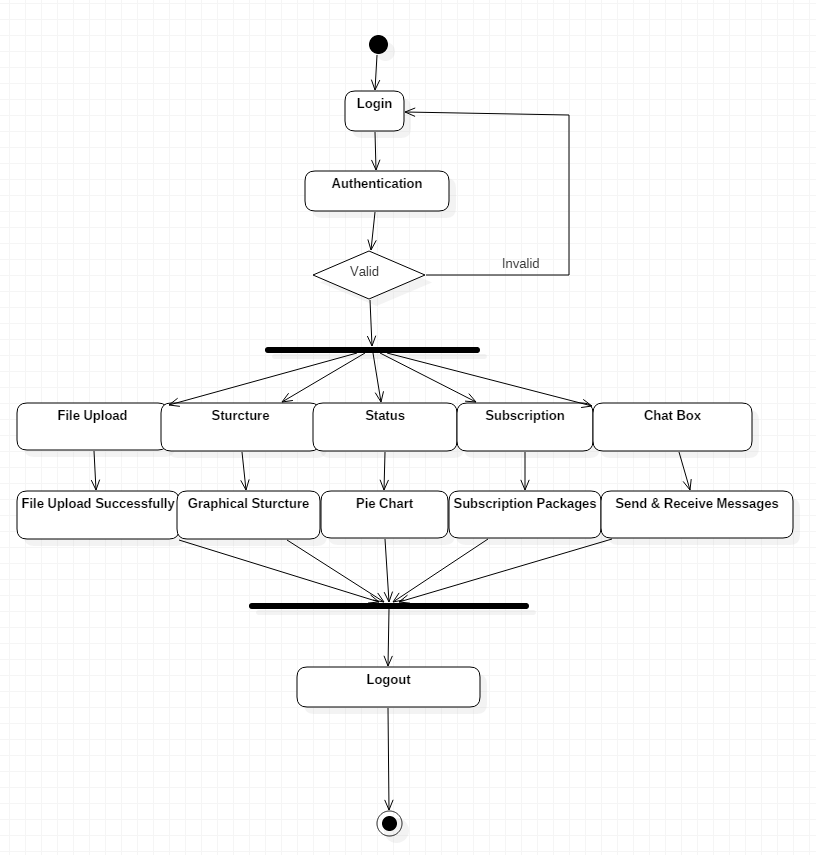
**Second Progress Report**

**System Study**

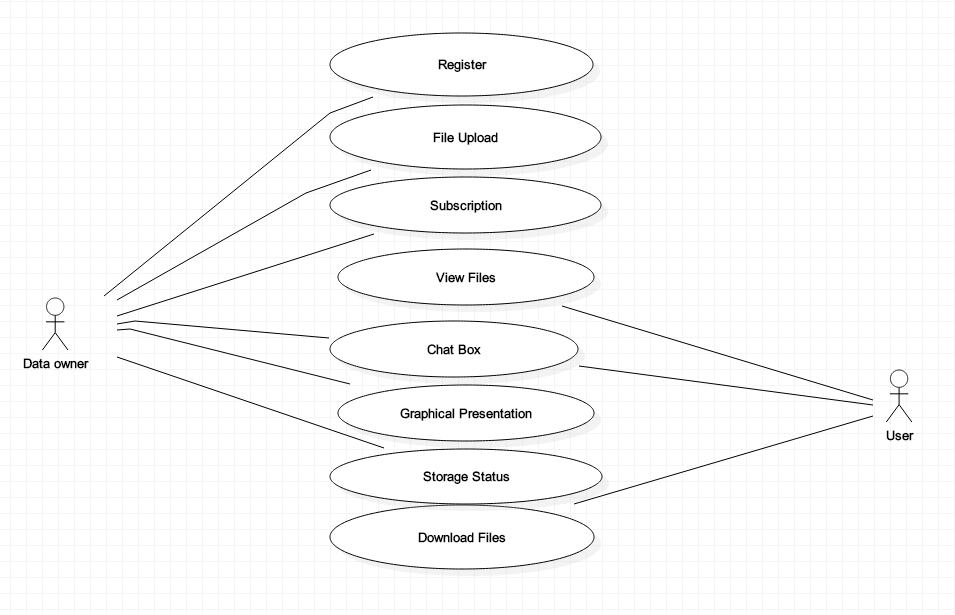
* **Problem definition :**
* Cloud storage is a service model in which data is maintained, managed, backed up remotely and made available to users over a network (typically the Internet).
* Users generally pay for their cloud data storage on a per-consumption, monthly rate.
* Although the per-gigabyte cost has been radically driven down, cloud storage providers have added operating expenses that can make the technology more expensive than users bargained for.
* Cloud security continues to be a concern among users.
* Providers have tried to deal with those fears by building security capabilities, such as encryption and authentication, into their services
* **Existing System :**
* In Existing System we use a straightforward integration method. In straightforward integration method Storing data in a third party’s cloud system causes serious concern on data confidentiality. In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying an erasure code method to encode and store messages. When he wants to use a message, he needs to retrieve the
* Codeword symbols from storage servers, decode them, and then decrypt them by using cryptographic keys.
* General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data.
* A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority.
* **Disadvantages of Existing System :**
* The user can perform more computation and communication traffic between the user and storage servers is high.
* The user has to manage his cryptographic keys otherwise the security has to be broken.
* The data storing and retrieving, it is hard for storage servers to directly support other functions.
* **Proposed System :**
* In our proposed system we address the problem of forwarding data to another user by storage servers directly under the command of the data owner.
* We consider the system model that consists of distributed storage servers and key servers.
* Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user.
* These key servers are highly protected by security mechanisms.
* The distributed systems require independent servers to perform all operations.
* We propose a new threshold proxy re-encryption scheme and integrate it with a secure decentralized code to form a secure distributed storage system.
* The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages.
* **Advantages of Proposed System** :
* Tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.
* The storage servers independently perform encoding and re-encryption process and the key servers independently perform partial decryption process.
* More flexible adjustment between the number of storage servers and robustness.



**Fig- Activity Diagram**

**Problem Definition :**

* In Existing System we use a straightforward integration method. In straightforward integration method Storing data in a third party’s cloud system causes serious concern on data confidentiality.
* In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying an erasure code method to encode and store messages.
* When he wants to use a message, he needs to retrieve the Codeword symbols from storage servers, decode them, and then decrypt them by using cryptographic keys.
* General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data.
* A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority.
* The user can perform more computation and communication traffic between the user and storage servers is high.
* The user has to manage his cryptographic keys otherwise the security has to be broken.
* The data storing and retrieving, it is hard for storage servers to directly support other functions.
* In our proposed system we address the problem of forwarding data to another user by storage servers directly under the command of the data owner.
* We consider the system model that consists of distributed storage servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user.
* These key servers are highly protected by security mechanisms.
* The distributed systems require independent servers to perform all operations.
* We propose a new threshold proxy re-encryption scheme and integrate it with a secure decentralized code to form a secure distributed storage system.
* The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages.
* Tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.
* The storage servers independently perform encoding and re-encryption process and the key servers independently perform partial decryption process.
* More flexible adjustment between the number of storage servers and robustness.



**Fig – Use Case Diagram**

**Requirement Gathering :**

**Feasibility Study :**

* A feasibility study is the study of positive possibilities of the project. It is also measure of how beneficial or practical development of clinic management system would be to an organization.
* The different types of feasibilities are as follows:
  + Economic Feasibility
  + Technical Feasibility
  + Behavioral Feasibility

1. **Economic Feasibility :** Our application is economically feasible as it determines the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. Along with this it need not require any additional hardware resources as well as it will be saving lot of time.
2. **Technical Feasibility :** According to feasibility analysis procedure the technical feasibility of the Cloud storage System is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. This application offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced.

The Technical Feasibility composed of estimating:

1. Hardware Cost.
2. Software Cost.
3. **Behavioral Feasibility:**

Behavioral Feasibility states that an estimate should be made of how strong the user is likely to move towards the development of cloud system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the Cloud storage.

**Test Cases :**

**Login Module :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Input Data | | Expected O/P | Actual O/P | Status |
| email | password |
| T001 | User is being authenticated using ID & Pass | [abc@xyz.com](mailto:abc@xyz.com) | NULL (not provided) | Throws an Error Message | Throws an Error Message | Pass |
|  |  | NULL (not provided) | NULL (not provided) | Throws an Error Message | Throws an Error Message | Pass |
| T002 | Assumption : It is wrong ID & Password | [abc@xyz.com](mailto:abc@xyz.com) | 123 | Throws an Error Message | Throws an Error Message | Pass |
| T003 | Assumption : It is right ID & Password | [abc@xyz.com](mailto:abc@xyz.com) | 1234 | Redirected to Home Page | Redirected to Home Page | Pass |
| T004 | Asssumption:  Connection is disconnected | [abc@xyz.com](mailto:abc@xyz.com) | 1234 | Error 404 | Page not found | Pass |
| T005 | Assumption:User  Is blocked or inactive | [abc@xyz.com](mailto:abc@xyz.com) | 1234 | Throws an Error Message | User is Inactive | Pass |
| T006 | Assumption:  subscription has ended |  |  | Throws an Error Message | Your subscription has been expired renew it. | Pass |

**Registration Module :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Steps | Input Data | Expected O/P | Actual O/P | Status |
| T0001 | User is Registering  Himself/herself | Entering Name | abc | Validated | Validated | Pass |
|  |  |  | abc1 | Error Messgae | Alpha characters only | Pass |
|  |  |  | abc$ | Error Messgae | No special characters | Pass |
|  |  | Entering Email | abc@gmail.com | Validated | Validated | Pass |
|  |  |  | abc@ | Error Messgae | Error Messgae | Pass |
|  |  |  | Abc.gmail.com | Error Messgae | Error Messgae | Pass |
|  |  | Entering password | 123 | Error Messgae | Error Messgae  Atleast 8 characters | Pass |
|  |  |  | aB@1234 | validated | validated | Pass |
|  |  |  | 123456 | Error Messgae | Use atleast one upper case, lower case , special character | Pass |
|  |  | Entering mobile no | 1234567890 | Validated  . | validated | Pass |

**File Upload Module :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Steps | Input Data | Expected O/P | Actual O/P | Status |
| T0001 | User is Uploading  File | Entering file name | abs | validated | validated | Pass |
|  |  | Choosing file | a.html | Select file | File selected | Pass |
|  |  | Description | An html file | validated | validated | Pass |
|  |  | Send alerts | None of them is selected | Error Message | Please Select  At least one | Pass |
|  |  |  | One of them is  selected | validated | validated | Pass |
|  |  | Encryption  technique | Any one of them is selected | validated | validated | Pass |
|  |  |  | None selected | Keep Open | Publicly  available | Pass |
|  |  | password | password | validated | validated | Pass |
|  |  |  |  |  |  |  |
| T0002 | If success upload | Checking for Session |  | validated | validated | Pass |
| T0003 | If upload fails |  |  | Redirected to login | Redirected to login | Pass |

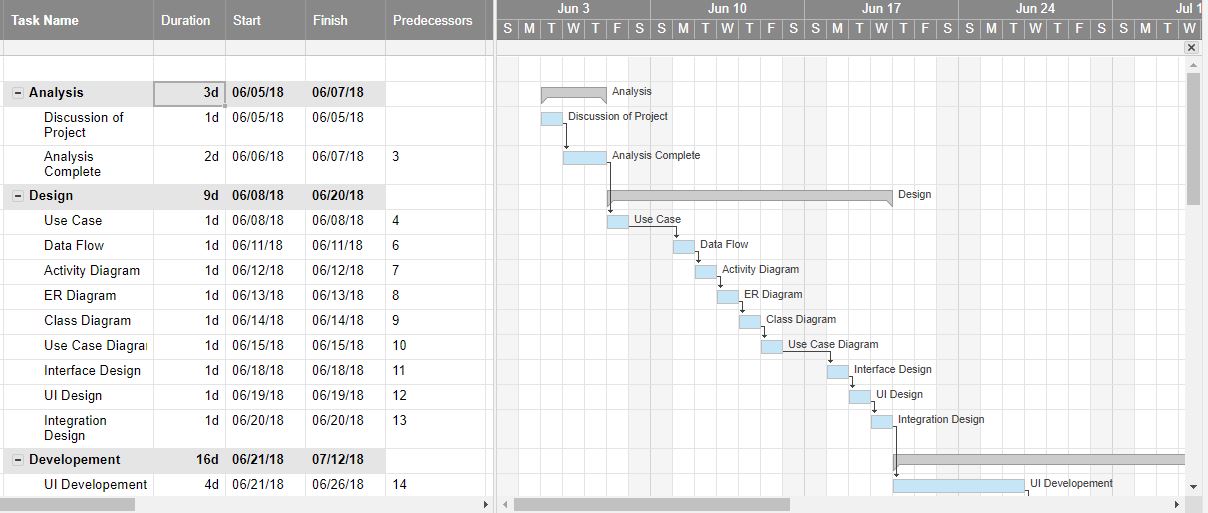
**File Download Module :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Steps | Input Data | Expected O/P | Actual O/P | Status |
| T0001 | User downloading his/her own file | Chechking session |  | Allowed | Allowed | Pass |
|  |  | If fails |  | Redirection | Redirected to login | Pass |
|  |  |  |  |  |  |  |
| T0002 | User Requesting for download | Asking Permission to owner |  |  |  |  |
|  |  | If allowed |  | Output Sent | Output Sent | Pass |
|  |  | If ignored |  | Rejected | Rejected | Pass |
|  |  |  |  |  |  |  |
| T0003 | User has OTP trying to download | Entering OTP | Entering Key 1234  Pass:ATAS  Assuming it is right | Success | Allowed to download | pass |
|  |  |  | Key:1234  Pass:AATS  Assuming it is wrong | Discarded | Discarded;  Sent an email to owner about incident | Pass |

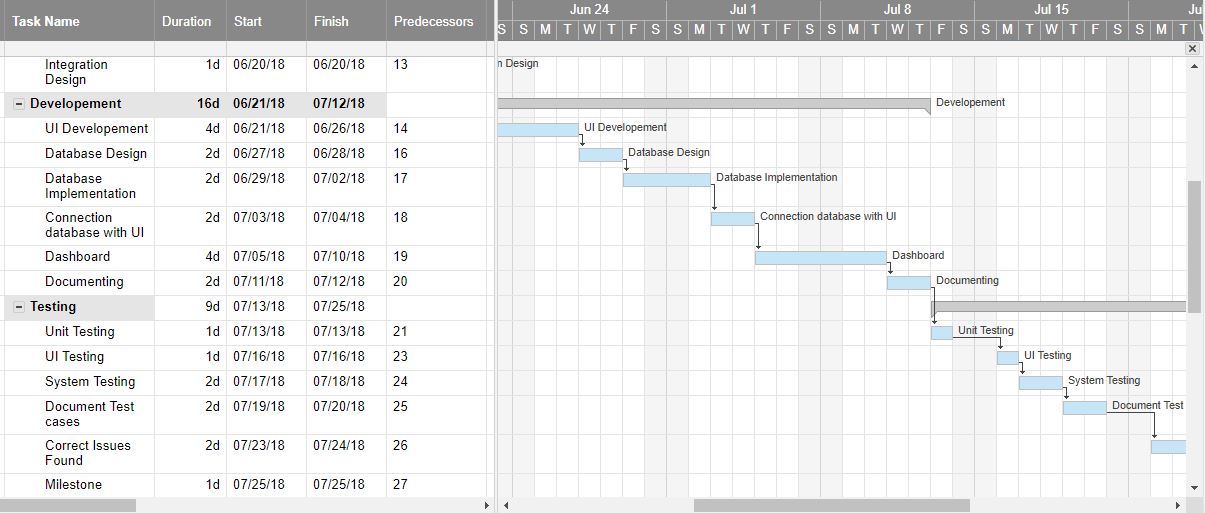
**Subscription Module :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Steps | Input Data | Expected O/P | Actual O/P | Status |
| T001 | Check For user  Subscription status |  |  |  |  |  |
|  |  | If user is in subscription period |  | Redirected to home | Redirected to home | Pass |
|  |  | If user is not in subscription period |  | Redirected to Subscription  page | Redirected to Subscription  page | Pass |

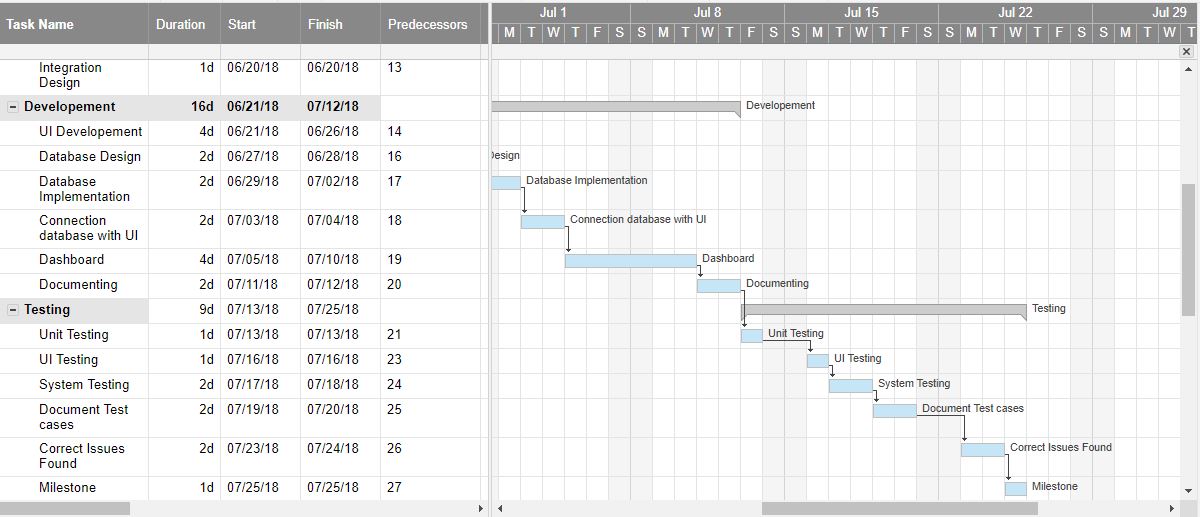
**Gantt Chart :**

****

**Fig : Gantt Chart -1**

****

**Fig : Gantt Chart -2**

****

**Fig : Gantt Chart -3**

**Operating Tools and Technologies:**

* **Hardware Requirements :**

1. Processor             - Pentium –III
2. Speed                    -    2.0 Ghz
3. RAM                    -    512 MB(min)
4. Hard Disk            -    20 GB
5. Key Board            -    Standard Windows Keyboard
6. Mouse                  -    Two or Three Button Mouse
7. Monitor                -    SVGA

* **Software Requirements:-**

1. Operating System          : Windows XP/7/8/8.1/10
2. Application  Server       : GlassFish Server 4.X
3. Front End                      : Java, JSP
4. Script                            : JavaScript.
5. Server side Script          : Java Server Pages.
6. Database                        : Microsoft SQL Server 2012
7. IDE USED : NETBEANS 8.2