**CHAPTER 1**

**INTRODUCTION**

**1.1 GENERAL**

The Platform-as-a-Service cloud storage has become an infrastructure service of the Internet as a promising way to simplify storage management for enterprises and individual users. Coupled with the increasing demand for multi-device data synchronization and sharing, it is emerging as a new paradigm that helps migrate storage applications to the cloud. Due to its practical impact, significant research endeavors have been undertaken to address the problems in cloud storage based applications, such as the security of storage outsourcing, data consistency, and cost optimization. A large body of work has advanced the state of art of cloud storage research, including but not limited to the topics mentioned above. In particular, a recent workable [38] proposed a cloud-based storage solution, called blue sky for the enterprise, which acts as a proxy to provide the illusion of a traditional file server and transfer the requests to the cloud via a simple HTTP-based interface. CLOUD computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk .

From users’ perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward users’ outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user’s ultimate control over the fate of their data.

As a result, the correctness of the data in the cloud are being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that have not been or are rarely accessed, or even hide data loss incidents to maintain a reputation.

CLOUD computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk . From users’ perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward users’ outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user’s ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that have not been or are rarely accessed, or even hide data loss incidents to maintain a reputation.

A growing number of online services, such as Google, Yahoo!, and Amazon, are starting to charge users for their storage. Customers often use these services to store valuable data such as email, family photos and videos, and disk backups. Today, a customer must entirely trust such external services to maintain the integrity of hosted data and return it intact. Unfortunately, no service is infallible. To make storage services accountable for data loss, we present protocols that allow a third party auditor to periodically verify the data stored by a service and assist in returning the data intact to the customer.

Cloud computing is the long dreamed vision of computing as a utility, where data owners can remotely store their data in the cloud to enjoy on-demand high-quality applications and services from a shared pool of configurable computing resources. While data outsourcing relieves the owners of the burden of local data storage and maintenance, it also eliminates their physical control of storage dependability and security, which traditionally has been expected by both enterprises and individuals with high service-level requirements. In order to facilitate rapid deployment of cloud data storage service and regain security assurances with outsourced data dependability, efficient methods that enable on-demand data correctness verification on behalf of cloud data owners have to be designed.

The cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter); the third-party auditor, who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to FCFS for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from FCFS. Using third-party auditing service provides a cost-effective method for users to gain trust in cloud.

. CLOUD computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk . While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward users’ outsourced data. With the booming cloud computing industry, computational resources are readily and elastically available to the customers. In order to attract customers with various demands, most Infrastructure-as-a-service (IaaS) cloud service providers offer several pricing strategies such as pay as you go, pay less per unit when you use more.

**1.2 SCOPE OF THE PROJECT:**

In this project we are going to implement we present the design and implementation of a cost-effective file system based on the platform-as-a-Service cloud.

**1.3 EXISTING SYSTEM**

**1.3.1 EXISTING CONCEPT:**

In a traditional file system, extensive optimizations for local disk-Based storage back end cannot fully exploit the inherent features of the cloud to obtain desirable performance. By intelligently organizing storage objects in a local cache.

**1.3.2 EXISTING TECHNIQUE:**

HTTP-based interface

**1.3.3 TECHNIQUE DEFINITION:**

A server-side web API is a programmatic interface consisting of one or more publicly exposed endpoints to a defined request–response message system, typically expressed via the web—most commonly by means of an HTTP-based web server.

**1.3.4 DRAWBACKS:**

* High cost file system based on the cloud storage.
* Local disk-Based storage.

**1.4 PROPOSED SYSTEM**

**1.4.1 PROPOSED CONCEPT:**

Here we present the design, implementation, and evaluation of Coral, a cloud based file system that strikes a balance between performance and monetary cost.

**1.4.2 PROPOSED TECHNIQUE:**

FCFS- First Come First Serve is a method employed by many IT companies. It is base is taken from the operating system concepts. First Come First , from the name itself it is understood that whatever comes first is always given the first priority.

**1.4.3 TECHNIQUE DEFINITION:**

A frugal storage model optimized for scenarios concerning multiple cloud storage services. Similar to local hierarchical storage systems, FCFS integrates cloud services with very different structures. By dynamically adapting the storage volume sizes of each service.

**1.4.4 ADVANTAGES:**

The design and implementation of a cost-effective file system based on the cloud storage. A frugal storage model optimized for scenarios concerning multiple cloud storage services.

**CHAPTER 2**

**LITERATURE SURVEY**

**Title:** A Five-Year Study of File-System Metadata.

**Author:** Nitin Agrawal, William J. Bolosky , John R. Douceur, Jacob R. Lorch

**Year:** 2007

**DESCRIPTION:**

For five years, we collected annual snapshots of file system metadata from over 60,000 Windows PC file systems in a large corporation. In this paper, we use these snapshots to study temporal changes in file size, file age, file-type frequency, directory size, namespace structure, file-system population, storage capacity and consumption, and degree of file modification. We present a generative model that explains the namespace structure and the distribution of directory sizes. We find significant temporal trends relating to the popularity of certain file types, the origin of file content, the way the namespace is used, and the degree of variation among file systems, as well as more pedestrian changes in sizes and capacities. We give examples of consequent lessons for designers of file systems and related software.

**Title:** Multidimensional Binary Search Trees Used for Associative Searching

**Author:** J. L. Bentley,

**Year :**2009

**DESCRIPTION:**

This paper develops the multidimensional binary search tree (or k-d tree, where k is the dimensionality of the search space) as a data structure for storage of information to be retrieved by associative searches. The k-d tree is defined and examples are given. It is shown to be quite efficient in its storage requirements. A significant advantage of this structure is that a single data structure can handle many types of queries very efficiently. Various utility algorithms are developed; their proven average running times in an n record file are : insertion, O(log n); deletion of the root, 0 (n (k--1)/k) ; deletion of a random node, O(log n); and optimization (guarantees logarithmic performance of searches), 0 (n log n). Search algorithms are given for partial match queries with t keys specified [proven maximum running time of O (n (k-t)/k) ] and for nearest neighbor queries [empirically observed average running time of O(log n). ] These performances far surpass the best currently known algorithms for these tasks. An algorithm is presented to handle any general intersection query. The main focus of this paper is theoretical. It is felt, however, that k-d trees could be quite useful in many applications, and examples of potential uses are given

**Title:** DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds

**Author:** A. N. Bessani, M. P. Correia, B. Quaresma, F. Andre and P. Sousa

**Year:** 2011

**DESCRIPTION:**  
 The increasing popularity of cloud storage services has lead companies that handle critical data to think about using these services for their storage needs. Medical record databases, power system historical information and financial data are some examples of critical data that could be moved to the cloud. However, the reliability and security of data stored in the cloud still remain major concerns. In this paper we present DEPSKY, a system that improves the availability, integrity and confidentiality of information stored in the cloud through the encryption, encoding and replication of the data on diverse clouds that form a cloud-of-clouds. We deployed our system using four commercial clouds and used Planet- Lab to run clients accessing the service from different countries. We observed that our protocols improved the perceived availability and, in most cases, the access latency when compared with cloud providers individually. Moreover, the monetary costs of using DEPSKY on this scenario is twice the cost of using a single cloud, which is optimal and seems to be a reasonable cost, given the benefits.

**Title:** SCFS: A Shared Cloud-backed File System

**Author:** A. N. Bessani, R. Mendes, T. Oliveira, N. F. Neves, M. Correia, M. Pasin, and P. Verissimo

**Year:** 2014

**DESCRIPTION:**

Despite of their rising popularity, current cloud storage services and cloud-backed storage systems still have some limitations related to reliability, durability assurances and inefficient file sharing. We present SCFS, a cloud-backed file system that addresses these issues and provides strong consistency and near-POSIX semantics on top of eventually-consistent cloud storage services. SCFS provides a pluggable backplane that allows it to work with various storage clouds or a cloud-of-clouds (for added dependability). It also exploits some design opportunities inherent in the current cloud services through a set of novel ideas for cloud-backed file systems: always write and avoid reading, modular coordination, private name spaces and consistency anchors.

**Title:** BORG: Block-reorganization for Self-optimizing Storage Systems

**Author:** M. Bhadkamkar, J. Guerra, L. Useche, S. Burnett, J. Liptak, R. Rangaswami, and V. Hristidis

**Year:**2009

**DESCRIPTION:**

This paper presents the design, implementation, and evaluation of BORG, a self-optimizing storage system that performs automatic block reorganization based on the observed I/O workload. BORG is motivated by three characteristics of I/O workloads: non-uniform access frequency distribution, temporal locality, and partial determinism in non-sequential accesses. To achieve its objective, BORG manages a small, dedicated partition on the disk drive, with the goal of servicing a majority of the I/O requests from within this partition with significantly reduced seek and rotational delays. BORG is transparent to the rest of the storage stack, including applications, file system(s), and I/O schedulers, thereby requiring no or minimal modification to storage stack implementations. We evaluated a Linux implementation of BORG using several real-world workloads, including individual user desktop environments, a web-server, a virtual machine monitor, and an SVN server. These experiments comprehensively demonstrate BORG’s effectiveness in improving I/O performance and its incurred resource overhead.

**Title:** Hystor: Making the Best Use of Solid State Drives in High Performance Storage Systems

**Author:** F. Chen, D. A. Koufaty, and X. Z. 0001

**Year:** 2011

**DESCRIPTION:**

With the fast technical improvement, flash memory based Solid State Drives (SSDs) are becoming an important part of the computer storage hierarchy to significantly improve performance and energy efficiency. However, due to its relatively high price and low capacity, a major system research issue to address is on how to make SSDs play their most effective roles in a high-performance storage system in cost- and performance-effective ways. In this paper, we will answer several related questions with insights based on the design and implementation of a high performance hybrid storage system, called Hystor. We make the best use of SSDs in storage systems by achieving a set of optimization objectives from both system deployment and algorithm design perspectives. Hystor manages both SSDs and hard disk drives (HDDs) as one single block device with minimal changes to existing OS kernels. By monitoring I/O access patterns at runtime, Hystor can effectively identify blocks that (1) can result in long latencies or (2) are semantically critical (e.g. file system metadata), and stores them in SSDs for future accesses to achieve a significant performance improvement. In order to further leverage the exceptionally high performance of writes in the state-of-the-art SSDs, Hystor also serves as a write-back buffer to speed up write requests. Our measurements on Hystor implemented in the Linux kernel 2.6.25.8 show that it can take advantage of the performance merits of SSDs with only a few lines of changes to the stock Linux kernel. Our system study shows that in a highly effective hybrid storage system, SSDs should play a major role as an independent storage where the best suitable data are adaptively and timely migrated in and retained, and it can also be effective to serve as a write-back buffer.

**Title:** To Cloud Or Not To Cloud? Musings On Costs and Viability

**Author:** Y. Chen and R. Sion

**Year:** 2012

**DESCRIPTION:**

In this paper we aim to understand the types of applications for which cloud computing is economically tenable, i.e., for which the cost savings associated with cloud placement outweigh any associated deployment costs. We discover two scenarios. In an (i) “unified client” scenario, once cloud-hosted, applications are meant to be accessible only to a single cloud customer (or small set of associates). It then becomes important to ensure that the cost savings (mainly computation-related) can offset the often significant client-cloud distance (network costs etc). In a (ii) “multi-client” setting on the other hand, outsourced applications serve numerous different thir-parties. We show that then clouds begin to act similarly in nature to content-distribution networks – by comparison, their better network integration is simply too good to pass on, when compared to locally hosting the applications (and incurring associated network costs). Ultimately, we hope this work will constitue a first step in an objective evaluation of the technological side of costs of outsourcing and computing in general. We discover two scenarios. In an (i) “unified client” scenario, once cloud-hosted, applications are meant to be utilized.

**Title:** Inside Dropbox: Understanding Personal Cloud Storage Services

**Author:** I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A. Pras.

**Year:** 2005

**DESCRIPTION:**

Personal cloud storage services are gaining popularity. With a rush of providers to enter the market and an increasing offer of cheap storage space, it is to be expected that cloud storage will soon generate a high amount of Internet traffic. Very little is known about the architecture and the performance of such systems, and the workload they have to face. This understanding is essential for designing efficient cloud storage systems and predicting their impact on the network. This paper presents a characterization of Dropbox, the leading solution in personal cloud storage in our datasets. By means of passive measurements, we analyze data from four vantage points in Europe, collected during 42 consecutive days. Our contributions are threefold: Firstly, we are the first to study Dropbox, which we show to be the most widely-used cloud storage system, already accounting for a volume equivalent to around one third of the YouTube traffic at campus networks on some days. Secondly, we characterize the workload users in different environments generate to the system, highlighting how this reflects on network traffic. Lastly, our results show possible performance bottlenecks caused by both the current system architecture and the storage protocol. This is exacerbated for users connected far from storage data-centers.

**CHAPTER 3**

**SOFTWARE REQUIEMENTS SPECIFICATION**

**3.1 GENERAL**

We analyze the problem of unsecure storage of files in a cloud environment.

**3.2 PROBLEM DEFINITION**

The cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter); the third-party auditor, who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance.

They may also dynamically interact with the CS to access and update their stored data for various application purposes. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to FCFS for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from FCFS.

Using third-party auditing service provides a cost-effective method for users to gain trust in cloud. We assume the FCFS, who is in the business of auditing, is reliable and independent. However, it may harm the user if the FCFS could learn the outsourced data after the audit. Note that in our model, beyond users’ reluctance to leak data to FCFS; we also assume that a cloud server has no incentives to reveal their hosted data to external parties.

As there an increase in usage of computational resource in these days. The cloud service providers like Information As A Service(IAAS) provides discounts for the brokers. Brokers act as mediators between the cloud service providers and customers. The broker provides the customers only required amount of computational resource at low cost. The organizations are connected to the storage application services. In this the customers are also directly connected to the storage application service providers.

**3.3 METHODOLOGIES**

Methodologies are the process of analyzing the principles or procedure for enabling secure uploading of files against data integrity with the support of First Come First Serve Process in a cloud environment.

**3.3.1 MODULES**

* **USER INTERFACE DESIGN**
* **COMPANY GET SPACE**
* **ADMIN PROCESS**
* **FILE UPLOAD**
* **COMPANY ADMIN PROCESS**
* **SEARCH FOR FILES**

**3.3.2 MODULE DESCRIPTION**

**User Interface Design**

To connect with server user must give their username and password, then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. The Server will create the account for the entire user to maintain upload and download rate. The name will be set as user id. . Logging is usually used to enter a specific page

**Company Get Space**

The company gets the space for cloud that time we need to pay the amount for cloud providers. The company goes to the money transfer the process the company having an account and transfer the amount the amount gets the cloud provider and give cloud provide for the company. The company is using the cloud space that was used in the file and cloud space.

**Admin Process**

The admin process admin is can login and go into the process for the user how much data company used. The admin how much file size are used the company and company user. The process of the admin that only does the values and it can be used the company.

**File Upload**

The Company enters the process and file upload and show the file for cloud, the cloud show for the file was encrypted it was working to file stored in the cloud. The Authorizes are getting data and upload the data in the database. Process file upload that time file was encrypted the file store the value in the database.

**Company Admin Process**

The Company admin how many members buy into the cloud for us and how much of amount of data are used and how much of the data we are using. To monitor the all data usage the data usage for the details it was working it was to store the file amount and storage.

**Search for Files**

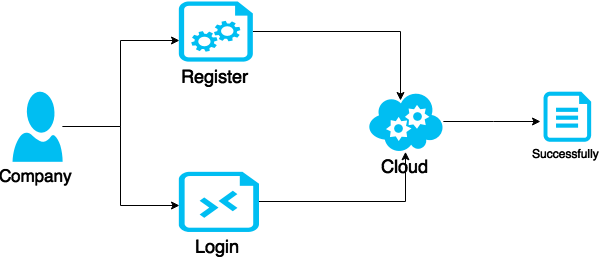
Once you are going to enter the page there is a lot of files there is search button whatever files you want just enter the file name and click submit button. It will go and fetch the file from the database and display to the user. If the file you want select and download the files whatever the files you need it.They may also dynamically interact with the CS to access and update their stored data for various application purposes.

As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to FCFS for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from FCFS. As there an increase in usage of computational resource in these days. The cloud service providers like Information As A Service(IAAS) provides discounts for the brokers. Brokers act as mediators between the cloud service providers and customers.

Using third-party auditing service provides a cost-effective method for users to gain trust in cloud. We assume the FCFS, who is in the business of auditing, is reliable and independent. However, it may harm the user if the FCFS could learn the outsourced data after the audit.

**3.3.3 MODULE DIAGRAM**

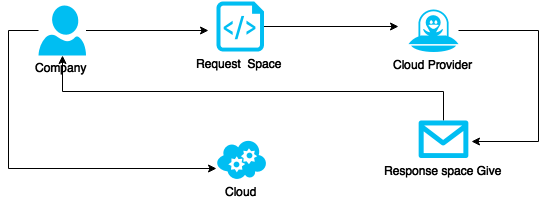
**User Interface Design**

****

**Fig 3.3.1: User interface design**

As Shown in the Fig 3.3.1 Firstly, to connect with server user must give their username and password, then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. The Server will create the account for the entire user to maintain upload and download rate. The name will be set as user id. Logging is usually used to enter to a specific page.

**Company Get Space**

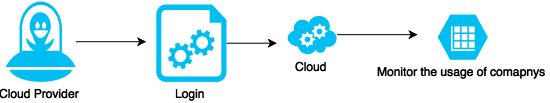
****

**Fig 3.3.2: Company Get Space**

As observed from the above figure, The Company gets the space for cloud that time we need to pay the amount for cloud providers. The company goes to the money transfer the process the company having an account and transfer the amount the amount gets the cloud provider and give cloud provide for the company. The company is using the cloud space that was used in the file and cloud space.

The cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter); the third-party auditor, who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request.

**Admin Process**



**Fig 3.3.3 Admin Process**

The admin process admin is can login and go into the process for the user how much data company used. The admin how much file size is used the company and company user. The process of the admin that only does the values and it can be used the company.

**File Upload**



**Fig 3.3.4: File Upload**

The Company enters the process and file upload and show the file for cloud, the cloud show for the file was encrypted it was working to file stored in the cloud. The Authorizes are getting data and upload the data in the database. Process file upload that time file was encrypted the file store the value in the database.

**Company Admin Process**

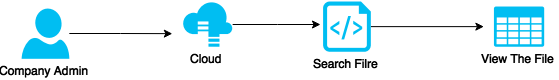


**Fig 3.3.5: Company Admin Process**

From the above figure it is understood that the Company administrates how many members buy into the cloud for us and how much of amount of data are used and how much of the data we are using. To monitor the all data usage the data usage for the details it was working it was to store the file amount and storage.

The Company gets the space for cloud that time we need to pay the amount for cloud providers. The company goes to the money transfer the process the company having an account and transfer the amount the amount gets the cloud provider and give cloud provide for the company. The company is using the cloud space that was used in the file and cloud space.

**Search for Files**

****

**Fig 3.3.6: Search for Files**

Once you are going to enter the page there is a lot of files there is search button whatever files you want just enter the file name and click submit button. It will go and fetch the file from the database and display to the user. If the file you want select and download the files whatever the files you need it.

**3.4 SYSTEM TECHNIQUE**

Step 1. We first discuss the procedure of evicting cached blocks to make room for new data. As shown in Algorithm 1, the metadata, including the last access time of blocks/inodes, the size and access counts for each block in the cache space, is queried from the database.

Step 2. Commonly, when a block is selected as a candidate for eviction, it can be locked to prevent any further accesses. However, the eviction operation consists of several sub-steps such as metadata querying, data merging, and encryption/compression, each demanding different. The main steps performed by the caching subsystem.

Step 3. Algorithm 2 shows the cache replacement logic in Copos. The data unit downloaded from the cloud is a segment that contains data blocks with correlated relationship, and its structure is detailed in Section 3.5. Segment based data transfer also implies that blocks swapped in can act as pre-fetched data to improve the hit rate of future requests.

**CHAPTER 4**

**REQUIREMENTS**

**4.1. GENERAL**

Using this requirement, our application provides high service with efficiently. Software requirements deal with defining software resource requirements and pre-requisites that need to be installed on a server that provide optimal functionality of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, the hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements. The project is concentrate on push a gathering of clients to fully utilize the volume rebate estimating methodology offered by cloud administration suppliers through cost productive.

**4.2. HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It showed what the system do and not how it should be implemented.

PROCESSOR : PENTIUM IV 2.60 GHz, Intel Dual Core.

RAM : 4 GB DD RAM

MONITOR : 15” LCD, LED MONITOR

HARD DISK : 40 GB

**4.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating costs, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

Front End : JAVA (j2ee, Servlets, JSP)

Back End : My SQL

Operating System : Windows 10s

IDE : Net Beans, Eclipse

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 GENERAL**

Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

**5.1.1 USE CASE DIAGRAM**

A use case diagram is a type of behavioral diagram created from a Use-case analysis.



**Fig 5.1.1: Use Case Diagram**

The main purpose of a use case diagram is to show what system functions are performed for user If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. The company gets the space for cloud that time we need to pay the amount for cloud providers. The company goes to the money transfer the process the company have an account and transfer the amount the amount get the cloud provider and give cloud provide for the company. The company is using the cloud space that was used in the file and cloud space. The process for the user how much data company used. File uploads to the and show the file for cloud. The Company admin how many members for the cloud for us and how much of amount of data are used and how much of the data we are using. A lot of files there are in search button whatever files you want just enter the file name and click submit button.

**5.1.2 CLASS DIAGRAM**



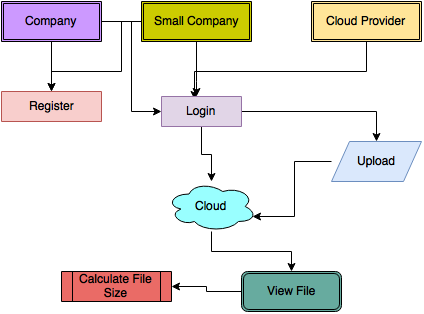
**Fig 5.1.2: Class Diagram**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code in the Diagram If the person already exits at once can login into the server else consumer need to check are their information consisting of username, password and email identification, into the server. The employer gets the space for cloud that time we want to pay the amount for cloud providers. The business enterprise visit the money transfer the manner the enterprise have an account and transfer the quantity the quantity get the cloud provider and provide cloud offer for the company. The corporation is using the cloud area that became used to the record and cloud area. The manner for the user how an awful lot records, business enterprise used. File add to the and show the document for the cloud. The employer admin how many members buy into the cloud for us and how much amount of facts are used and how much of the information we are using. Lots of files there may be seek button something documents you need just input the record call and click publish button.

**5.1.3 OBJECT DIAGRAM**

An **object diagram** in the Unified Modeling Language (UML) is a diagram that shows a complete or partial view of the structure of a modeled system at a specific time.

An Object diagram focuses on some particular set of object instances and attributes, and the links between the instances.Object diagrams are more concrete than class diagrams, and are often used to provide examples, or act as test cases for the class diagrams. Only those aspects of a model that are of current interest need be shown on an object diagram.



**Fig 5.1.3: Object Diagram**

The organization gets the distance for cloud that point we need to pay the amount for cloud providers. The enterprise organization, visit the money switch the manner the organization have an account and switch the amount the quantity get the cloud company and offer cloud provide for the company. The company are using the cloud place that have become used to the record and cloud region. The manner for the user how a lousy lot records enterprise employer used. Document add to the and show the document for the cloud. The agency admin what number of member purchase to the cloud for us and how much of amount of information are used and what kind of information we are using. Lots of files there may be are seeking for the button a few element files you need just input the record call and click on the publish button.

**5.1.4 STATE DIAGRAM**

A **state diagram** is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction.



**Fig 5.1.4: State Diagram**

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.In the Diagram If the man or woman already exits immediately can login into the server else patron want to check in their statistics which includes username, password and email identification, into the server. The organization gets the distance for cloud that point we need to pay the amount for cloud providers. The enterprise organization, visit the money switch the manner the organization have an account and switch the amount the quantity get the cloud company and offer cloud provide for the company. The company is using the cloud place that had become used to the record and cloud region. Lots of files there may be are seeking for the button a few element files you need just input the record call and click on the publish button.

**5.1.5 ACTIVITY DIAGRAM**

Activity diagram is a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.



**Fig 5.1.5: Activity Diagram**

In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. In the Diagram If the man or woman already exits immediately can login into the server else patron want to check in their statistics, which includes username, password and email identification, into the server. The organization gets the distance for cloud that point we need to pay the amount for cloud providers. The enterprise organization, visit the money switch the manner the organization have an account and switch the amount the quantity get the cloud company and offer cloud provide for the company. The company are using the cloud place that have become used to the record and cloud region. The manner for the user how a lousy lot records enterprise employer used. Document add to the and show the document for the cloud. The agency admin what number of member purchase to the cloud for us and how much of amount of information are used and what kind of information we are using. Lots of files there may be are seeking for the button a few element files you need just input the record call and click on the publish button.

**5.1.6 SEQUENCE DIAGRAM**

A sequence diagram in UML is a kind of interaction diagram that shows how the processes operate with one another and in what order. It is a construct of a message sequence chart.

Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams. The below diagram shows the sequence flow shows how the process occurs in this project. The organisation get the distance for cloud that point we need paid the amount for cloud provider

. The enterprise organization visit the money switch the manner the organization have Account and switch the amount the quantity get the cloud company and offer cloud provide for the company. The company are use the cloud place that have become used to the record and cloud region. the manner for the user how an lousy lot records enterprise employer used. document add to the and show the document for cloud. The agency admin what number of member purchase to the cloud for us and how much of amount of information are used and what kind of of information we are used.

A sequence diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, and the relationships between the classes. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class.



**Fig 5.1.6: Sequence Diagram**

In our sequence diagram specifying processes operate with one another and in order. In our sequence diagram for If the man or woman already exits immediately can login into the server else patron want to check in their statistics which includes username, password and email identification, into the server. The organisation get the distance for cloud that point we need paid the amount for cloud provider. The enterprise organization visit the money switch the manner the organization have Account and switch the amount the quantity get the cloud company and offer cloud provide for the company. The company are use the cloud place that have become used to the record and cloud region. the manner for the user how an lousy lot records enterprise employer used. document add to the and show the document for cloud. The agency admin what number of member purchase to the cloud for us and how much of amount of information are used and what kind of of information we are used. lots of files there may be are seeking for the button a few element files you need just input the record call and click on the publish button.

**5.1.7 COLLABORATION DIAGRAM**

A collaboration diagram shows the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction.



**Fig 5.1.7: Collaboration Diagram**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.In the Diagram If the man or woman already exits immediately can login into the server else patron want to check in their statistics which includes username, password and email identification, into the server. The organisation get the distance for cloud that point we need paid the amount for cloud provider. the manner for the user how an lousy lot records enterprise employer used. document add to the and show the document for cloud. The agency admin what number of member purchase to the cloud for us and how much of amount of information are used and what kind of of information we are used. lots of files there may be are seeking for button a few element files you need just input the record call and click on publish button.

**5.1.8 COMPONENT DIAGRAM**

Components are wired together by using an *assembly connector* to connect the required interface of one component with the provided interface of another component.

****

**Fig 5.1.8: Component Diagram**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and they are used to illustrate the structure of arbitrarily complex systems.In the Diagram we show for is to expose what device capabilities are performed for user can login and company also login. The person are upload file and view the files. The file are two type are store in public and private. The file also download the file. User And Company store the file for public. The user store the private file it will be show only that user and organization. The file also view online and edit the file and store the file in cloud. The user only can edit the file for the user can view the file.

**5.1.9 DATA FLOW DIAGRAM**

Level 0:

Cloud Provider

Company 1

Company 2

Company 3

Level 1:

Cloud Provider

Company 1

Company 2

Company 3

Small Company 1

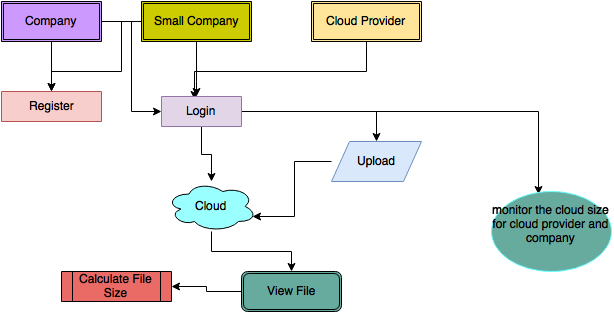
Small Company 2

Small Company 3

**Fig 5.1.9: Data Flow Diagram**

It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel. In the DFDs the level zero process is based on the login validations. What is the cloud user contained constraints send to the cloud provider. A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

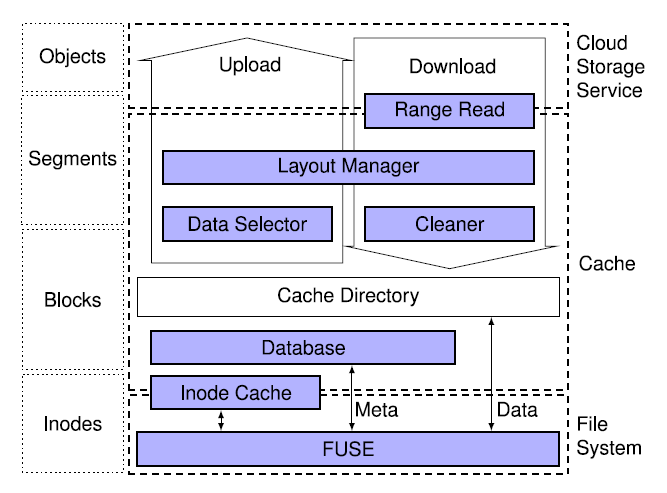
**5.1.10 E-R DIAGRAM**



**Fig 5.1.10: E-R diagram**

Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database.In the Diagram If the man or woman already exits immediately can login into the server else patron want to check in their statistics which includes username, password and email identification, into the server. The organization gets the distance for cloud that point we need to pay the amount for cloud providers. The enterprise organization, visit the money switch the manner the organization have an account and switch the amount the quantity get the cloud company and offer cloud provide for the company. The company is using the cloud place that had become used to the record and cloud region. The manner for the user how a lousy lot records enterprise employer used. Document add to the and show the document for the cloud.

**5.1.11 SYSTEM ARCHITECTURE**



**Fig 5.1.11: System Architecture**

**CHAPTER 6**

**IMPLEMENTATION**

In this we implement the coding part using eclipse. Below are the coding’s that are used to generate the domain module for Cloud Computing. Here the proposed techniques are used in the coding part to Cloud to Cloud Interaction.

**Index.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Hybrid Cloud</title>

<link href=*"css/style1.css"* rel=*"stylesheet"* type=*"text/css"*

media=*"screen"* />

</head>

<body>

<div id=*"full"*>

<div id=*"header"*>

<div id=*"logo"*>

<img alt=*"img"* src=*"images/logos.png"* height=*"150"* width=*"200"*>

</div>

<div id=*"logo1"*>

<img alt=*"img"* src=*"images/logo1.png"* height=*"130"* width=*"350"*>

</div>

<div id=*"nav"*>

<ul>

<li id=*"selected"*><a href=*"index.jsp"*>Home</a></li>

<li><a href=*"login.jsp"*>Login</a></li>

<li><a href=*"register.jsp"*>Register</a></li>

<li><a href=*"progress.jsp"*>Progress</a></li>

<li><a href=*"company.jsp"*>Company</a></li>

<li><a href=*"admin.jsp"*>Admin</a></li>

</ul>

</div>

</div>

<div id=*"section"*>

</div>

<div id=*"footer"*>

</div>

</div>

</body>

</html>

**Register.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Hybrid Cloud</title>

<link href=*"css/style1.css"* rel=*"stylesheet"* type=*"text/css"*

media=*"screen"* />

</head>

<body>

<div id=*"full"*>

<div id=*"header"*>

<div id=*"logo"*>

<img alt=*"img"* src=*"images/logos.png"* height=*"150"* width=*"200"*>

</div>

<div id=*"logo1"*>

<img alt=*"img"* src=*"images/logo1.png"* height=*"130"* width=*"350"*>

</div>

<div id=*"nav"*>

<ul>

<li><a href=*"index.jsp"*>Home</a></li>

<li><a href=*"login.jsp"*>Login</a></li>

<li id=*"selected"*><a href=*"register.jsp"*>Register</a></li>

<li><a href=*"progress.jsp"*>Progress</a></li>

<li><a href=*"company.jsp"*>Company</a></li>

<li><a href=*"admin.jsp"*>Admin</a></li>

</ul>

</div>

</div>

<div id=*"section"*>

<div class=*"content2"*>

<div class=*"grids-heading gallery-heading signup-heading"*>

<h2>Register</h2>

</div>

<form action=*"Register"* method=*"post"*>

<input type=*"text"* name=*"Company"* placeholder=*"Company"* >

<input type=*"email"* name=*"Email"* placeholder=*"Email"*>

<input type=*"password"* name=*"Password"* placeholder=*"Password"* >

<input type=*"text"* name=*"cloud"* value=*"Hybrid"* readonly=*"readonly"*>

<input type=*"tel"* name=*"Phone"* placeholder=*"Phone"* >

<input type=*"submit"* class=*"register"* value=*"Register"*>

</form>

<a href=*"login.jsp"*>Goto login</a>

</div>

</div>

<div id=*"footer"*>

</div>

</div>

</body></html>

Register.java

Package servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import bean.RegisterBean;

import implement.implementation;

import interfaces.interfaces;

/\*\* \* Servlet implementation class Register\*/

@WebServlet("/Register")

public class Register extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\* @see HttpServlet#HttpServlet() \*/

public Register() {

super();

// TODO Auto-generated constructor stub

}

/\*\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response) \*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*\*@see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stubdoGet(request, response);

String Company = request.getParameter("Company");

String Email = request.getParameter("Email");

String Password = request.getParameter("Password");

String Cloud = request.getParameter("cloud");

String Phone = request.getParameter("Phone");

RegisterBean rb = new RegisterBean();rb.setCompany(Company);rb.setEmail(Email);

rb.setPassword(Password);

rb.setCloud(Cloud);

rb.setPhone(Phone);

interfaces ifs = new implementation();

int i = ifs.register(rb);

if (i == 1) {

response.sendRedirect("login.jsp");

} else {

response.sendRedirect("error.jsp");

}}}

**Login.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Hybrid Cloud</title>

<link href=*"css/style1.css"* rel=*"stylesheet"* type=*"text/css"*

media=*"screen"* />

</head>

<body>

<div id=*"full"*>

<div id=*"header"*>

<div id=*"logo"*>

<img alt=*"img"* src=*"images/logos.png"* height=*"150"* width=*"200"*>

</div>

<div id=*"logo1"*>

<img alt=*"img"* src=*"images/logo1.png"* height=*"130"* width=*"350"*>

</div>

<div id=*"nav"*>

<ul>

<li><a href=*"index.jsp"*>Home</a></li>

<li id=*"selected"*><a href=*"login.jsp"*>Login</a></li>

<li><a href=*"register.jsp"*>Register</a></li>

<li><a href=*"progress.jsp"*>Progress</a></li>

<li><a href=*"company.jsp"*>Company</a></li>

<li><a href=*"admin.jsp"*>Admin</a></li>

</ul>

</div>

</div>

<div id=*"section"*>

<div class=*"content2"*>

<div class=*"grids-heading gallery-heading signup-heading"*>

<h2>Sign Up</h2>

</div>

<form action=*"Login"* method=*"post"*>

<input type=*"text"* name=*"Company"* placeholder=*"Company"* >

<input type=*"email"* name=*"Email"* placeholder=*"Email"*>

<input type=*"password"* name=*"Password"* placeholder=*"Password"* >

<input type=*"submit"* class=*"register"* value=*"Sign Up"*>

</form>

<a href=*"register.jsp"*>Register Here</a>

</div>

</div>

<div id=*"footer"*>

</div>

</div>

</body>

</html>

**Login.java**

Package servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import bean.RegisterBean;

import implement.implementation;

import interfaces.interfaces;

/\*\*\* Servlet implementation class Login \*/

@WebServlet("/Login")

Public class Login extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*\* @see HttpServlet#HttpServlet()\*

public Login() {

super();

// TODO Auto-generated constructor stub

}

/\*\*\*@see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response) \*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

String Company=request.getParameter("Company");

HttpSession session = request.getSession();

session.setAttribute("Company", Company);

String Email=request.getParameter("Email");

String Password=request.getParameter("Password");

RegisterBean rgb=new RegisterBean();

rgb.setCompany(Company);

rgb.setEmail(Email);

rgb.setPassword(Password);

interfaces ifs=new implementation();

int i=ifs.login(rgb);

if(i==1){

response.sendRedirect("progress1.jsp");

}

else{

response.sendRedirect("error.jsp");

}}}

**Process.jsp**

<%@page import=*"database.\*"*%>

<%@page import=*"java.sql.\*"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Hybrid Cloud</title>

<link href=*"css/style1.css"* rel=*"stylesheet"* type=*"text/css"*

media=*"screen"* />

<script type=*"text/javascript"*>

**function** num()

{

**var** data=document.myform.data.value;

**if** (data<300){

alert("Please Enter minimum 300");

**return** **false**;

}

}

</script>

<style type=*"text/css"*>

*.animatedButton2*{

text-decoration: *none !important*;

background-color:*#a30e12*;

padding-left: *40px*;

padding-right: *40px*;

height: *38px*;

line-height: *38px !important*;

display: *inline-block*;

border: *1px solid #450003*;

text-shadow:*0px 1px 1px #000*;

-webkit-box-shadow: *inset 0px 1px 1px #fff, 0 5px 0px 0px #450003*;

-moz-box-shadow: *inset 0px 1px 1px #fff, 0 5px 0px 0px #450003*;

box-shadow: *inset 0px 1px 1px #fff, 0 5px 0px 0px #450003*;

-webkit-border-radius: *5px*;

-moz-border-radius: *5px*;

border-radius: *5px*;

-webkit-transition: *all 0.15s linear*;

-moz-transition: *all 0.15s linear*;

-o-transition: *all 0.15s linear*;

}

*.animatedButtonText2*{

font-family: *Oswald, Arial, sans-serif*;

line-height: *38px !important*;

font-size: *18px*;

color: *#EAEAEA*;

text-align:*center*;

-webkit-transition: *all 0.15s linear*;

-moz-transition: *all 0.15s linear*;

-o-transition: *all 0.15s linear*;

transition: *all 0.15s linear*;

}

*.animatedButton2:hover*{

background-color:*#880408*;

margin-top:*5px*;

-webkit-box-shadow: *inset 0px 1px 1px #000, 0 0px 0px 0px #000*;

-moz-box-shadow: *inset 0px 1px 1px #000, 0 0px 0px 0px #000*;

box-shadow: *inset 0px 1px 1px #000, 0 0px 0px 0px #000*;

}

*.animatedButton2:hover* *.animatedButtonText2*{

text-shadow:*0px -1px 1px #000*;

color:*#b2161a*;

}

</style>

</head>

<body>

<div id=*"full"*>

<div id=*"header"*>

<div id=*"logo"*>

<img alt=*"img"* src=*"images/logos.png"* height=*"150"* width=*"200"*>

</div>

<div id=*"logo1"*>

<img alt=*"img"* src=*"images/logo1.png"* height=*"130"* width=*"350"*>

</div>

<div id=*"nav"*>

<ul>

<li><a href=*"index.jsp"*>Home</a></li>

<li><a href=*"login.jsp"*>Login</a></li>

<li><a href=*"register.jsp"*>Register</a></li>

<li id=*"selected"*><a href=*"progress.jsp"*>Progress</a></li>

<li><a href=*"company.jsp"*>Company</a></li>

<li><a href=*"admin.jsp"*>Admin</a></li>

</ul>

</div>

</div>

<div id=*"section"*>

<%String Company = session.getAttribute("Company").toString();System.out.println(Company);

Connection con;PreparedStatement ps;ResultSet rs;String Cloud="";

con=Connections.con();String query="SELECT cloud FROM `itjcc29`.`hybridreg` where Company='"+Company+"'";ps=con.prepareStatement(query);rs=ps.executeQuery();

**if**(rs.next()){

Cloud=rs.getString(1);

}

%>

<div id=*"sec1"*>

<div id=*"welcome"*>

<h2 style="font-family:*times new roman*; font-style:*italic*; color: *#9f0ccc*">

<%out.print("Welcome To ");%>

<% out.print(Cloud+" Cloud ");%>

<%=request.getSession().getAttribute("Company")%>

</h2>

</div>

<div id=*"log"*>

<button style="margin-top: *30px*;margin-left: *60px*;background: *transparent*; border-color: *transparent*; border-radius:*20px*; ">

<a href=*"index.jsp"* style="text-decoration: *none*;margin-top: *20px*; color: *fuchsia*; font-size: *16px*;">Logout</a>

</button>

</div>

</div>

<div id=*"sec3"*>

<div id=*"bt1"*>

<a href=*"getspace.jsp"* class=*"animatedButton2"* >

<span class=*"animatedButtonText2"*>Get Data Space</span></a>

</div>

<div id=*"bt2"*>

<a href=*"admindetail.jsp"* class=*"animatedButton2"*>

<span class=*"animatedButtonText2"*>&nbsp;Admin Process&nbsp;</span></a>

</div>

</div>

<div id=*"sec1"*>

<div id=*"bt1"*>

<a href=*"file.jsp"* class=*"animatedButton2"* >

<spanclass=*"animatedButtonText2"*>&nbsp;&nbsp;&nbsp;FileContent&nbsp;&nbsp;&nbsp>;

</span>

</div>

<div id=*"bt2"*>

<a href=*"morespace.jsp"* class=*"animatedButton2"*>

<span class=*"animatedButtonText2"*>Get More Space</span></a>

</div>

</div>

<div id=*"sec1"*>

<div id=*"bt1"* style="margin-left: *360px*;">

<a href=*"datarequest.jsp"* class=*"animatedButton2"* >

<spanclass=*"animatedButtonText2"*>&nbsp;&nbsp;&nbsp;DataRequest&nbsp;&nbsp;&nbsp;

</span></a>

</div>

</div>

</div>

<div id=*"footer"*>

</div>

</div>

</body>

</html>

**GetSpace.jsp**

<%@page import=*"database.\*"*%>

<%@page import=*"java.sql.\*"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Hybrid Cloud</title>

<link href=*"css/style1.css"* rel=*"stylesheet"* type=*"text/css"*media=*"screen"* />

<script type=*"text/javascript"*>

**function** num()

{

**var** data=document.myform.data.value;

**if** (data<300){

alert("Please Enter minimum 300");

**return** **false**;

}

}

</script>

<style type=*"text/css"*>

*.number*{

padding: *10px*;

width: *90%*;

border: *1px solid black*;

color: *#000*;

text-align: *left*;

outline: *none*;

font-size: *1em*;

margin-top: *5%*;

}

</style>

</head>

<body>

<div id=*"full"*>

<div id=*"header"*>

<div id=*"logo"*>

<img alt=*"img"* src=*"images/logos.png"* height=*"150"* width=*"200"*>

</div>

<div id=*"logo1"*>

<img alt=*"img"* src=*"images/logo1.png"* height=*"130"* width=*"350"*>

</div>

<div id=*"nav"*>

<ul>

<li><a href=*"index.jsp"*>Home</a></li>

<li><a href=*"login.jsp"*>Login</a></li>

<li><a href=*"register.jsp"*>Register</a></li>

<li id=*"selected"*><a href=*"progress.jsp"*>Progress</a></li>

<li><a href=*"company.jsp"*>Company</a></li>

<li><a href=*"admin.jsp"*>Admin</a></li>

</ul>

</div>

</div>

<div id=*"section"*>

<%String Company=session.getAttribute("Company").toString();System.out.println(Company);%>

<div id=*"sec1"*>

<div id=*"welcome"*>

<h2 style="font-family:*times new roman*; font-style:*italic*; color: *#9f0ccc*">

<%-- <%out.print("Welcome To Hybrid Cloud ");%>

<%=request.getSession().getAttribute("Company")%>--%>

<%String Cloud="";String Account1="";Connection con;PreparedStatement ps,ps1;

con=Connections.con();String query="SELECT cloud FROM `itjcc29`.`hybridreg` where company='"+Company+"'";ps=con.prepareStatement(query);ResultSet rs=ps.executeQuery();

**if**(rs.next()){

Cloud=rs.getString(1);

}

String query1="SELECT Account FROM `itjcc29`.`bankregister` where Username='"+Cloud+"'";ps1=con.prepareStatement(query1);ResultSet rs1=ps1.executeQuery();

**if**(rs1.next()){

Account1=rs1.getString(1);HttpSession sess = request.getSession();

session.setAttribute("Account1", Account1);

}

%>

</h2>

</div>

<div id=*"log"*>

<button style="margin-top: *30px*;margin-left: *60px*;background: *transparent*; border-color: *transparent*; border-radius:*20px*; ">

<a href=*"index.jsp"* style="text-decoration: *none*;margin-top: *20px*; color: *fuchsia*; font-size: *16px*;">Logout</a>

</button>

</div>

</div>

<div id=*"sec2"*>

<div class=*"content2"*>

<div class=*"grids-heading gallery-heading signup-heading"*>

<h2>Cloud Data Space</h2>

</div>

<form action=*"banklogin.jsp"* method=*"post"* name=*"myform"*>

<input type=*"number"* class=*"number"* name=*"data"* onchange=*"num()"* placeholder=*"minimum value is 300"* required=*"required"*/>

<input type=*"text"* class=*"number"* name=*"Account"* value=*"*<%=Account1 %>*"*

readonly=*"readonly"*/>

<input type=*"submit"* class=*"register"* value=*"Make a payment"*>

</form>

</div>

</div>

</div>

<div id=*"footer"*>

</div>

</div>

</body>

</html>

**Datarequest.java**

Package servlet;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import bean.DataRequestBean;

import implement.implementation;

import interfaces.interfaces;

/\*\*\* Servlet implementation class DataRequest\*/

@WebServlet("/DataRequest")

public class DataRequest extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\* \* @see HttpServlet#HttpServlet() \*/

public DataRequest() {

super();

// TODO Auto-generated constructor stub

}

/\*\*\*@seeHttpServlet#doGet(HttpServletRequestrequest,HttpServletResponse \*response)\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException

{

// TODO Auto-generated method stubresponse.getWriter().append("Served at").append(request.getContextPath());

String Company = request.getSession().getAttribute("Company").toString();

String Company1 = request.getParameter("Company1");

System.out.println("Company="+Company);

System.out.println("Company1111111="+Company1);

String Data = request.getParameter("Data");

System.out.println("Dataaaaaaaaaaaa="+Data);

String Amount = request.getParameter("Amount");

System.out.println("Amount="+Amount);

DataRequestBean drb = new DataRequestBean();

drb.setCompany(Company);

drb.setCompany1(Company1);

drb.setData(Data);

drb.setAmount(Amount);

interfaces ifs = new implementation();

int i = ifs.datarequesst(drb);

if (i == 1) {

response.sendRedirect("dataresponse.jsp");

}

Else

{

Response.sendRedirect("error.jsp");

}

}

/\*\*\*@seeHttpServlet#doPost(HttpServletRequestrequest,HttpServletResponse \*response) \*/

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}}

**Fileupload.jsp**

<%@page import=*"database.\*"*%>

<%@page import=*"java.sql.\*"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Hybrid Cloud</title>

<link href=*"css/style1.css"* rel=*"stylesheet"* type=*"text/css"*

media=*"screen"* />

<script type=*"text/javascript"*>

**function** num()

{

**var** data=document.myform.data.value;

**if** (data<300){

alert("Please Enter minimum 300");

**return** **false**;

}

}

</script>

<style type=*"text/css"*>

*.animatedButton2*{

text-decoration: *none !important*;

background-color:*#a30e12*;

padding-left: *40px*;

padding-right: *40px*;

height: *38px*;

line-height: *38px !important*;

display: *inline-block*;

border: *1px solid #450003*;

text-shadow:*0px 1px 1px #000*;-webkit-box-shadow: *inset 0px 1px 1px #fff, 0 5px 0px 0px #450003*; -moz-box-shadow: *inset 0px 1px 1px #fff, 0 5px 0px 0px #450003*; box-shadow: *inset 0px 1px 1px #fff, 0 5px 0px 0px #450003*;

-Webkit-border-radius: *5px*;

-moz-border-radius: *5px*;

border-radius: *5px*;

-webkit-transition: *all 0.15s linear*;

-moz-transition: *all 0.15s linear*;

-o-transition: *all 0.15s linear*;

}

*.animatedButtonText2*{

font-family: *Oswald, Arial, sans-serif*;

line-height: *38px !important*;

font-size: *18px*;

color: *#EAEAEA*;

text-align:*center*;

-webkit-transition: *all 0.15s linear*;

-moz-transition: *all 0.15s linear*;

-o-transition: *all 0.15s linear*;

transition: *all 0.15s linear*;

}

*.animatedButton2:hover*

{

background-color:*#880408*;

margin-top:*5px*;-webkit-box-shadow: *inset 0px 1px 1px #000, 0 0px 0px 0px #000*;-moz-box-shadow: *inset 0px 1px 1px #000, 0 0px 0px 0px #000* box-shadow: *inset 0px 1px 1px #000, 0 0px 0px 0px #000*;

}*.animatedButton2:hover* *.animatedButtonText2*

{

text-shadow:*0px -1px 1px #000*;

color:*#b2161a*;

}

</style>

</head>

<body>

<div id=*"full"*>

<div id=*"header"*>

<div id=*"logo"*>

<img alt=*"img"* src=*"images/logos.png"* height=*"150"* width=*"200"*>

</div>

<div id=*"logo1"*>

<img alt=*"img"* src=*"images/logo1.png"* height=*"130"* width=*"350"*>

</div>

<div id=*"nav"*>

<ul>

<li><a href=*"index.jsp"*>Home</a></li>

<li><a href=*"login.jsp"*>Login</a></li>

<li><a href=*"register.jsp"*>Register</a></li>

<li id=*"selected"*><a href=*"progress.jsp"*>Progress</a></li>

<li><a href=*"company.jsp"*>Company</a></li>

<li><a href=*"admin.jsp"*>Admin</a></li>

</ul>

</div>

</div>

<div id=*"section"*>

<%String Company = session.getAttribute("Company").toString();

System.out.println(Company);

Connection con;

PreparedStatement ps;

ResultSet rs;

String Cloud="";con=Connections.con();

String query="SELECT cloud FROM `itjcc29`.`hybridreg` where Company='"+Company+"'";

ps=con.prepareStatement(query);

rs=ps.executeQuery();

**if**(rs.next()){

Cloud=rs.getString(1);

}

%>

<div id=*"sec1"*>

<div id=*"welcome"*>

<h2 style="font-family:*times new roman*; font-style:*italic*; color: *#9f0ccc*">

<%out.print("Welcome To ");%>

<% out.print(Cloud+" Cloud ");%><%=request.getSession().getAttribute("Company")%>

</h2>

</div>

<div id=*"log"*>

<button style="margin-top: *30px*;margin-left: *60px*;background: *transparent*; border-color: *transparent*; border-radius:*20px*; ">

<a href=*"index.jsp"* style="text-decoration: *none*;margin-top: *20px*; color: *fuchsia*; font-size: *16px*;">Logout</a>

</button>

</div>

</div>

<div id=*"sec2"*>

<div class=*"content2"*>

<div class=*"grids-heading gallery-heading signup-heading"*>

<h2>File Upload</h2>

</div>

<form action=*"Upload"* method=*"post"* enctype=*"multipart/form-data"*>

<input type=*"text"* name=*"Company"* value=*"*<%=Company %>*"* readonly=*"readonly"*>

<input type=*"file"* name=*"File"*>

<input type=*"submit"* class=*"register"* value=*"Upload"*>

</form>

</div>

</div>

</div>

<div id=*"footer"*>

</div>

</div>

</body></html>

Fileupload.java

package servlet;

import java.io.File;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.oreilly.servlet.multipart.FilePart;

import com.oreilly.servlet.multipart.MultipartParser;

import com.oreilly.servlet.multipart.ParamPart;

import com.oreilly.servlet.multipart.Part;

import bean.UploadBean;

import database.Connections;

import implement.implementation;

import interfaces.interfaces;

/\*\*

\* Servlet implementation class Upload

\*/

@WebServlet("/Upload")

public class Upload extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public Upload() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

PreparedStatement ps, ps1;

ResultSet rs;

Connection con;

Part part;

FilePart filepart;

ParamPart parampart;

String filename = null;

String title = "";

String filetype = "";

long size=0;

String Company1="";

String Company="";

long Totalspace=0;

long usages=0;

long full=0;

String realpath = getServletContext().getRealPath("/");

System.out.println("realpath="+realpath);

String Originalpath = realpath.substring(0, realpath.indexOf("."))

+ "HybridCloud\\WebContent\\local\\";

System.out.println("original"+Originalpath);

ArrayList<String> paramValues = new ArrayList<String>();

MultipartParser mpp = new MultipartParser(request, 999999999);

while ((part = mpp.readNextPart()) != null) {

if (part.isParam()) {

System.out.println("comming....");

parampart = (ParamPart) part;

title = parampart.getStringValue();

System.out.println("title" + title);

paramValues.add(title);

}

else if (part.isFile()) {

try {

Company=paramValues.get(0);

System.out.println("Company="+Company);

con = Connections.con();

String query = "SELECT company,totalspace,usages FROM `itjcc29`.`hybridmain` where company='"+Company+"'";

ps = con.prepareStatement(query);

rs = ps.executeQuery();

rs.next();

Company1=rs.getStringSystem.out.println("Company1="+Company1);

Totalspace=rs.getLong(2);

System.out.println("Totalspace="+Totalspace);

usages=rs.getLong(3);

System.out.println("usages="+usages);

filepart = (FilePart) part;

filename = filepart.getFileName();

System.out.println("filename"+filename);

filetype = filepart.getContentType();

System.out.println("filetype"+filetype);

Originalpath = Originalpath + filename;

System.out.println("Originalpath"+Originalpath);

File file = new File(Originalpath);

System.out.println("file"+file);

size = filepart.writeTo(file);

System.out.println("long size"+size);

}

catch (Exception e) {

// TODO Auto-generated catch blocke.printStackTrace();

}

}

}

System.out.println("Company.equalsIgnoreCase(Company1)"+Company.equalsIgnoreCase(Company1));

if(Company.equalsIgnoreCase(Company1)){

System.out.println("Cominggggggggggg111111111");

full=usages+size;

System.out.println("Full="+full);

if(Totalspace>=full)

System.out.println(paramValues);

UploadBean ub=new UploadBean();

ub.setCompany(paramValues.get(0));

ub.setFilename(filename);

ub.setFiletype(filetype);

ub.setFilesize(size);

interfaces ifs=new implementation();

int i=ifs.upload(ub);

System.out.println("iiiiiiiiiiiiiii="+i);

if (i==1) {

response.sendRedirect("uploadsuccess.jsp");

}

else

{

response.sendRedirect("error.jsp");

}

}

else

{

response.sendRedirect("error.jsp");

}

}

else

{

response.sendRedirect("error.jsp");

}

}

}

**Proposed System Model Explanation**

In this project our proposed system presents we present the design, implementation, and evaluation of Copos, a cloud based file system that strikes a balance between performance and monetary cost Here we present the design, implementation, and evaluation of Copos, a cloud based file system that strikes a balance between performance and monetary cost.Storage objects are not stored in a local cache. A Prudent storage model optimized for scenarios concerning multiple cloud storage services. Similar to local hierarchical storage systems, FCFS integrates cloud services with very different structures. By dynamically adapting the storage volume sizes of each service. The design and implementation of a cost-effective file system based on the cloud storage. A Prudent storage model optimized for scenarios concerning multiple cloud storage services.

**ADVANTAGES:**

The design and implementation of a cost effective file system based on the cloud storage.

A Prudent storage model optimized for scenarios concerning multiple cloud storage services.

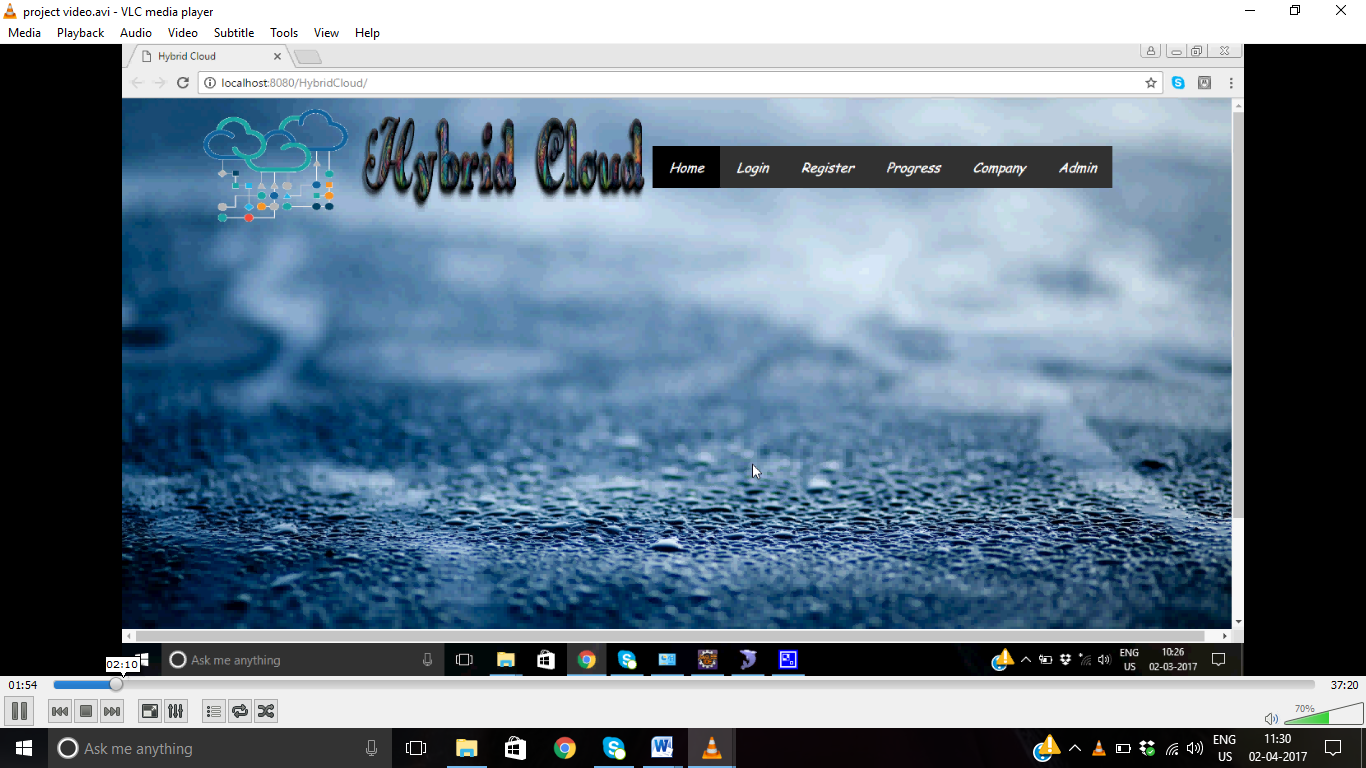
**CHAPTER 7**

**RESULTS**

**7.1 GENERAL**

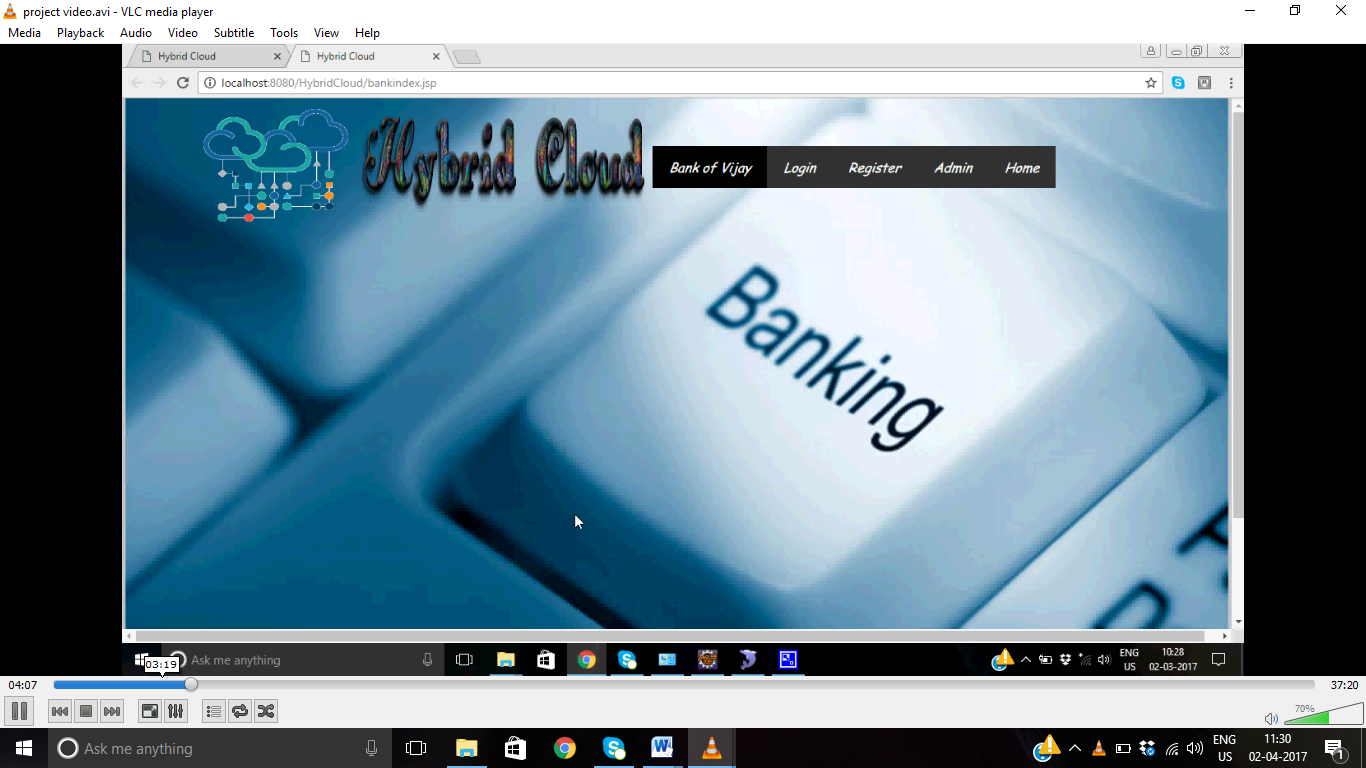
The snapshot is nothing but every moment of the application while running. It gives the clear, elaborated view of the application. It will be useful for the new user to understand for the future steps.

**7.2. VARIOUS SNAPSHOTS**

****

**Fig 7.2.1: Login Page**

This page opens, as the user requests and if already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. The Server will create the account for the entire user to maintain upload and download rate. The name will be set as user id. Logging is usually used to enter to a specific page. . It will be useful for the new user to understand for the future steps.

****

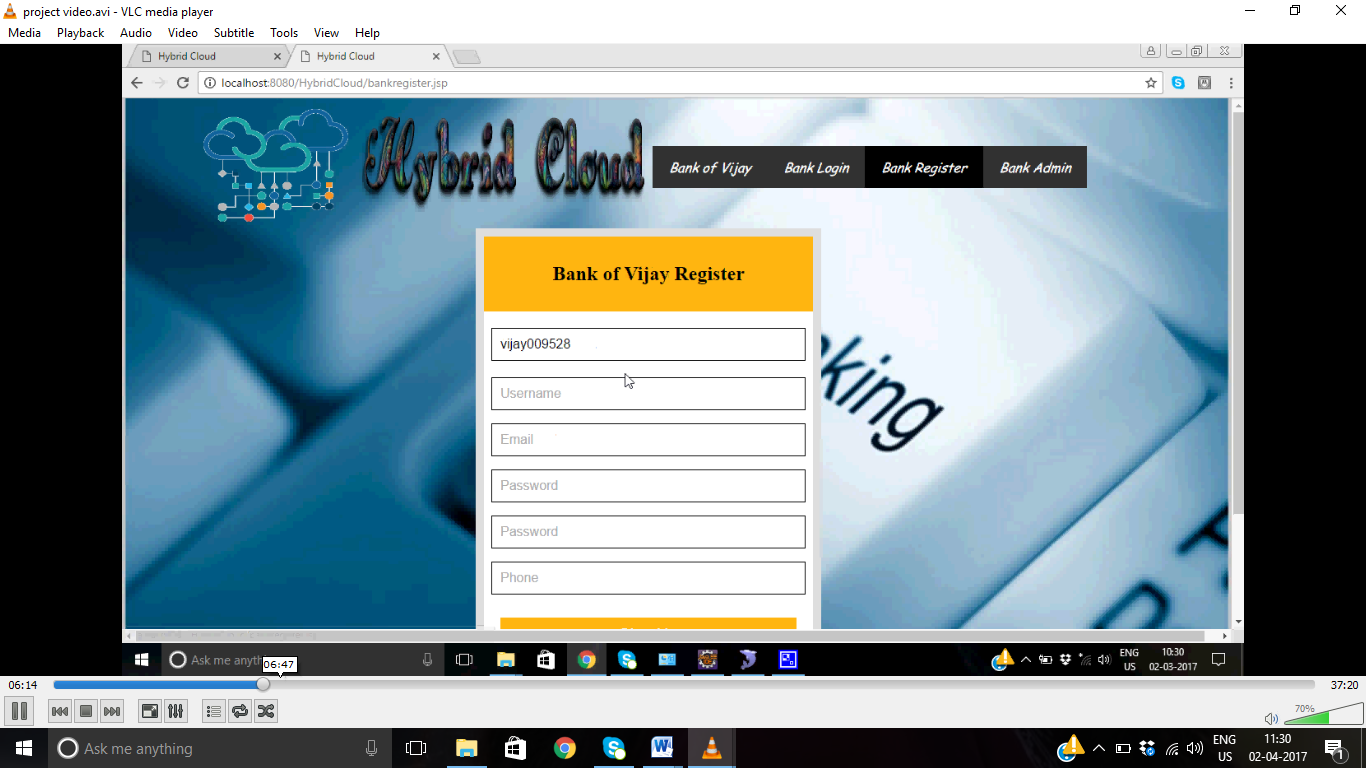
**Fig 7.2.2: Admin Process**

This page opens when the accounts to be created for both large and small companies. Moreover, the admin is requested to proceed request for getting space. Admin can login and go into the process for the user how much data company used. The admin how much file size is used the company and company user.

The process of the admin that only does the values and it can be used the company. The admin process admin is can login and go to the process for the user how much data company used. The admin how much file size are used the company and company user.

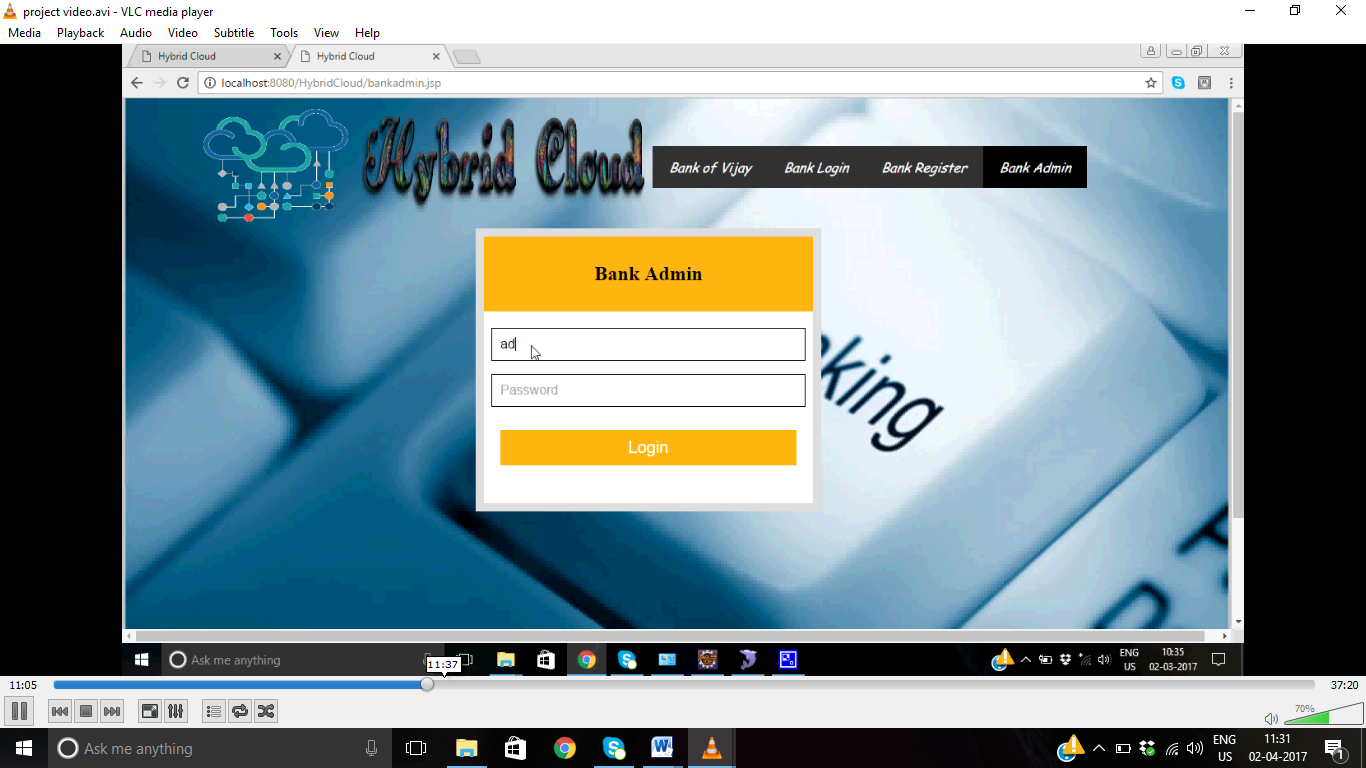
The process for the admin that only do the values and it can be used the company. The user need to enter exact Username and password which is given in the registration, if login success means it will take up to main page else it will remain in the login page itself. If it is a new user then it will move to the registration page.

Admin Can Fix the rate to some huge amount of cloud space that amount is convenient to buy for Cloud brokers and they can use that cloud space to sell with some convenient cost.

****

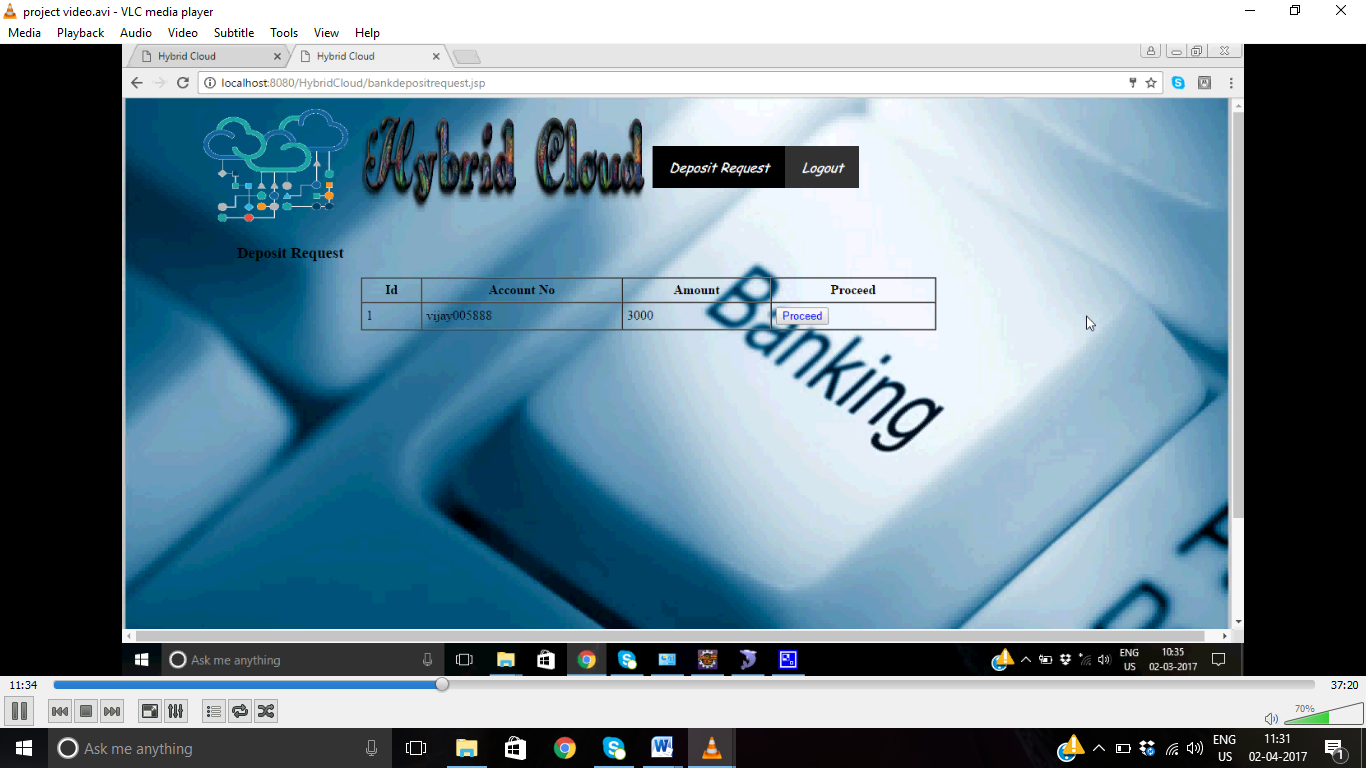
**Fig 7.2.3: Registration process**

In this case, the small company and the large company need to register in order to perform transactions like deposit, transfer etc. Registration page contains the following credentials like Username, Email, Password, Phone. After filling all the credentials we can login into the bank account.

****

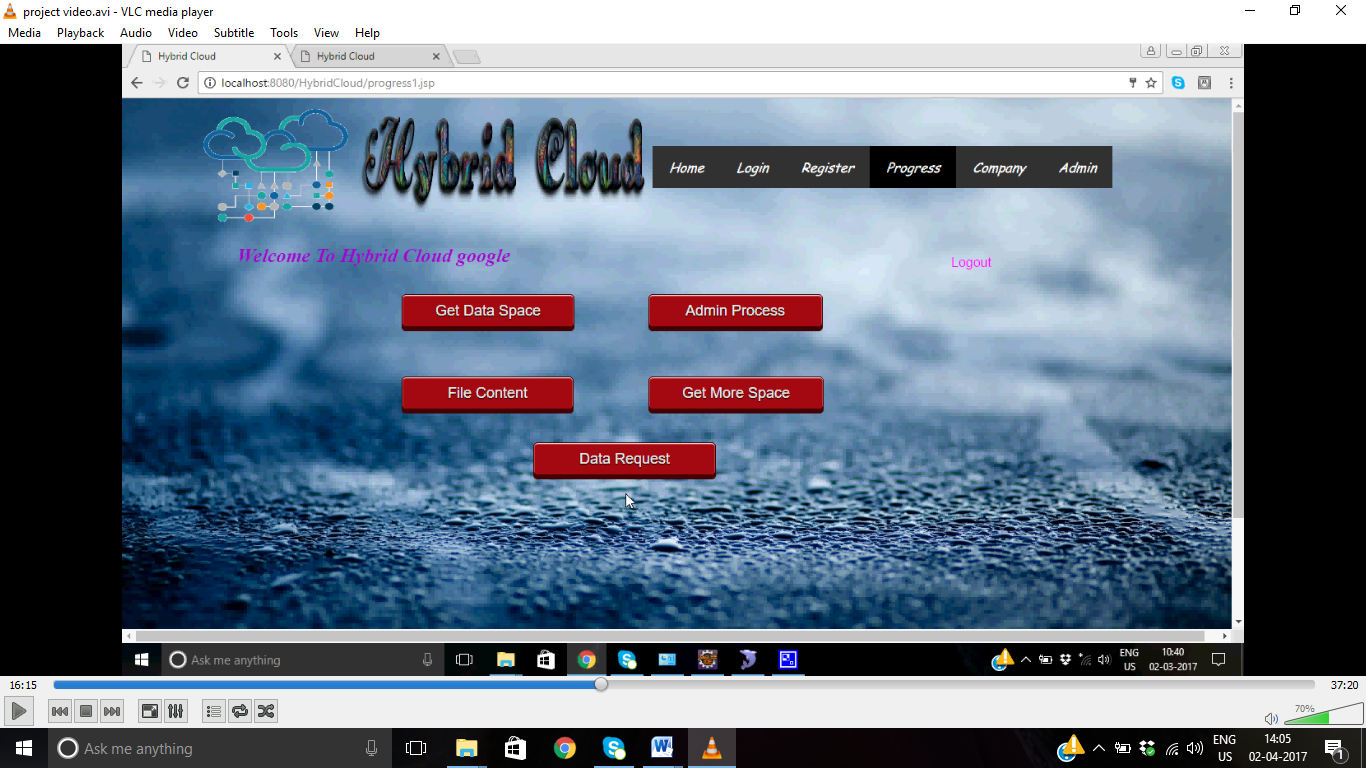
**Fig 7.2.4: Admin Login**

From this it can be observed that admin also requires Userid and password to access the bank account, where the requests are made by the small and large companies. Admin can login and go into the process for the user how much data company used. The admin how much file size is used the company and company user.

****

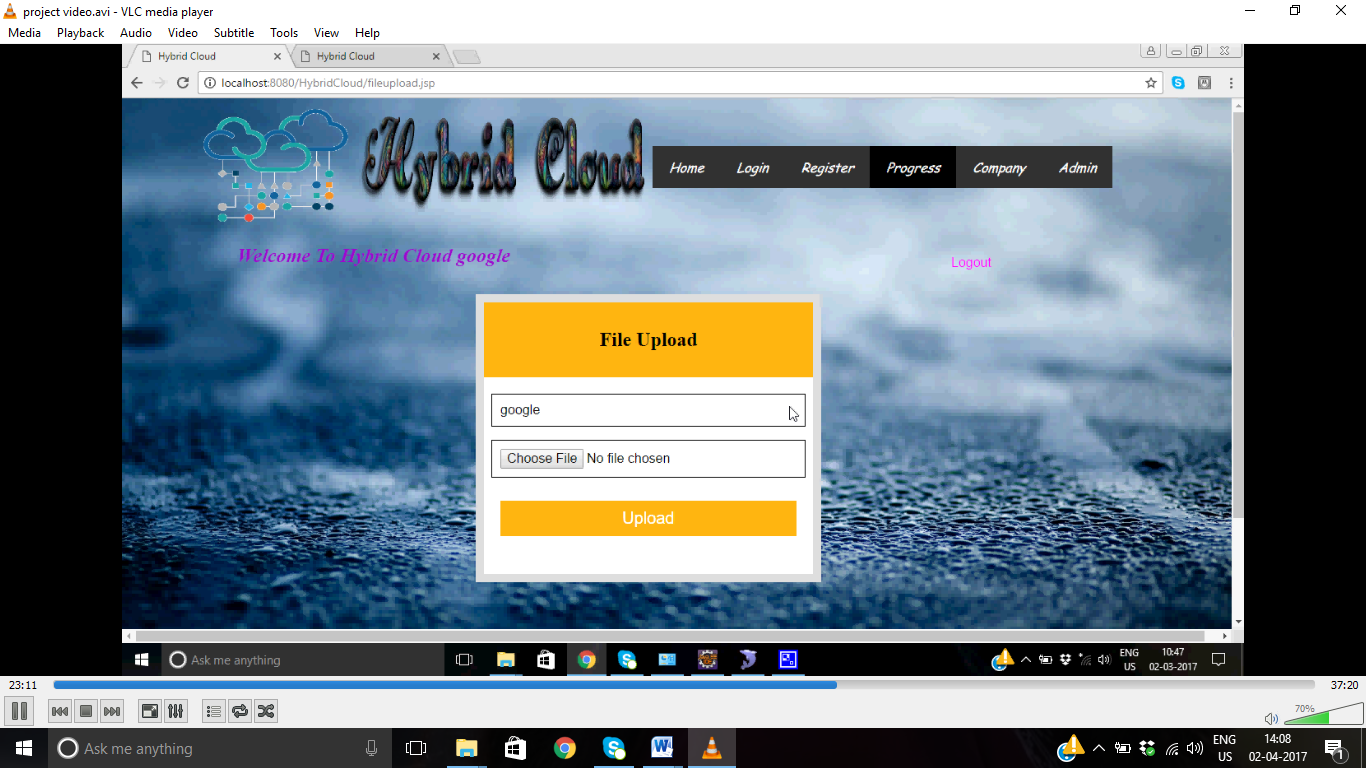
**Fig 7.2.5: Admin Process**

Once, the admin logins into the account where the requests are made by the companies then, the decision regarding allotting space is done. Admin administrates how many members buy into the cloud for usage and how much of amount of data are used and how much of the data we are using. To monitor the all data usage the data usage for the details it was working it was to store the file amount and storage.

****

**Fig 7.2.6: Company Home page**

The Company Home page contains the following attributes like Get Data Space, Admin Process, File Content, Get More Space, Data Request.The Company gets the space for cloud that time we need to pay the amount for cloud providers.

****

**Fig 7.2.7: Uploading Files into Cloud**

Finally,The Company enters the process and file upload and show the file for cloud, the cloud show for the file was encrypted it was working to file stored in the cloud. The Authorizes are getting data and upload the data in the database. Process file upload that time file was encrypted the file store the value in the database.

**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3 TYPES OF TESTS**

**8.3.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**8.3.2 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**8.3.3 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 Integration Test**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**8.3.6 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance testing for Data Synchronization:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node.
* The Route adds operation is done only when there is a Route request in need
* The Status of nodes information is done automatically in the Cache Updating process

**8.3.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identifty the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**8.3.8 Test Cases**

**Case 1**

|  |
| --- |
| Test case1:User Interface Design. |
| Test objective: Checking the authorized user. |
| Test Description: Providing the authorized to continue or the next process. |
| Requirements verified: Yes |
| Test Environment: The test environment includes J2EE and MYSQL. |
| Pre-Conditions: Databases should be available. |
| Problems/Issues: NIL |
| Notes: Successfully executed. |

This case represents the authenticate.

**Case2**

|  |
| --- |
| Test case2:Company Get cloud space |
| Test objective: allocating the cloud space. |
| Test Description: allocating required amount cloud space for the user |
| Requirements verified: Yes |
| Test Environment: The test environment includes J2EE and MYSQL. |
| Pre-Conditions: Databases should be available. |
| Problems/Issues: NIL |
| Notes: Successfully executed. |

**Case3**

|  |
| --- |
| Test case 3: Admin Process. |
| Test objective: Cloud Admin will check the company space requested. |
| Test Description: Fixing the rate for the amount of cloud provided to the company. |
| Requirements verified: Yes |
| Test Environment: The test environment includes J2EE and MYSQL. |
| Pre-Conditions: Databases should be available. |
| Problems/Issues: NIL |
| Notes: Successfully executed. |

**Case 4**

|  |
| --- |
| Test case4: File Upload. |
| Test objective: To upload the Files into allotted cloud space. |
| Test Description: allocating the required amount of data for the cloud space provided for the companies. |
| Requirements verified: Yes |
| Test Environment: The test environment includes J2EE and MYSQL. |
| Pre-Conditions: Databases should be available. |
| Problems/Issues: NIL |
| Notes: Successfully executed. |

**Case 5**

|  |
| --- |
| Test case 5: Company Admin Process. |
| Test objective: Large Company Admins the Small Company. |
| Test Description: It views the user data given to the cloud service provider. |
| Requirements verified: Yes |
| Test Environment: The test environment includes visual studio and MYSQL. |
| Pre-Conditions: Databases should be available. |
| Problems/Issues: NIL |
| Notes: Successfully executed. |

**CHAPTER 9**

**CONCLUSION AND FUTURE ENHANCEMENT**

**CONCLUSION**

This project presents the design, implementation and evaluation of Copos, a cloud based file system specifically designed for cloud environments in which improving performance and monetary cost are both principally important for end users. With the efficient data structures and algorithmic designs, it achieves our goals of high performance and cost-effective. In the future, we plan to investigate new ways to further reduce the storage cost. For example, using byte-addressable compression algorithms, we can precisely control how much data the client needs to download instead of fetching a complete segment each time.

**FUTURE ENHANCEMENT**

In Future precisely control how much data the client needs to download instead of fetching a complete segment each time.Investigate new ways to further reduce the storage cost. Considering Itmay concurrently handle multiple storage sessions from different users for their outsourced data files, as a future work, we further extend our privacy-preserving public auditing protocol into a multiuser setting, where the Itcan perform multiple auditing tasks in a batch manner for better efficiency.