Deliverable 4 documentation

for

Secure Data Sharing using Blockchain Technology

Version 1.0

Prepared by **Team Sigma**

University of North Texas

*Akhila Pam (11711224)
*Akshara Reddy Bathula (11713259)
*Jyothi Anjan Manini (11715079)
*Lakshmichatura Medidi (11682526)
*Manoj Kumar Bandari (11711378)
*Nimitha Bangalore Sathyanarayana (11649788)
*Nitin Reddy Balaiahgari (11698724)
*Satya Laxman Pranav Vadlamani (11701928)
*Sumuk Reddy Kalagiri (11702970)

1. Requirements Specification for phase - 2 (Advanced Functionalities & Security Enhancement)

1.1 Functional Requirements

Each of the below requirements is imperative to achieve the intended functionality and seamless operation of the system, ensuring secure and efficient interactions between different components and users.

1.1.1 Home Page

- Development of Home page with Title, Project Description and Team member names.
- Creation of a Menu with ability to navigate to all the five modules namely Data Owner, Data User, Trusted Authority, Proxy Server and CSP.
- On clicking on a menu item, the home page should be redirected to the respective login pages.

1.1.2 Data Owner Requirements

- **Registration:** The system should facilitate secure registration and login functionality for the Data Owner. New owners should be able to open the registration page by clicking hyperlink on the login page. Owners can register by providing Username, Password, Date of Birth, Email Id, Address and Mobile Number. All these fields are mandatory to register successfully and there is a constraint on password field to have a minimum length of eight to ensure security. On submitting, the owner record will be created in the database.
- **Login:** A newly registered owner should be authorized by trusted authority before logging in for the first time. This feature is added to enhance security by not allowing valid user to register as owner in the cloud. Once trusted authority approves, data owner should be able to login to his account by providing valid Username and Password. If credentials are wrong, an error message should display.
- **Home:** On successful login, Data Owner should be redirected to the home page, and the menu should be displayed with items such as Home, Upload, View Files, View Requests and Logout. On clicking logout it should be navigated to the owner login page.
- **Upload File:** Once logged in, the data owner should have the ability to upload files. The system should allow the uploading of text files of different sizes.
- **View Files:** Data owners should have a clear and concise dashboard that lists all their uploaded files with metadata such as upload date, file size and file name. They should also have the ability to filter and sort this list based on the metadata.
- **View Requests:** This feature allows the data owner to review all the requests from data users to download files. If data owner is willing to approve the request, he should have the ability to send re-encryption request to proxy server.

1.1.3 Data User Requirements

- **Registration:** Data users access a secure registration and login interface. They can open the registration page from the login screen, where they provide a Username, Password (with an eight-character minimum), Date of Birth, Email ID, Address, and Mobile Number for mandatory registration. Upon submission, user records are created in the database.
- **Login:** Similar to data owner, a newly registered user should be authorized by trusted authority before logging in for the first time. This feature is added to enhance security by not allowing valid user to register as owner in the cloud. Once trusted authority approves, data users should be able to login to his account by providing valid Username and Password. If credentials are wrong, an error message should display.

- **Home:** On successful login, data user should be redirected to the home page, and menu should be displayed with items Home, Search File, View Response, Downloads and Logout. On clicking logout user should be navigated to the login page.
- **Search File:** Data users, post-login, should have the capability to search for files uploaded by the owner using a keyword of file name. Results should be displayed in a tabular format and users should be able to request access to download those files. These requests should be sent to data owner.
- **View Response:** After the two-step verification by proxy server and CSP, the file is safely transported to its destination(user) along with the private key and is ready for decryption. All this information should be displayed in this tab, Using this private key, user should be able to decrypt and download the file.
- **Downloads:** All the history of files downloaded by user should be visible in this page.

1.1.4 Trusted Authority(Blockchain) Requirements

- **Login:** Trusted Authority should be able to log in securely with username and password. If Invalid credentials are entered, an error message should display.
- **Home:** On successful login, Trusted authority should be redirected to the home page, with menu items as View owners, View users, View ciphers and Logout. On clicking logout, TA should be navigated to the login page.
- **View Owners:** All new owner registrations should be displayed here, and TA can authorize them, by generating a membership key for each owner.
- **View Users:** All new user registrations should be displayed here, and TA can authorize them, by generating a membership key for each user.
- **View Cipher:** After two step encryption, all encrypted files should be stored(here in blockchain) and visible here in this page to oversee both the encryption and decryption procedures.

1.1.5 Proxy Server Requirements

- **Login:** The proxy server should have a secure login page. If Invalid credentials are entered, an error message should display.
- **Home:** On successful login, Proxy server should be redirected to the home page, with menu items as View requests, View URLs, and Logout. On clicking logout, proxy server should be navigated to the login page.
- **View Request:** This page is used by proxy server to track all re-encryption requests from data owners. Proxy server can authorize those requests and then send to CSP for further encryption.
- **View URLs:** Once CSP authorization is granted, a URL should be created. All such created URLs will be listed in this page. These URLs will be used to generate a re-encryption key, which is required before downloading. When the link is activated, encrypted file will be sent to both the user and blockchain(TA). This feature is added to prevent data users from sharing files to others after obtaining authorization through the two-step process involving owner, proxy server and CSP. This feature ensures that a new re-encryption key is required for each download instance.

1.1.6 Cloud Service Provider Requirements

- **Login:** The cloud service provider should have secure login functionality. If Invalid credentials are entered, an error message should display.
- **Home:** On successful login, CSP should be redirected to the home page, with menu items as View requests, View files, and Logout. On clicking logout, CSP should be navigated to the login page.

- **View Files:** Within this designated space, users gain the capability to browse through the entirety of files stored in the cloud. The user is assured that they are granted access solely for viewing purposes, providing a secure and controlled environment for file access without the ability to modify or alter the content.
- **View Requests:** This particular section is designed to display and manage all incoming requests from proxy server. CSP can then authorize those requests, which then enables the proxy server to generate private key to decrypt the file.

1.2 Non Functional Requirements - Phase-2

1.2.1 Security Requirements

Data Protection: The system must employ strong encryption algorithms to protect user data during transmission and while at rest.

1.2.2 Scalability Requirements

Load Management: The system should be able to handle a large number of simultaneous users. Ensuring file uploads/download without degrading performance.

Extension Capability: The architecture should allow for the addition of new features Components with minimal disruption to existing services.

1.2.3 Performance Requirements

Response Time: The system must respond to user requests, whether for uploading, downloading, or processing data, within acceptable time frames, ensuring a smooth user experience.

Throughput: The system should be able to process a high volume of data efficiently and should be capable of managing multiple transactions simultaneously.

1.2.4 Usability Requirements

User Friendly Interface: Usability plays an important role as this application works as a bridge between the user interface and should be well-designed. We will be using java server pages for making web pages user friendly and the UI will be intuitive and user-friendly, enabling users to easily navigate through the system and perform required operations without unnecessary complications.

Accessibility: The system should be accessible from various devices and browsers ensuring a broad user base can access it.

1.2.5 Maintainability Requirements

Modularity: The system should be modular to allow for easier maintenance, updates. The addition of new features.

Documentation: Comprehensive documentation should be maintained for every component. Functionality of the system to facilitate maintenance and further development.

2. Interfaces(User/Hardware/Software, and/or Communication)Developed Under phase-2

2.1 Hardware Interface requirements

This application would need a browser installed on Laptop / PC.

2.2 Software Interface requirements

1. HTML : We will be using html to structure our website.
2. CSS : We will be using css to design our website.
3. SQL: We will be using SQL for accessing the database.
4. JAVA : We will be using java to write our main components.
5. Apache Tomcat: Embedded web server.
6. NETBeans IDE: We will be using net beans as Integrated Development Environment for the Project.

2.3 User Interface

2.3.1 Home Page: Users will have the capability to access various modules on the home page, which include Data Owner, Data User, Trusted Authority, Proxy Server, Cloud Service Provider(CSP).

2.3.2 Sign Up Page: In the registration interface, users will be able to complete the registration process by providing the following essential details:

- Username: Users must select a unique and non-repetitive username to proceed; otherwise, an error will be generated.
- Password: It is imperative to create a secure and suitable password during registration.
- Date of Birth: Users are required to furnish their date of birth.
- Email Address: The registration process necessitates the provision of a valid and functioning email address.
- Gender: Users have the flexibility to select their gender from the available options, including male, female, or other.
- Mobile Number: The registration process includes the entry of the user's mobile number.

2.3.3 Login Interface: Users can access the various modules by providing the following details during the login process Username/Password,In addition, two buttons are available:

- Login: This button enables data owners to access their account by submitting the provided credentials.
- Clear: The "Clear" button offers users the convenience of swiftly removing or resetting their entries, enhancing the user experience for data owners.

2.3.4 Upload/Download Interface: Data owner and data user will have the capability to upload and download documents respectively using a dedicated button that facilitates the effortless transfer of files to and from their local hard disks.

- Upload : This button allows us to upload files into the cloud.
- Download : This button allows us to download the required files.

2.3.5 Table Interface : We will be able to view the data in tabulated manner in most of the webpages to provide more detailed and comprehensive information in each stage of Encryption, Re-encryption, Decryption and other aspects.

- View files and View requests in Data owner module
- View response and Downloads in Data user module
- View owners, View Users and View ciphers in Trusted Authority module
- View requests and View URLs in Proxy server module
- View files and View requests in CSP module

2.3.6 Dashboard Interface : Upon logging into each module, users will have access to multiple pages, each offering a range of distinct options. These options include:

Data Owner View:

- Home: The default landing page for data owners.
- Upload: This option empowers users to upload files securely by using the uploaded button.
- View Files: This feature allows users to access and view their uploaded files.
- View Requests: This option will offer access to view incoming requests.
- Logout: This function provides a streamlined process for data owners to log out of the module.

Data User View:

- Home : Here we will be able to see the homepages of data users.
- Search File: This option empowers data users to search for specific files of interest.
- View Response: Users can utilize this option to access and review responses to their requests.
- Download: This function allows data users to securely download files they have been granted access to.
- Logout: The Logout option provides a convenient means for data users to log out from the data user module when they have completed their session.

Trusted Authority View:

- Home : we will be able to view the home page of trusted authority which has multiple other files working in cohesion with this.
- View owners :New owner registrations appear here, and TA authorizes them, generating membership keys.
- View users : New user registrations are displayed here, and TA can authorize them, generating membership keys for access.
- View ciphers : we can view the cipher text and the encrypted and decrypted text here.
- Logout : we can logout of this module after we have completed working on this component.

Proxy Server View:

- Home: You will be able to see the home page of the proxy server module which is connected to multiple other webpages.
- View request: User requests are visible, and the proxy server authorizes via 'Owner Authenticity with Cloud' before forwarding the verified request to the CSP.
- View Urls: Once CSP authorizes, a URL is generated for re-encryption key creation, which must be entered prior to downloading. Upon link activation, it's transmitted to both the user and the blockchain.

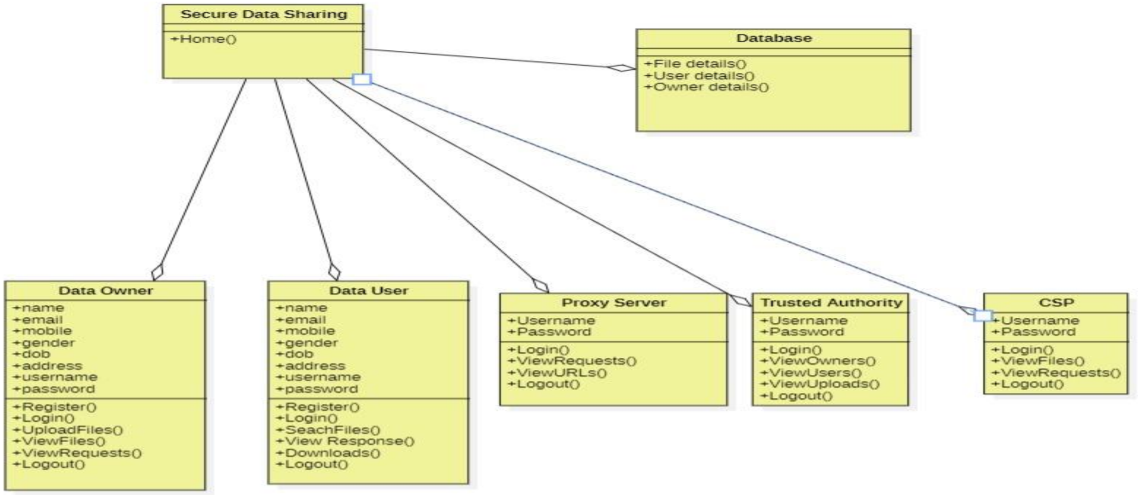
- Logout: we will be able to logout of the proxy server module as we complete our tasks.

CSP View:

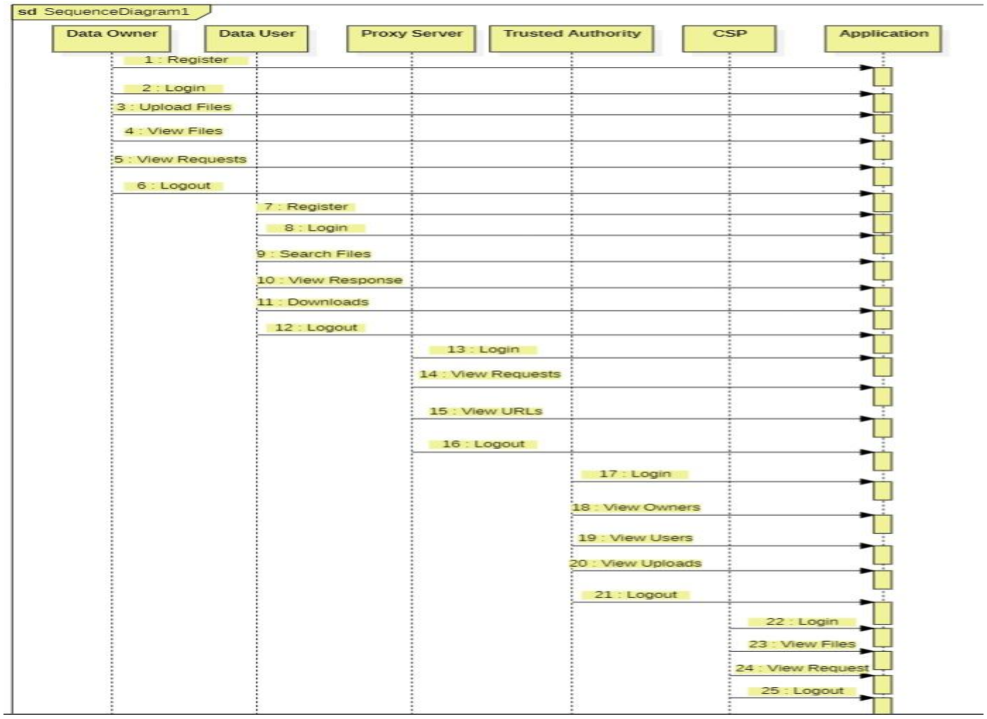
- Home: We will be able to view the cloud service provider’s home page which is connected to many other pages.
- View files: All files receive secure storage within the cloud, guaranteeing data protection and accessibility and the files can be viewed here.
- View request: Requests are not only displayed but also thoughtfully reviewed and accepted, ensuring a comprehensive and meticulous approach to handling each request.
- Logout: we can logout of the csp module after completing our respective tasks.

3. UML Diagrams

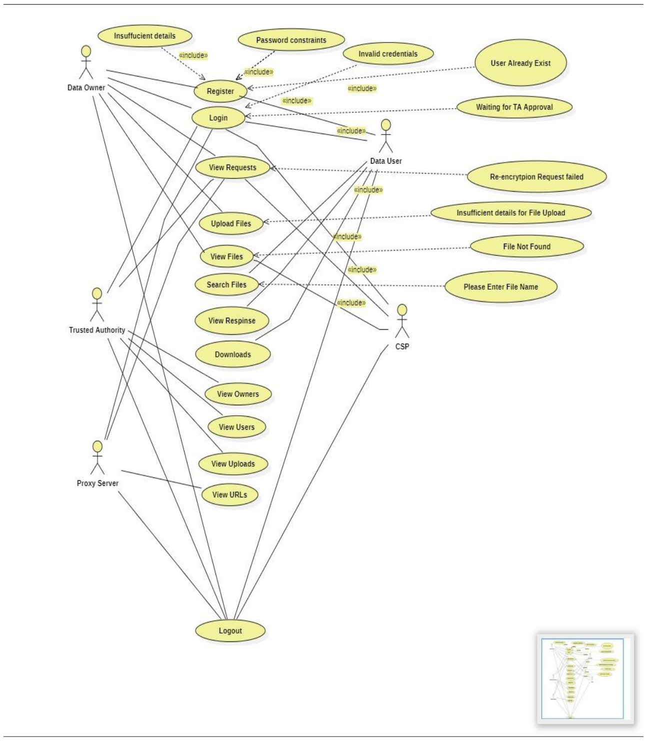
3.1 Class Diagram : Class diagram for “Data sharing using blockchain technology” with respect to the Phase – 2 Requirements. Rectangles depict objects with class names and attributes, while links illustrate object relationships through connecting lines, creating a visual representation of the system's structure and connections.

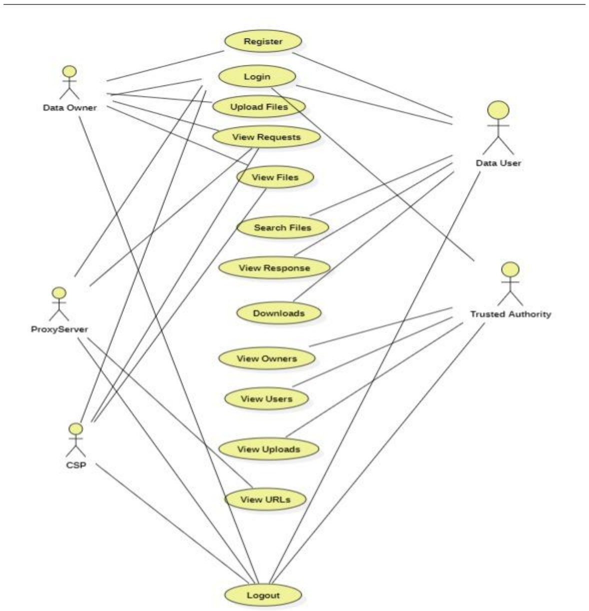


3.2 Sequence Diagram : Sequence diagram for “Data sharing using blockchain technology” with respect to the Phase – 2 Requirements. Entities communicating within the system are represented as objects, and in the sequence diagram, vertical lines denote their presence and interaction.



3.3 Use case diagram for Success and Error Case scenario: Use Case diagram for "Data sharing using blockchain technology" with respect to the Phase – 2 Requirements.





4. Unit Test Cases

Test Description	Input	Output	Test Result
Data owner upload a .mov file	any .mov file	Upload fail	Pass
Data owner upload a .txt file	any .txt file	Upload success	Pass
Data user search for non existing file	search string: zzzzzz	empty list	Pass
Data user search for existing file	search string: abstract	file details displayed	Pass
Data user downloading file uploaded by Data owner	any .txt file	File downloaded	Pass
View all uploaded files as CSP	login to CSP	See files in view all files	Pass
Data user search and request for file download	data user send download request	request sent successfully	Pass
View all download requests as CSP	data user send download request	All requests are shown	Pass
Send process request as Data user	Process request	status changed	Pass
Generate Member key when a user is created	data user is created	member key is generated	Pass
Process download request as Proxy server and send to CSP	process request as proxy server	Request moved to CSP	Pass
Reencrypt the file as Proxy	process file request	send request to user after re encryption	Pass
Decrypt the file as data user user private key	private key	file should be decrypted and show download link	Pass
Enter wrong private key as data user	Random String	Verification Failed alert	Pass
Generate private keys for data user	user is created	private key is generated	Pass
Download the file as a data user	data user click on download link	file is downloaded	Pass

5. User Manual for Phase - 2

The utilization of blockchain technology for data sharing is encapsulated within a web-based application, which serves as a platform for users to securely store and disseminate their data while preserving a comprehensive record of its usage. To initiate the implementation of this application's second phase on your local computer, kindly adhere to the program instructions delineated within this document.

5.1 Home page: On the main page, users will employ the navigation buttons positioned at the top of the interface to traverse through the various sections of the website. Multiple login options have been incorporated into the website to cater to diverse user profiles, each granting access to dedicated login pages. It is important to note that this page also encompasses a project description and introduction for further context and understanding.

5.2 Data Owner: The Data Owner page will feature a registration and login interface, where data owners can create their accounts. To successfully register, data owners are required to furnish credentials like Username,password,date of birth ,email-id,gender and mobile number. Additionally, users will encounter both "Submit" and "Clear" buttons on the registration page.Following a successful login, the data owner will gain access to their dedicated page, which encompasses several tabbed options for navigation. These options include :

- **Home:** The default landing page for the data owner.
- **Upload:** This option enables the user to upload files.
- **View Files:** This feature allows the user to view their uploaded files.
- **View Requests:** This feature allows the data owner to see all the requests placed to download his files.owner can authorize by clicking on 'Send Re-Encrypt Request'. re-encryption requests will be sent to the proxy server.
- **Logout:** This option will allow you to logout of this module and redirect you back to the webpage.

5.3 Data User: The Data User page provides a login and registration interface, similar to the one used by data owners. Data users can create an account and log in to place requests to access files from other users. To register or log in we need to use username and password.Upon successful login, data users will have access to various options within their tab. These options include:

- **Home:** The default landing page for the data User.
- **Search File:** This option enables data users to search for specific files of interest by entering a key word in the space and entering the search button all the related files will be displayed on the page.
- **View response:** After implementing two-step verification and proxy encryption, the file is securely transferred to its destination and is primed for decryption. Begin by copying the private key, then proceed to the "Decrypt File" option. Paste the private key, and initiate the decryption process by clicking "Verify." At this point, the file's data will be revealed and is now available for download.
- **Downloads:** Data user can see all the files he downloaded previously here.
- **Logout:** This option will allow you to logout of this module and redirect you back to the webpage.

5.4 Trusted Authority: The Trusted Authority page requires users to provide the following details for authentication and login: Username/Password.This page will include multiple features like home, view owners, view users, view ciphertext, logout and these include the following functionality :

- **Home:** The default landing page for the Trusted authority.
- **View Owners:** New owner registrations are displayed here, and TA can authorize them, generating a membership key for each.
- **View Users:** New user registrations are listed here, and TA can authorize them, generating a membership key for each.
- **View ciphers:** This is where you can view all the encrypted text in files and monitor the encryption and decryption processes.
- L**ogout:** This option will allow you to logout of this module and redirect you back to the webpage.

5.5 Proxy Server: The "Proxy Server" tab encompasses a server login page. To access this server, users are required to provide the following inputs Username/Password.Upon successful login we will be able to view the home page of the proxy server.The proxy server tab home, view request, view urls, logout will contain the following functionalities :

- **Home:** The default landing page for the Proxy Server.
- **View request:** All user requests are visible here. Proxy server authorizes via 'Owner Authenticity with Cloud' after two-step verification. Then, the request is sent to the CSP.
- **View Urls:** Once the CSP has granted authorization, a URL will be generated. This URL is used to create a re-encryption key, which must be entered before downloading. When the link is clicked, it will be sent to both the user and the blockchain.once user got authorization from owner and proxy server in two step way to download file, he can share it to many users. To avoid this, we are using a proxy server again here to reencrypt, so that it will generate a new re-encryption key every time.
- **Logout:** This option will allow you to logout of this module and redirect you back to the webpage.

5.6 Cloud Service Provider: The CSP (Cloud Service Provider) login page requires users to provide the following details for authentication and login: Username/Password.We can see the following Tabs after login home, view all files, view request, encryption time graph, re-encryption time graph, decryption time graph, all downloads graph, attack file graph, logout and these are the functionality included :

- **Home:** The default landing page for the Cloud service provider.
- **View files:** In this dedicated space, you have the ability to view each file that has been uploaded to the cloud, with the assurance that only viewing access is granted.
- **View request:** In this designated area, all incoming requests are not only visible but also thoughtfully reviewed and accepted, ensuring a comprehensive and meticulous approach to handling each request.
- **Logout:** This option will allow you to logout of this module and redirect you back to the webpage.

6. Program Compilation and Run Instructions

Now let us understand the compilation instructions. The compilation instructions of phase-1 and phase-2 have very minimal changes as phase-2 includes addition of multiple other functionality and running of some excess files which can be cloned from github using the “Git clone” command.
Pre-Requisites :

1) Navigate to the Control Panel on your Windows device within the Program Folder. Delete any components associated with Apache Tomcat by right-clicking and selecting 'Uninstall' for all related directories.

2) Proceed to remove any existing Java Development Kit installations on your device, as we will be utilizing Java SE 8 (version 202 and later).
3) Uninstall any applications in your Programs folder related to MySQL Server.
4) To ensure the complete removal of certain files from the C drive, navigate to the Program Files and perform a permanent deletion (Shift + Delete) of the following items:

1. Apache Software Foundation
2. All files associated with MySQL
3. All files related to Java Development Kits

6.1 Program Run Instructions

1) Clone the designated GitHub repository using the Command Prompt with the command 'git clone' followed by the path to the GitHub repository containing all project software.(https://github.com/NitinReddyUNT/SE_Project_Teamsigma)
2) We have to install multiple software components to make our project feasible.
3) We need to install jdk-8u144 windows-x64" on our computer from https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html
4) We need to install my sql version “mysql-essential-5.0.67-win32” from the github program file “SE_Project_Teamsigma”
5) After the installation is complete, access the Control Panel and search for "MySQL Server Instance Config Wizard." Start the installation process and set the new root password as "root" before completing the installation.
6) Install apache tomcat version 8.0.27 with the following link: https://ipt.gbif.org/manual/en/ipt/latest/tomcat-installation-windows.
7) We need to download netbeans ide version “"netbeans-8.1-windows" and Configure the NetBeans IDE installer by opening the file. Choose the 'Customize' option and select Apache Tomcat 8.0.27. Complete the installation process.
8) Download "Webyog_SQLyog_6.5.6_enterprise" and use the key “TaMaBMBolo” and serial : “270c1144ab1730d”.
9) Launch "Webyog_SQLyog_6.5.6_enterprise/SQLyog656Ent" to initiate the installation process, and input the give key and serial.
10) After entering the credentials, create a new connection named "New Connection," with the password set as "root." You will be presented with a window labeled "SQLyog Enterprise - MYSQL GUI - [New Connection - root@localhost]."
11) As we have already cloned our github repository for SE_Project_Teamsigma.
12) Open the "Proxy.sql" file in Notepad and copy the included queries.
13) Paste the Queries in SQLyog Enterprise -MySQL GUI and click the execute all queries button on the top menu bar to execute the queries after the execution you will be able to see the test folder and we can view the tables in the database in the proxy folder in SQLyog Enterprise.
14) Now open netbeans ide 8.1 in the system and select the file option /select open project option from the dropdown box and select the file which has been cloned from github.
15) Now we will be able to view all the code files in Netbeans IDE. Next we have to import all the libraries. For that we need to right click on the project main folder named “SE_project_teamsigma” a drop down box will appear and we need to select the properties option.
16) Click on libraries on the options in the properties menu click on add.jar folder option in the menu and navigate to the “SE_Project_Teamsigma/libs” folder in your local computer select all the libraries in that folder and click open.
17) Delete all the previously existing references and only keep the newly added libraries.
18) After completing the process, enter ok and close the window and click the run option in the netbeans library to run the project on your local host.
19) In phase -2 you will be able to see excess files when you have cloned the repository which will get executed using netbeans and you will be able to test the excess webpages and functionality.

7. Code Inspection Feedback

During the Code Inspection session for our project, we received the following feedback from our peers and we made the following changes to our code/project and we have taken the following measures to improve.

1) Suggestion: They have asked us to share project updates and challenges regularly.
Action Taken: We have started updating weekly status meetings to review project progress and risks. We document action items and follow up on them during subsequent meetings. We have shared project updates and taken measures. overcome challenges.
2) Suggestion: Discuss potential process improvements for the projects during meetings and time management.
Action Taken: We held a process improvement brainstorming session to gather ideas. We identified a few potential enhancements to explore further and assigned team members to set up schedules during meetings.
3) Suggestion: They asked us to add authors to each code file to make it more clear.
Action Taken: We will analyze our codebase and add developer name tags for each file for clear attribution. We will update the coding standards wiki to include this practice for all new files.
4) Suggestion: Review and address recent paired team feedback.
Action Taken: We compiled the feedback from our last paired team. We identified the top improvement areas and have created action plans to improve them.
5) Suggestion: Consider enhancing certain functionalities.
Action Taken: We are evaluating the proposed enhancements from a technical feasibility and priority perspective.
6) Suggestion: Improve password restrictions for our login pages.
Action Taken: Based on the guidance, we implemented improved password logic adding length, complexity, and other requirements per our security policy.
7) Suggestion: Include encryption/decryption for the documents.
Action Taken: We communicated that encryption features are planned for Phase 2. We provided high-level design docs for review and we have implemented the features we intended to in phase 2 succesfully.

8. Report Reflection

We have accomplished the tasks that we have committed to do in the development of phase-2 as specified in the requirement specification documentation. We have specified to develop advanced functionalities such as file encryption and decryption and concentrate on secure file transactions while enhancing security protocols and addressing vulnerabilities. To achieve this we have implemented multiple other functionalities in Data Owner, Data User, Trusted Authority, Proxy Server, Cloud Service Provider modules. So far we are good at developing all these functionalities with security policies intact. Below are some of the additional functionalities in our modules :

Data Owner module: In the data owner module we have added view requests functionality where the data owner can view the requests placed by data users to download the files whichever the owner have uploaded. This helps the data owner to track the users whoever downloaded his files.

Data User module: In the data user module we have added a view response tab where the user will be able to decrypt the file using the private key for further verification which will allow the data to be downloaded only to the users whoever requested access to it.

Trusted Authority module: In the trusted authority module we have added three new functionalities named view owners which will allow the trusted authority to view all the owners, view users is also a similar functionality which will allow the trusted authority to view all the data users and finally we have view ciphers functionality which allows to view the re-encrypted file details.

Proxy Server module: In the proxy server module we have implemented two functionalities named view requests which allows the server to view all re-encryption requests from data owners and other functionality named View urls which generates a re-encryption private key needed for downloading. This key is sent to both the user and the blockchain. When a user receives permission from the owner and proxy server for a file download, they can share it widely. To prevent this, we use a proxy server to re-encrypt the file, creating a new key each time.

CSP module: Finally in the csp module we have included two functionalities in this phase which include view files tab where every file which is secured in the cloud can be overseen and view request functionality which has the capability to see the requests and decide whether to accept or reject them.

So far the implementation of our project phase-2 went well as per our plan without any potential major errors and we are planning on enhance our csp module in our next phase by adding additional functionalities like calculated graphs.

9. Member Contribution Table

Member Name	Contribution Description	Overall contribution (%)	Note(If Applicable)
Akhila pam(11711224)	Updated Userlog.jsp and added TA_ViewCipher, TA_ViewUsers, TA_ViewOwners.	11.11	
Akshara Reddy Bathula (11713259)	Worked on SP_viewRequest.jsp, PS_ownerAuthenticity.jsp.	11.11	
Jyothi Anjan Manini (11715079)	Updated SearchFile.jsp, SearchAction.jsp,VerifyAction .jsp, Download.jsp Files. Created ViewRespone.jsp, VerifyPVTKey.jsp,Request action.jsp and worked on Report Reflection,UML Diagram in Deliverable 4 Documentation	11.11	
Lakshmichatura Medidi (11682526)	Worked on UML diagrams, Report requirements in documentation. Worked on upload and download files as well as generation of	11.11	

	membership key functionality.		
Manoj Kumar Bandari (11711378)	Worked on view request and re-encryption requests functionality as well as in documentation part	11.11	
Nimitha Bangalore Sathyanarayana (11649788)	Worked on Unit test cases, CSP module, fixed styling issues,worked on documentation, taking meeting minutes, and updated SQL commands.	11.11	
NitinReddy Balaiahgari (11698724)	Update Userhome.jsp and Ownerhome.jsp and added additional java classes Mail.java,encryption.java,Decryption.java and blockchain java class and worked on deliverable 4 documentation.	11.11	
Satya Laxman Pranav Vadlamani (11701928)	Worked on code inspection feedback part in documentation and home pages of proxy server and CSP modules	11.11	
Sumuk Reddy Kalagiri (11702970)	Worked on url response and re encryption parts in proxy server as well as documentation in deliverable 4	11.11	