**Lemp Stack Cloud Drive**

**StuffMyFiles.com**

**Team 1**

**Abdulsalam Bdeir, Willam Rood, Kevin Tran, Jae Park**

**Software Requirements Specification Document**

**Version: (0.0)** **Date: (03/6/2021)**

Contents

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms, and Abbreviations. 5

1.4 References 5

1.5 Overview 6

2. The Overall Description 6

2.1 Product Perspective 6

2.1.1 System Interfaces 6

2.1.2 Interfaces 7

2.1.3 Hardware Interfaces 7

2.1.4 Software Interfaces 7

2.1.5 Communications Interfaces 8

2.1.6 Memory Constraints 8

2.1.7 Operations 8

2.1.8 Site Adaptation Requirements 9

2.2 Product Functions 9

2.3 User Characteristics 10

2.4 Constraints 10

2.5 Assumptions and Dependencies 10

2.6 Apportioning of Requirements. 10

3. Specific Requirements 11

3.1 External Interfaces 12

3.2 Functions 12

3.3 Performance Requirements 13

3.4 Logical Database Requirements 13

3.5 Design Constraints 14

3.5.1 Standards Compliance 14

3.6 Software System Attributes 14

3.6.1 Reliability 15

3.6.2 Availability 15

3.6.3 Security 15

3.6.4 Maintainability 15

3.6.5 Portability 15

**1. Introduction**

The cloud drive will be a file hosting service that allows account holders to save files on a cloud and access them from any computer with an internet connection. Users will be able to make accounts that allow them to sign in, sign out, add friends, set privacy permissions to files, and share files with other users. The following are what the cloud drive must be able to do:store/alter/download/deletes files — Move, copy, delete, rename, and caption your photos and files after you upload them. Organize files—Arrange the user’s files in top-level folders and subfolders the user creates. Sets File Permissions — Choose permissions for each top-level folder that the user creates. Keep the photos, files, and favorites in personal folders so the user can keep them private; in shared folders so the user can share them with people on his/her contact list; or in public folders so that they can be viewed by anyone on this Cloud network. The Cloud Drive storage meter — shows how much storage space the user has used + their total capacity.

Purpose: the web application will be used to store, share, and retrieve files. Offload user data from a local computer to an external host machine to save space on a local machine and be able to access those files from any machine connected to the network/internet. It will be able to save user data in a secure source and retrieve data from any host and hide or share specific files with anyone on the network.

**1.1 Purpose**

Intended use - It should be used for storing files that need to be shared among multiple machines between one or more people. Files may also be kept private in the case an entity wishes to keep sensitive material secure from unauthorized access yet still wishes to have said material easily accessible to themselves.

The purpose of this cloud drive is to allow people to access an online file manager where they have most of the options that they would normally have on their personal local file manager. The intended audience is anyone who wants to have an extra spot to share and save files. The point of this cloud drive is to be easily accessible allowing users to do all of the functions listed in the introduction.

Intended Audience - All users with, at least a novice understanding for file systems and cloud-based websites such as dropbox, google drive, or microsoft onedrive.

**1.2 Scope**

The web application will help facilitate quick, easy, and secure means of offloading files from one host machine to a cloud system that can be accessed by any user with an online account and a machine capable of connecting to the internet. The benefits of a project such as this is the offloading of data from local machines, the secure and private way of storing/organizing documents and the ability to specify varying levels of viewing privileges to other dignitaries allotted as the user has permission to do so.

The site will create user accounts for cloud storage, but will not create any accounts if the storage capacity cannot allocate space for a new user. In addition, the program will allow users to create an account and sign into existing accounts. Each account made will have a folder that corresponds to that account and that folder can only be accessed by the account that is linked to the folder.

This application will give users a certain amount of storage to manage any files they insert into said folder. Users will be able to share files inside their respective folder with other users that have registered and made an account for the server. The main benefit is that it gives customers extra storage that they can access from anywhere as long as they have a device that has a connection to the internet and are able to connect to the server. The main objective of this application is to give users the ability to either create a network of shared files or just to have an online storage that is easily accessible.

**1.3 Definitions, Acronyms, and Abbreviations**

**Software project** - Refers to the cloud drive.

**Cloud-storage** - Uses computer servers to physically store data while making the data available online to users via the web.

**User** - The person, or persons, who operate or interact directly with the product. Any user that may create an account to store, share, and view files that are uploaded through their account.

**Primary user** - Any user that shares one or more files or folders that are uploaded through their account.

**Secondary user** - Any primary user that is given a level of access to a file/folder that was not uploaded by that user.

**System administrator-**  Any user that has permissions to view and edit aspects of the server, database, web interface, and any of the created accounts on the application.

**Developer-** The persons (or persons) who are developing code and UI elements to create and get the application working.

**Risks** - Deletion of files are permanent and not retrievable by all other users who had previous access to said file(s). There is no offline support to access/store files when network connection is not made. Another risk is that hardware (for example, hard disk failure) always has the possibility to fail, which can lead to data loss or other hardware related issues.

**Host machine-** The main computing server that hosts the website, processes functions, and stores the physical storage devices where all the cloud data is kept.

**1.4 References**

LEMP Stack:[How To Install Linux, Nginx, MySQL, PHP (LEMP stack) on ...www.digitalocean.com › community › tutorials › how-t…](https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-on-ubuntu-20-04)

Initial Server Setup: <https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-20-04>

**1.5 Overview**

In this SRS document, one can expect to see everything they need to know about the software project in development. As far as format goes, there are different sections that might meet one’s interest. Section 1 explains the details of this document. Section 2 describes how the product works, while still remaining somewhat general. The third section talks more about the specifics for the product. This includes the specific requirements, the performance requirements, the functions of the product, the external interfaces, the design constraints, the database requirements, and the system attributes.

**2. The Overall Description**

User needs - System admins will need to monitor account creation (maybe verification) to see how many users are registered and ensure the application can provide an even distribution of storage space. System admins must approve the allocation of extra storage space for a user. System admins will need to remove users when necessary and delete “Free-floating files” files that no longer have any users attributed to them. System admins must also monitor system backups of files and database tables in case of drive or application failure.

**2.1 Product Perspective**

The business case and operational concept of the system: a business that requires a variety of file types to be shared and retrieved through multiple machines at any given time. A business that relies on storing a moderate amount of data and is looking to offload their individual machines as well as decompartmentalize their data storage.

Operational Concept - User A frees up space on their machine by offloading unused files on their machine by offloading unused files or less frequently used files on to a secure cloud server. User B uses StuffMyFiles to upload files required on multiple machines through one machine then downloads said files on the multiple machines without relying on “sneakerware”. User C uses StuffMyFiles to share various files between one or multiple users and specifies their personal limitations on accessing the files.

**2.1.1 System Interfaces**

OS: Ubuntu v20.04 All of the software used in this project will rely on Ubuntu as the operating system because of our project’s reliance on a lot of Ubuntu’s default package repositories.

**2.1.2 Interfaces**

External Interfaces-

For the login interface, there will be an option to log in, which will consist of 2 text boxes requesting a username and a password that corresponds to said username. Another option listed in this UI is the option to sign up, and this will be in the form of a button that users will click if they choose to create an account for the server. This then will bring up a new interface with 2 more text fields, requesting the user to create a username and a password.

For the main interface, the biggest element will be the contents inside the folder designated to the account that is logged in at the time being. There will be a side button that will bring out a bar that will show all the files made public to everyone by other users also in the application. On the bottom of the screen, there will be a bar that will show the user how much of the allotted storage has been used up, and a button to add files to their folder.

System Admins, with proper authorization and credentials, may connect to the host machine remotely from other machines in order to manage the web server and user accounts using 3rd party terminal emulators.

The interfaces will be optimized in such a way that will not require users to have any knowledge of coding in order to use this application. Model-view-controller models will be implemented in order to keep the interfaces as simple as possible.

**2.1.3 Hardware Interfaces**

SSDs are the preferred storage devices for their fast reads/writes and allocation methods and durability/longevity. However, if cost is a factor, harddisks may be used in conjunction or by themselves but system speeds may decrease. Additionally, power usage and the risk of data loss may increase due to the mechanical nature of hard disks. If cost is not an issue, it is recommended the motherboard offer 1 or more PCI interfaces and storage devices should take advantage of those PCI slots considering the nature of what the system is expected to accomplish. M.2 PCIe cards are notably faster than SSDs that use SATA connections even though both utilize PCIe bus and the NVME protocol.

The host machine should include a motherboard that can accommodate a processor capable of multithreading to support multiple processes to be able to run at the same time.

A host machine will be expected to receive, process, and return a large volume of requests, therefore the host machine should include an Ethernet card and I/O port to receive and send large amounts of data. It is not recommended for the host machine to use a wifi card, as it is not required for the machine to be portable and the decrease in performance will be substantial.

The host machine should be capable of storing a large amount of system memory. As such, engineers should be capable of increasing or decreasing the allocated memory size of the machine during maintenance periods as they see fit to handle the needs of their user base.

The host machine may include a physical monitor and keyboard but it is only required during initial setup and potentially during physical maintenance of the server.

**2.1.4 Software Interfaces**

**Database management system:** MySQL v8.0.23 This is the database/SQL server that the interface will communicate with in regards to certain user information. The information in the database will not be seen by the user, but will be accessible by developers/admins only. As of now, the database will hold passwords, usernames, access abilities with shared files/foreigh key pointers to separate users, plus more if needed. When a user creates an account or grants access to other users, they will be interacting with MySQL.

**Server:** Nginx v1.18.0 In order to serve our web pages to site visitors, we are going to employ Nginx, a modern, efficient web server. This software is a high performance server that is the host of the online interface when users need to access the website, Ngnix has many capabilities and can act as an HTTP and reverse proxy server, a m mail server, a standard TCP/UDP proxy server, and more. In the scope of this software project, it will be used as a TCP/UDP proxy server and when users are accessing certain web pages associated with the cloud drive, they will be directly communicating with this server.

**Server API:** PHP-FPM (PHP v5.3.3) will be used to generate the dynamic content of the site. PHP-FPM is useful for heavy loaded sites therefore will be implemented in our project to anticipate site load in the long term.

**2.1.5 Communications Interfaces**

The host machine will use SSH and SFTP protocols for uploading/sharing files on the site and HTTP, HTTPS, and TCP to allow users to interact with the website. The host machine will also use SSH and Telnet protocols when System Admins and developers need to access the host machine’s command-line interface remotely to manage the web server, database, or storage devices.

Refer to the Active IETF working groups for RFCs. Current RFC Link:<https://tools.ietf.org/html/rfc7100>

**2.1.6 Memory Constraints**

Regarding memory constraints, there are a few aspects to worry about. One, for example, is that the user file allocation is dependent on the total storage. The server load throttles when too many users request files all at once. The server used in this case is a repurposed Macbook pro late 2011 model running Ubuntu on a moderately new SSD. The actual server associated with the cloud drive has a memory constraint of 1G of RAM, while user machines must have at least 256M of RAM.

**2.1.7 Operations**

* Every several days or so, there will be downtime for a short duration of time just to keep the performance of the server up to par. This would mean that users that are trying to access this application during these times will be unable to use the application for the time being.
* A database will be used to keep track of all the usernames, passwords, sharing permissions, and number of days inactive for each user. This makes it so requesting data and accessing the data would be much faster than any other alternatives, making the experience a lot smoother for users on the cloud drive.
* Our methods of backup and recovery would be if time is allowed, is installing 2 hard drives of the same size and copying over from the original to the copy every night, therefore the user is able to recover files when he makes a mistake before the day is gone.

**2.1.8 Site Adaptation Requirements**

Because of the nature of the cloud drive, the user will not be required to install any software. Instead, they will be accessing a website to use the cloud drive. Once they first connect to the site, however, there is an initial setup process. The first thing the user must do is create an account that they will use to log in whenever they need to access the drive. The site is intended to be used on any computer browser and will support that. The drive will not be available in any other form, such as browsers on mobile devices or apps. To put simply, all the user needs to connect is a computer with a browser of their choice and an internet connection.

**2.2 Product Functions**

The final product will have these functions:

1. Create user account; establish username, password, and generate a main page for the user
2. Be able to upload and download files, photos, and folders from the cloud drive
3. Be able to view and modify their uploaded files. Modification will be rename, delete, and move the file from one folder to the other
4. The product will display a visual meter that shows the individual user’s storage space used and how much free space they have left.
5. Be able to view other users available to share files with
6. Be able to share files, photos, and folders with other users
7. Be able to set permissions for individual files or folders containing multiple files such permissions will include: Modifiable/not modifiable and downloadable/not downloadable for each user the file or folder is shared with.
8. Unregistered users can make their own account and have their own storage space.
9. Registered users will have the option to renew their passwords in the case they forget or need to update their passwords

**2.3 User Characteristics**

The system will not allow more than a certain amount of accounts to be made and registered due to limited storage being set aside for the server.

During account creation the system must check first that the system contains enough available space on the drive for the account to be created. If there is, account creation can proceed, if not a message to the user should be displayed stating they can not create an account and account creation should not be allowed to continue.

This system will not allow users to exceed the storage limit allotted to each of the accounts under their possession, this does not limit the amount of files that can be shared with them, however.

The system will not stop users from making multiple accounts, which can turn into a loophole in order to have access to more storage.

The user’s download and upload speeds will be limited to the server’s download and upload rates.

The server can only handle so many accounts being on at the same time at once, since the amount of RAM allocated to the server will not be very large.

The server can only be active when the host computer is on and is actively running the server. The host machine should be capable of being kept running for long periods of time without crashing and should have a form of battery backup incase of power outage.

The server will have no automated way of restarting when an error occurs and stops the server, meaning that someone will have to manually restart the server and users will not be able to connect to the server until then. The webpage should display an error message at the login page to inform the user of the issue.

In order to apply updates, the server will have to temporarily shut down in order to implement new code and features and then start back up.

**2.5 Assumptions and Dependencies**

If the user wanted to make multiple accounts, the software would need to be changed accordingly to make sure to limit use to one per person. This way, everyone has fair use of the cloud drive and no one can manipulate operations.

If the user was somehow confused as to how the interface works, instructions and/or features to improve clarity would need to be added to the interface.

If the user tries to upload a file that exceeds the maximum storage limit that is allotted to them, the system must be able to prevent them from doing so, ideally giving them a prompt that tells them that said file exceeds the storage limit.

**2.6 Apportioning of Requirements.**

Users may want to suggest ideas as to what they want to see added to the cloud drive. These new functions that weren’t in the original plan may or may not be implemented and if they are, this will be after the release of the product. Some abilities of the cloud drive that the developers are considering are the ones listed in section 2.5. For example, tutorials, account resets, and account limits. These functions are not the main priority of the current developmental plan and may be delayed until much further on in the process.

**3. Specific Requirements**

System shall create accounts as long as there are storage spaces left for the user.

System shall upload files, photos, and folders from the file explorer in the computer as long as there is enough storage to hold it.

System shall be able to modify file rename, move, and delete files with a right click from a mouse.

System shall prompt the user to login or sign in in the front page.

System should be able to stop the creation of an account if the username is already in the database.

System should be able to reject users from logging in if they input a password that does not match with the password that corresponds with the username’s account.

System shall be able to download files from either their own cloud drive or shared files.

System should have a place that holds buttons for organization of files by names, date modified, date added.

System may have a backup drive that copies all the content from the original website every night.

**3.1 External Interfaces**

* Share Button: This button will modify permissions of the selected files/folders, assigning foreign keys to any users specified by the user trying to share the content. Additionally, the user sharing the files or folders will have radio buttons they will need to input to determine the secondary user’s file permissions. Permissions include:
  + Hidden: this is set by default, secondary users have no permission to view or access a file in any way.
  + Read: Secondary users can only view the file, no changes can be made to the file and the file cannot be downloaded.
  + read/download: Secondary users can view the file and download said file on their machine.
  + read/modify/download: Secondary users can view the file, change the file name, and download the file on their machine.

Before the user can actually share with accounts, the system will make sure that said accounts trying to be shared actually exist by requesting search queries from the database that match the accounts in question. Once the secondary users are confirmed, the database assigns the file forign keys of secondary users to allow the file to be viewed by them and then additional keys depending on the level of permission granted. Lastly the share button will now display the shared file a notification

* Sign in Button: Signin button takes two strings from the Login Username Textfield and Login Password Textfield entered by the user. If both checks by the database server return without error, the user is brought to their personalized site homepage.
  + Login Username Textfield: this will take the user input as a string inside the text field and will do a query search in the database to see if there is a username that matches whatever is in the text field. If the username is not found an error message occurs and the login is interrupted.
  + Login Password Textfield: After successfully finding a username that corresponds to the username typed in the login username text field, the system will see if the string typed in the text field by the user matches the password associated with the user account by requesting another search query to the database. If the input does not match what is stored in the username’s database an error message is displayed and login is prevented by returning an error to the sign in function.
* Registration button: this button takes two strings from the Register Username Textfield and Register Password Textfield entered by the user. If the function called returns without error the user is redirected to the site’s sign in screen.
  + Register Username Textfield: This will make a new entry to the database, first requesting a search query to see if there are any existing accounts that match the text inside the textfield. If there is a match, the system will pop up a prompt telling the user that an account with that username has already been made. If not, then a new SQL table entry with that username will be added to the system along with a surrogate primary key.
  + Register password text field: This will make a new entry in the database table that corresponds to the username typed into the register username text field ONLY IF that username has not been taken already.
* Tab button to display a side navigation menu: this button contains links to all users available to share files with (this is also where secondary users can view files/folders shared to them by primary users). Initially, the CSS has the table hidden from the user. Clicking on the tab button sets a boolean variable to its inverse to either display or hide the navigation menu from view. This alters the CSS of the table on the webpage leaving only the button in view.
* Sort Items Button: This button will display a drop down menu next with additional buttons on how to organize the files presented: A-Z alphabetical, date added, and date last accessed. The option selected by the user will be sent to the webserver to change the HTML/PHP on the site to what the user selected.

**3.2 Functions**

The system shall check if the user rightfully owns the file, folder, etc. when renaming it, or deleting it from the storage.

The system shall check an account’s shared permissions every time that account logs into the system.

The system shall not allow a user to upload a file if that file pushes the account over the max storage allotted to that specific account.

The system should show any file that is shared with the public to all accounts in the database.

The system shall check for storage space before it starts to upload to the cloud drive. If the file is larger than the storage left in the cloud drive, it shall halt the transaction.

The system shall backup to previously saved data when there is an error between upload, share, and download.

**3.3 Performance Requirements**

Accessing shared files should be less than 1 second.

The server should be able to handle more than 50 users at a time without slowing down the overall performance of the application.

Since the database will not be very large, navigating and requesting queries should almost be instantaneous.

Users should be able to upload and download files greater than 1 megabyte per second.

The system should be able to instantly respond to a login and register a request.

**3.4 Logical Database Requirements**

This software project relies quite heavily on the use of a database to store all the necessary information we need to work in concurrency with the user interface. No information within the database will be accessible by users. Information that will be stored is as follows:

* Usernames of the user accounts. These will be stored as strings between 5 and 15 characters. There will be some error handling if the username has been taken already during account setup. If the username is found in the database, the user will have to choose a different name. THere will be a foreign key in the usernames table referencing passwords to connect the account information.
* Passwords of user accounts. These will be stored as strings between 5 and 15 characters.
* Sharing capabilities between users. There will be foreing keys referencing other users and their level of permission for each file.
* There will also be information on user activity stored in the database. This isn’t a tracking method, but more of a way to make sure we have an active pool of users. If the days of inactivity crosses 60 days, the user will be purged allowing a slot for a new user.
* More to be determined

**3.5 Design Constraints**

Design constraints in regards to the cloud drive will mostly correspond with the interface that the users will see and interact with. One goal of this project is to have the program run smoothly and efficiently. In order to do this, there are a few things that must be considered. The user interface should use as little elements as possible to keep the usage of memory low. The number of users will be limited as we need to make sure the storage doesn’t exceed the 500G we have available for the drive. By keeping a total storage of this size and a pool of active users, we can keep the program running more efficiently. Another design constraint we have in place is that we want to try and keep the amount of web pages as little as possible, in fact, our goal is to have only one page that the user will see.

**3.5.1 Standards Compliance**

The only regulation that must be abided by in this software development process is keeping the documents in IEEE format. The documents included in this are the SRS (The current one), the SPMP (Software Project Management Plan), and the SDD (Software Design Description). Regarding other regulations, there are not many in place because this is a class project and we are not following a typical business’s or company’s regulations and restrictions. The overall standard of compliance is to finish the project and have it work.

3.6 Software System Attributes

**3.6.1 Reliability**

There should be a mechanism to automatically restart the server if an error occurs and crashes the server. Otherwise, the server would be down indefinitely until someone manually restarts the server.

The end device that will be hosting the server must have a stable connection to the internet, otherwise that would greatly decrease the reliability of the application, and would also require a decent connection to allow all users using the application to upload and download at speeds greater than 1 megabyte per second.

At certain time intervals, the server should shut down for a brief moment to allow developers to optimize and maintain the health of the server.

**3.6.2 Availability**

Cloud drive should be able to stop the transfer/download of a file before it has finished downloading.

Since the users access their cloud drive account at different times, the server should run 24/7 to satisfy all users that want to access their account at any point of time.

Users are able to access the website from any device that can secure a stable and steady connection to the internet as long as the server is up.

**3.6.3 Security**

For security, constantly checking with the database if the password and the username matches. If it does, the user should have access to everything that is in the folder that is controlled by the account. System should not allow users to login into accounts if they input the wrong password to the account they are trying to log into.

When a file is deleted, the system would check if the file that is deleted is deleted by the account that holds the file or has permission to delete the file.

System should not allow other users to view folders in other people’s account other than the ones that have been shared with them. The system will keep a constant check of which files are shared with whom.

**3.6.4 Maintainability**

For maintainability, it will include the following:

1. The system will keep in check of inactive users, and if the user is inactive for more than 500 days. There should be a mechanism that will keep track of the days inactive and store that into an integer associated with each account in the database.
2. The system will keep track of folders that are not accessed for a certain amount of days and will delete it to create storage space for the user. However, the user will be able to recover this folder.
3. The system will automatically log out users if they are inactive for more than 1 hours in order to keep the memory usage of the server low.

**3.6.5 Portability**

The program will be dependent on our host computer to handle everything including opening up the server and to store files the users want to store. This software can be used in computers that have access to linux, LEMP stacks, and Ubuntu since these softwares will take a big part in our cloud drive. Therefore, it will be specifically for our host computer since the software will be created around our host computer to have everything that is needed.

Around half of the code that is written for this application will be server dependent, since the server will host the website and hold all of the data that corresponds to the application. The users will request to see information either by dns requests or through other means, and the server will redirect them to the pages that they should see during the use of the application.