Deliverable 5 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 - 3 (Refinement & Optimization)

### 1.1 Functional Requirements

In this last phase -3, we have focused on refining the overall system, optimizing performance in some features like generation of membership key in trusted authority module and generation of url decryption key in proxy server. We also focused on fixing css issues, enhancing user interactions, and conducting extensive testing to ensure the reliability and stability of the system.

### 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.
- For this phase-3, we have enhanced this dashboard to display the file cipher data and actual dat along with the file name, uploaded date and file size.
- **View Requests:** This feature allows the data owner to review all the requests from data users to download files. If the data owner is willing to approve the request, he should have the ability to send re-encryption requests to the 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.
- For phase-3, we have reduced the private key generation time compared to the previous approach we have followed by adding a decrypt key to this ‘view response’ page itself so that user can copy and use it 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.
- For Phase-3, View Ciphers feature has been updated to display data in a tabular format with details such as file name, owner name, requested data user name, request status and re-encrypted file data.

### 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.

- For phase-3, View URLs feature has been updated to display data in a tabular format with details such as file name, owner name, requested data user name, request status along with the url to decrypt the file.

### 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.
- This feature has been updated in phase-3 to view all files uploaded by data owner along with date uploaded, file name, file cipher data in a tabular format.
- **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.
- **Encryption Time Graph:** This feature is implemented in cloud service provider module to enable the csp to track all the files with the time taken to encrypt those files. When data owner sends file to proxy server for encryption, based on the file size and time taken to encrypt the file, this graph is generated with files on X-axis and encryption time in nanoseconds on Y-axis.
- **Re-Encryption Time Graph:** This feature is implemented in cloud service provider module to track all the files along with the time taken to re-encrypt them. After proxy server encrypts the file, the file is sent to re-encryption by csp. This graph is generated with files on X-axis and re-encryption time in nanoseconds on Y-axis.
- **Decryption Time Graph:** This feature is added to cloud service provider module to track all the files along with the time taken to decrypt them. After, decryption url is generated by proxy server, decryption key is generated in data user to decrypt the file. This graph depicts the time taken for this process with files on X-axis and decryption time in nanoseconds on Y-axis.
- **All Downloads Graph:** This feature is implemented in cloud service provider module to track the number of files download by a data user. After file is decrypted and downloaded by data users, this graph will be generated with total number of files downloaded by a particular user with count of files on X-axis and users names on Y-axis.
- **Attacked File Graph:** This graph is implemented in cloud service provider module to track the total number of times a particular file is downloaded by different data users with count of files on Y-axis and file names on X-axis.

## 1.2 Non Functional Requirements - Phase-3

### 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-3

### 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 Graph Interface :** We will be able to view the graphical representation of the request and response time of some of the main features such as Encryption, Re-encryption, Decryption of the project in the cloud service provider module

- Encryption time graph between files and encryption time in CSP module
- Re-Encryption time graph between files and re-encryption time in CSP module
- Decryption time graph between files and decryption time in CSP module
- All Downloads graph between total number of a files and a single user in CSP module
- Attacked file graph between a file and total number of users downloaded it in CSP module

**2.3.7 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.

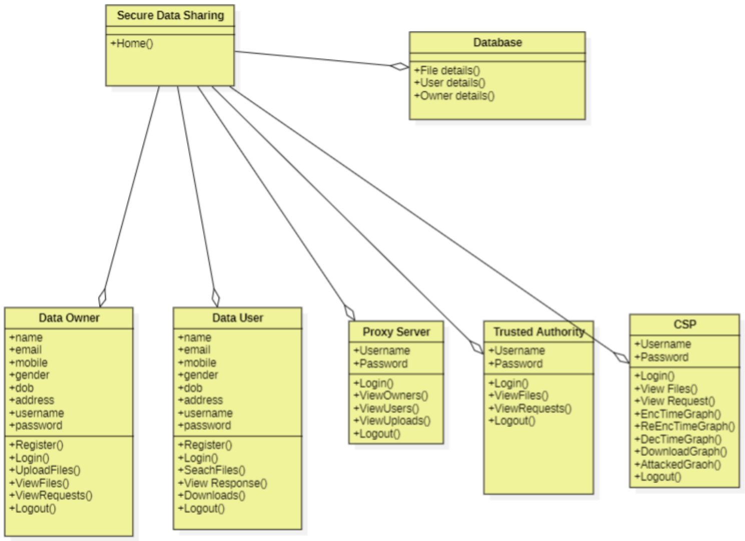
**CSP View:**

- 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.
- Encryption time graph: This graph is used to display the files against the time taken to encrypt the file with files on X-axis and encryption time in nanoseconds on Y-axis.

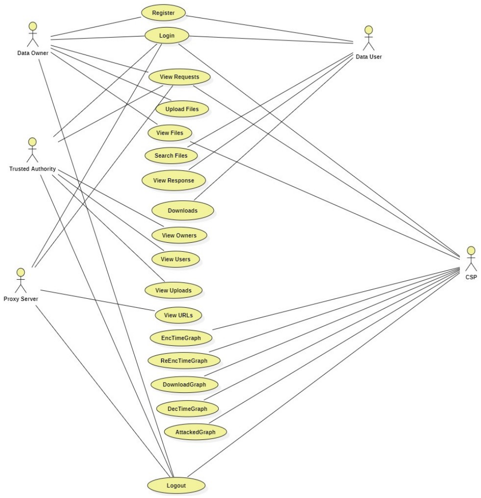
- Re-encryption time graph: This graph is used to display the files against the time taken to re-encrypt the file with files on X-axis and re-encryption time in nanoseconds on Y-axis.
- Decryption time graph: This graph is used to display the files against the time taken to decrypt the file with files on X-axis and decryption time in nanoseconds on Y-axis.
- All downloads graph: This graph is used to display the total number of files downloaded by a particular user with the count of files on X-axis and usernames on Y-axis.
- Attacked file graph: This graph is used to display the total number of times a particular file is downloaded by different data users with count of files on Y-axis and file names on X-axis.
- Logout: we can logout of the csp module after completing our respective tasks.

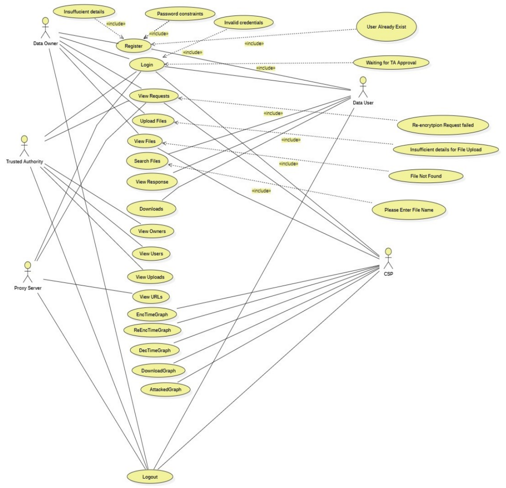
## 3. UML Diagrams

**3.1 Class Diagram :** Class diagram for “Data sharing using blockchain technology” with respect to the Phase – 3 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.

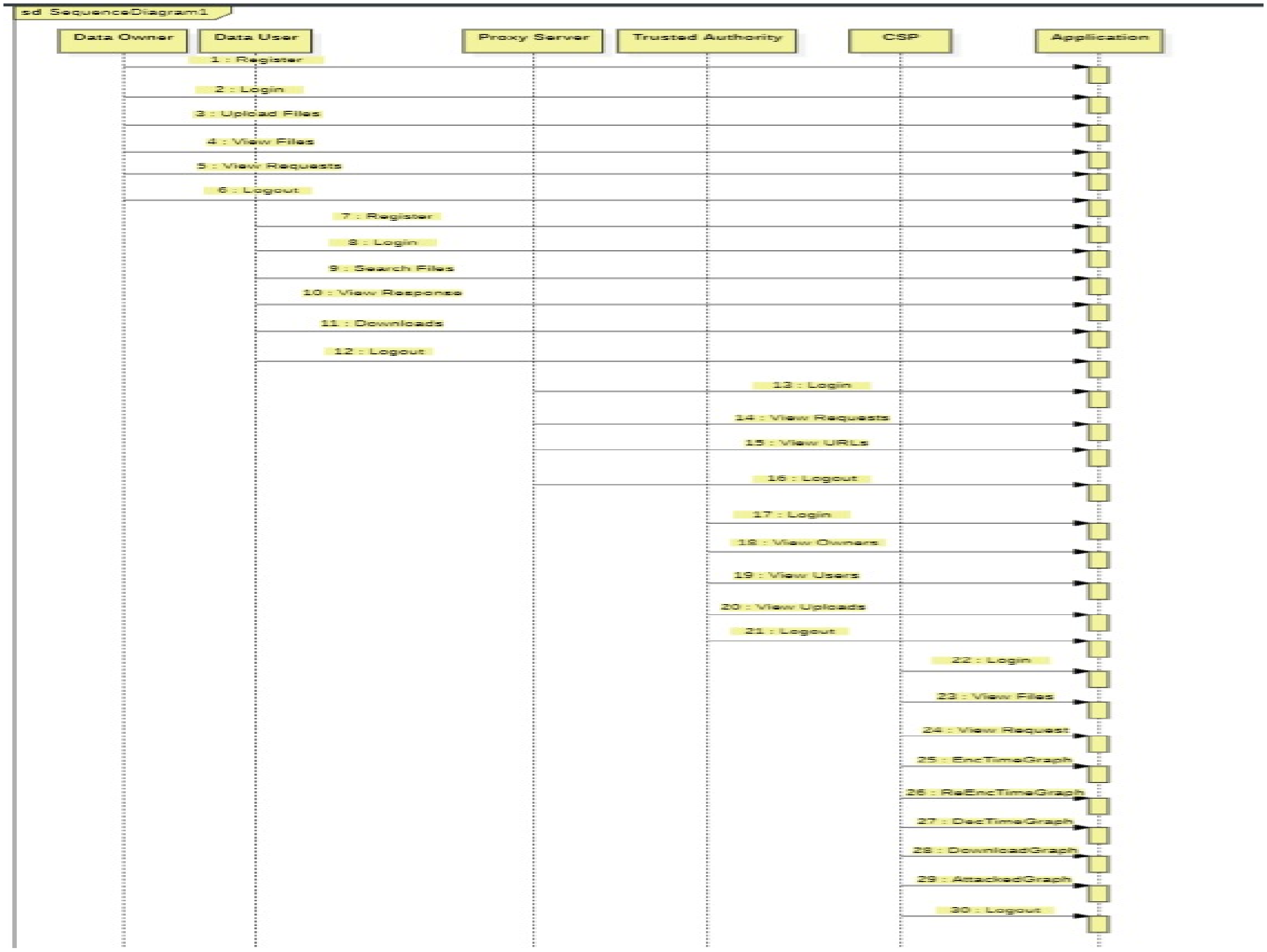


**3.3 Use case diagram for Success and Error Case scenario:** Use Case diagram for “Data sharing using blockchain technology” with respect to the Phase – 3 Requirements.





**3.2 Sequence Diagram :** Sequence diagram for "Data sharing using blockchain technology" with respect to the Phase – 3 Requirements. Entities communicating within the system are represented as objects, and in the sequence diagram, vertical lines denote their presence and interaction.



## 4.1 Unit Test Cases

| Test Description | Input | Output | Test Result |
|---|---|---|---|
| Upload a file and request for encryption | Any .txt file | Update graph in encryption file with the file name | Pass |
| Request a file and request for re-encryption | Any .txt file | Update graph in re-encryption file with the file name | Pass |
| Request a file and request for decryption | Any .txt file | Update graph in decryption file with the file | Pass |

| | | | |
|---|---|---|---|
| | | name | |
| Request a file and request for download | Any .txt file | Update graph in download file with the file name | Pass |
| Search for a file and request download multiple times | Any .txt file | Update graph in attacked file with the file name | Pass |
| Display time taken for a file encryption | Request any .txt file | Display time in encryption file with the file name | Pass |
| Display time taken for a file re-encryption | Request any .txt file for re-encryption | Display time in re-encryption file with the file name | Pass |
| Display time taken for a file decryption | Request any .txt file for decryption | Display time in decryption file with the file name | Pass |
| Display time taken for file download | Download any .txt file | Display time in downloads with the file name | Pass |
| Display time taken for a file for file download | Search for a file and request download multiple times | Display number of tries in attacked files with the file name | Pass |

## 4.1 System Test Cases

| Test Description | Input | Output | Test Result |
|---|---|---|---|
| Start the application server | Project set up as expected | Apache Server started | Pass |
| Verify Server is connected to database | Start server and check DB connection | Database is connected | Pass |
| Verify all pages are loaded as expected | Start server | HTML of pages is loaded | Pass |
| System performance | System is used by 5 | System doesn't time out | Pass |

| | | | |
|---|---|---|---|
| test | users at a time | or errors out | |

## 5. User Manual for Phase - 3

1) 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 third and final phase on your local computer, kindly adhere to the program instructions delineated within this document.

2) After running the project, we can see the homepage of the website which contains a mini project description and the team members names including their enrollment id. Users will be able to navigate to the menu items positioned at the top of the interface to traverse through the various sections of the website. (or) We can access the website with the below provided link:

https://www.securedatasharingubt.cloud/SE_Project_Teamsigma/index.html

3) We can browse all of the options by following the given steps. Start with the Data Owner option and click on ‘Register’ by providing the following details: Username, Password(of 8 characters), Date of Birth, Email id, Gender, Address, Mobile Number and we need to click ‘Submit’ to register a new owner account with the database.

3) If you try to login now, you will get an error message, because as this is a secured platform every new registration needs to be verified by the Trusted Authority. We can now login as Trusted Authority using the following credentials:

Username : Authority & Password : Authority

4) After we login as Trusted Authority, click on ‘View Owners’ menu and we can see a new registration request has been placed. Now we click on ‘Generate’ to generate a membership key to the new owner so they will be able to login.

5) Logout the Trusted Authority page and go to Data Owner login and provide your credentials and now you will be able to login. Here in the Data Owner page, click on ‘Upload’ files option, select a text file which you want to upload and click on ‘Upload’.

6) Now we will be able to view the Metadata, Hash code, Digital signature and other necessary details for uploading the files, click on ‘Upload’, now the file will be moved to cloud and upon clicking the ‘View Files’ option data owner will be able to see all the files uploaded by him.

7) Now logout from Data Owner and move to Data User module and ‘Register’ in the same procedure using steps 3,4,5.

8) Now login as Data User and you will be able to see the ‘Search file’ option, enter the file name which you have uploaded as Data Owner and click on search button, the file should appear now click on the ‘Request’ link to request for downloading the file, now the request will be sent to Data Owner.

9) Now login as Data Owner and click on ‘View Request’ option and owner can click on ‘Send Re-Encrypt Request’ then the request will be sent to the proxy server.

10) Now logout as data user and login as proxy server with the following credentials.

Username : server & Password : server

11) Go to the ‘View Request’ menu in the proxy server and select the file and click on ‘Owner Authenticity with Cloud’ and now the request will be sent to the Cloud Service Provider. This is the encryption step in the project where the file uploaded by the owner will be encrypted here.

12) Now logout of proxy server and login into Cloud Service Provider with the following credentials.

Username : csp & Password : csp

13) Now here in Cloud Service Provider, click on ‘View Requests’ option here you will be able to view all the requests and can verify them by clicking on ‘Owner authenticity verify’. This is the re-encryption of the file in our project by CSP.

14) Now Logout from CSP and login into Proxy Server with the previously provided credentials and open ‘View URLs’ option and click on the URL generated for your file, this will generate a decryption key and now the file is ready to decrypt by Data User and decryption keys will be sent to both Trusted Authority(blockchain) and to Data User.

15) Now login as Data User and click on the ‘View Response’ tab and search for the file you require. Now copy the ‘Private Key’ and click on ‘Decrypt File’ and enter the private key and click on verify and now we will be able to see the file data and download it.

16) All the files downloaded by Data User can be seen here in ‘Download’ tab in Data user module.

17) On clicking ‘View Files’ you will be able to view all the files uploaded by data owners in the cloud with their ciphered data.

18) In CSP we will also be able to see multiple graphs related to multiple functionalities. The first one is Encryption time graph between file name and time taken to encrypt in nano seconds. Verify the graph with the file name that was uploaded and time taken to encrypt it.

19) The second graph is Re-encryption time graph between file name and time taken to re-encrypt the file in nano seconds. Verify the graph with the file name that was uploaded and time taken to re-encrypt it.

20) The next graph is Decryption time graph between file name and time taken to decrypt the file in nano seconds. Verify the graph with the file name that was uploaded and time taken to decrypt it.

21) The next graph is All Downloads graph between the total number of files downloaded by a particular user. Download multiple files and verify the graph with the files uploaded against the data user name.

22) The last graph is Attacked file graph between total number of times a file is downloaded by multiple users.

## 6. Program Compilation and Run Instructions

**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

**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 - 3 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. Report Ending Feature Summary

With the completion of final phase-3 of the project, we have accomplished all the tasks we intended to do during project design phase and requirements specification phase. We have successfully developed all the five core modules of our project Data Owner, Data User, Trusted Authority, Proxy Server and Cloud Service Provider. Even though we have faced some issues during development phase-1 for integrating all the modules and establishing connections between them, we were able to resolve them with the good team work and finally we do not have any limitations. Below are some of the features of the project we successfully developed for the project segregated according to the modules..

**Data Owner:** We have successfully implemented the data owner module with registration, login, upload files, view requests, view files and logout functionalities without any issues.
**Data User:** In the data user module, we have successfully implemented the registration, login, search files, place requests, view response, download files and logout functionalities.
**Trusted Authority:** We have implemented the view owners and view users functionalities where authority can authorize the new owners and users by generating a membership key for them. Additionally we implemented the view ciphers where all the re-encrypted files are stored for future purpose.
**Proxy Server:** In the proxy server module, we have successfully implemented the view re-encrypt request feature so that requests can be approved or rejected and the url generations tab is also developed to generate a decryption key for files once csp approved the request.
**Cloud Service Provider:** In CSP module, we have implemented the view files module to see all the files uploaded to cloud and view requests feature to see all the re-encryption requests from proxy server. We have also added some time graphs to measure and analyze the time taken by files to encrypt, re-encrypt, decrypt and download.

**Future Work:**

- In the future, we want to introduce a new feature where Data owners can upload multiple files at a time and also where owners can upload files of different formats concurrently. This feature aims to enhance the overall user experience and increase the overall system efficiency.
- Additionally, we want to implement the handling of multiple download requests from data users in optimal time, so that the responsiveness of the whole system is further enhanced.
- We would like to introduce even more robust security measures such as one time passcode, captcha code to protect the data of users from potential risks and ensure the data integrity of the website is maintained.
- Furthermore, we would like to enhance the overall aesthetics of the website through css and provide a seamless and user-friendly interface to the users.
- And, finally we would like to introduce multiple languages features to make the website accessible and usable to a wide range of audience across the world.

## 8. Report Reflection

We have accomplished the tasks that we have committed to do in the development of final phase-3 as specified in the requirement specification documentation. We have specified to implement some new features such as graphs in cloud service provider modules as well as to refine the overall system by optimizing performance of some features like membership key generation and private key generation, enhancing user interface, fixing css issues, conducting extensive testing and checking the reliability and stability of the system. We also enhanced some existing functionalities in Data Owner, Data User, Trusted Authority, Proxy Server, Cloud Service Provider modules to achieve optimal performance. Below are the additional functionalities we have added in this phase.

**Data Owner module:** In the data owner module, we have thoroughly tested the upload files module by testing with files of different sizes. We have also added some css to view requests tab based on the request stage either pending or processed or rejected.

**Data User module:** In the data user module, we have optimized the view response tab by displaying private key along with file data which enables the user to copy the decryption key and download the file immediately. We have also worked on adding some font colors in this module.

**Trusted Authority module:** In the trusted authority module we have optimized the membership key generation method to generate membership keys for data owners and data users in an effective and fast way so that new owners and new users can immediately login after trusted authority approval.

**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:** In the CSP module we have added some new functionalities in this phase which include Encryption time graph, Re-encryption time graph, De-encryption time graphs, All Downloads graph and, Attacked file graphs. All the graphs are used to measure the total number of files and time taken to encrypt, re-encrypt, decrypt and download them.

Overall, the project has been implemented well, and we are happy with the features that were determined upon during the requirements process. We've completed every task and in future we would like to implement multiple features as described in the future work section. Although we faced multiple challenges during all project phases, all the teammates helped each other and displayed good teamwork to forfeit the challenges that occurred and resolved them successfully.

## 9. Member Contribution Table

| **Member Name** | **Contribution Description** | **Overall contribution (%)** | **Note(If Applicable)** |
|---|---|---|---|
| Akhila pam(11711224) | Worked on Decryption time graph web page as well as java class and on ppt presentation. | 11.11 | |
| Akshara Reddy Bathula (11713259) | Worked on Re-encryption time graph web page as well as java class and on ppt presentation. | 11.11 | |

| | | | |
|---|---|---|---|
| Jyothi Anjan Manini (11715079) | Worked on UML Diagrams in Deliverable 5 Documentation and on Attacked file graph feature in CSP module. | 11.11 | |
| Lakshmichatura Medidi (11682526) | Worked on Report requirements, user manual and on drafting final documentation for deliverable 5 as well as on testing, bug fix and on download time graph feature in CSP module. | 11.11 | |
| Manoj Kumar Bandari (11711378) | Worked on Encryption time graph feature in CSP module by developing web page & class and on ppt presentation. | 11.11 | |
| Nimitha Bangalore Sathyanarayana (11649788) | Worked on Unit test cases, css enhancements, handling git and on Encryption time graph feature in CSP. | 11.11 | |
| NitinReddy Balaiahgari (11698724) | Worked on ppt and enhancing existing functionalities and on Re-encryption time graph feature in CSP. | 11.11 | |
| Satya Laxman Pranav Vadlamani (11701928) | Worked on Decryption time graph feature in CSP as well as on ppt presentation. | 11.11 | |
| Sumuk Reddy Kalagiri (11702970) | Worked on Attacked file graph feature in CSP and on Usability testing. | 11.11 | |