the username of the user you want to share the file with in order to share the file.

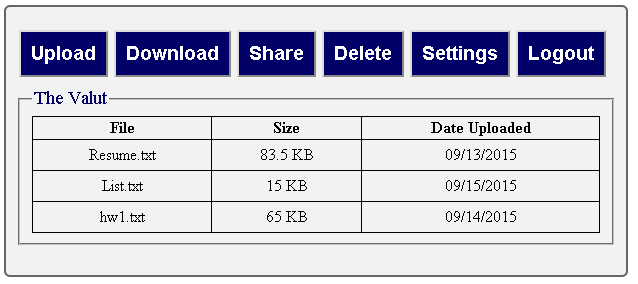
**Dependencies**

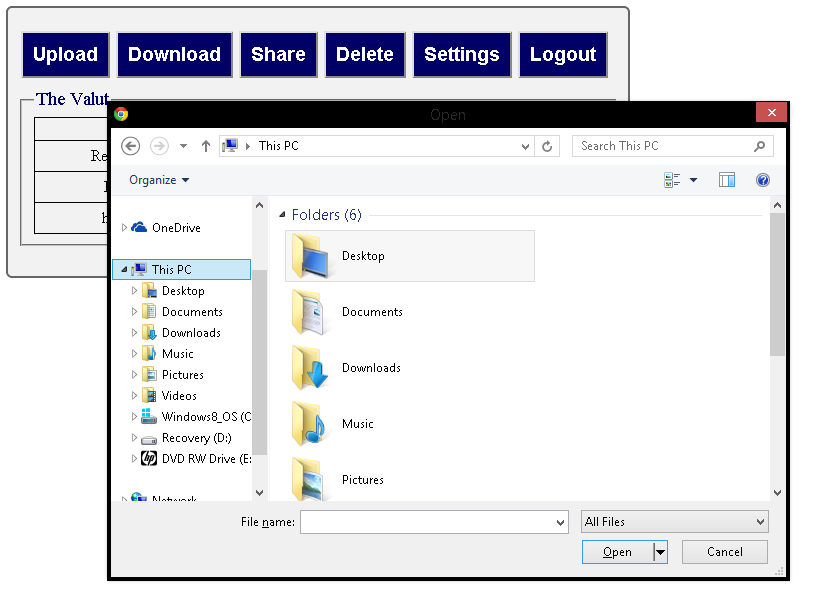
We are limited by hosting our website on AFS. Therefore, it might be hard to link to a SQL database to manage users. Additionally, we might run into memory problems since we all only have 2GB of space on AFS.

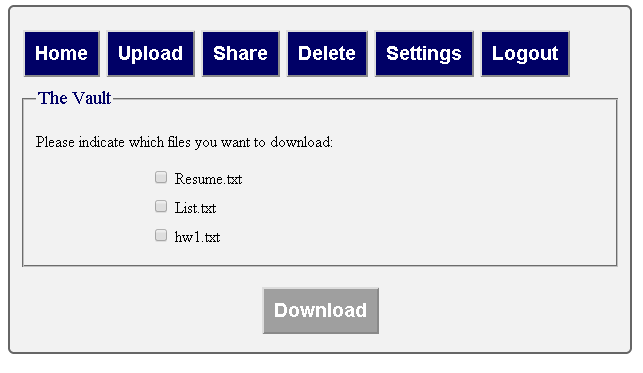
**UI Mock Up**

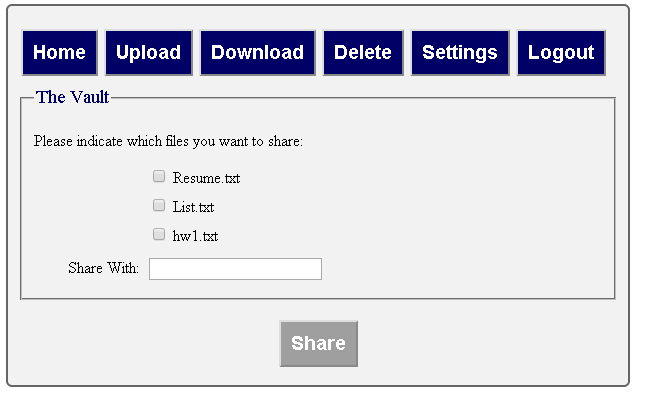
Login Page:****

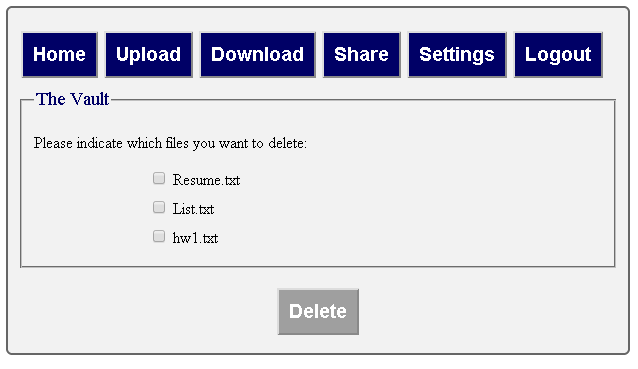
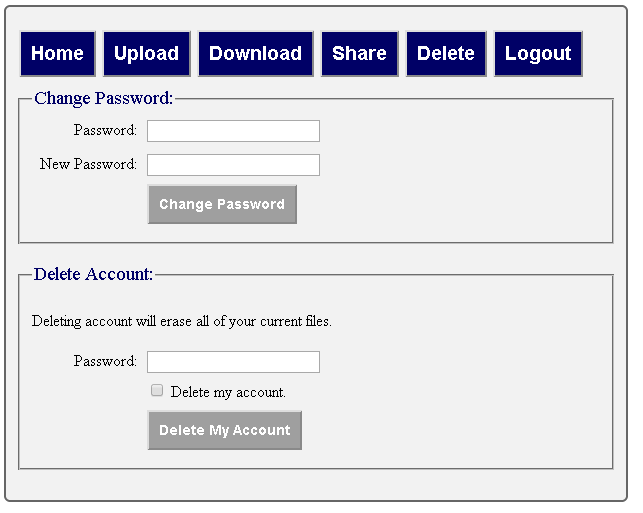
Registration Page:****

Main Page:****

Upload Button:****

Download Page:****

Share Page:****

Delete Page: ****Settings Page:****