3-2017

Security and Data De-Duplication Using Hybrid Cloud Technology

Mahender Korre
St Cloud State University, mkorre@stcloudstate.edu

Security and Data De-Duplication Using Hybrid Cloud Technology

by

Mahender Korre

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree

Master of Science

in Information Assurance

March, 2017

Starred Paper Committee:
Susantha Herath, Chairperson
Dennis Guster
Sneh Kalia

Abstract

Data de-duplication is a method used to compress information aiding in the removal of duplicate copies of information. It has been effective in cloud storage; it decreases the required storage space to secure such data by considering de-duplication this concurrent method has been projected.

Making sure that a company secures its data is very important. As such, this paper formally addresses the approval of data de-duplication. While it is similar to the common customary de-duplication, data de-duplication takes into consideration the different benefits of customers. In the same way, introducing fewer new copy checks for engineers could increase new de-duplication development supporting techniques. Security investigation have shown the strategy is secure concerning the descriptions shown in the projected security model.

This paper will actualize a prototype of a suggested, sanctioned copy check plan and perform experiments using the prototype. The study will demonstrate that the proposed prototype causes inconsequential overhead-differentiated archetypal processes.

Table of Contents

Chapter Page

Chapter Page

Chapter Page

List of Figures

Figure Page

Chapter I: Introduction

Introduction

Cloud computing provides a variety of "virtualized" assets to the clients as an administrator over the Internet. The present day cloud offers exceptional accessibility and huge benefits at a moderately low price. As distributed computing becomes more common, an ever-rising volume of information is stored on the cloud and imparted by clients with limited benefits, including the privacy of information (Bugiel, Nurnberger, Sadeghi, & Schneider, 2011). To make information administration more versatile in cloud computing, this study implements various de-duplication methods.

Information duplication particularly puts pressure on information systems and can require the removal of duplicate information (Anderson & Zhang, 2010). This project looks to diminish data utilization and, likewise, can be connected to network data exchanges to diminish the amount of bytes that would be charged. De-duplication can take place in either the document or composition levels. In document level, de-duplication disposes of copies of duplicates of the same book (Bellare, Namprempre, & Neven, 2009). In composition level, de-duplication, it disposes of copies of information that fall out in non- indistinguishable documents.

Problem Statement

Data de-duplication has been of great benefit. However, security and protection of client's data have not been guaranteed (Li et al., 2013). Furthermore, the current encryption though providing a wide scope of security is incompatible with information duplication. This type of encryption needs the diverse customers to scramble their information with their keys (Ng & Lee, 2013). Therefore, the purpose of this study is to make indistinguishable data duplicates of

various customers. Furthermore, it will promptly distinguish code text, hence, making de-duplication impossible.

Significance of the Study

Experts have suggested that concurrent encryption can sanction information confidentiality while at the same time making duplication possible (Pietro & Sorniotti, 2012). It states that the encryption decodes the information duplication using a concurrent key, which is obtained by registering cryptographic hash estimation **(**Ng, Wen, & Zhu, 2012**)**. After this, customers use the key to send the unscrambled text to the smarm. Since this process entails and determines information content, the duplicates will make focalized key and from this time cipher text.

Objective of the Study

The aim of this research is to avoid unauthorized access. Hence, an innocuous confirmation of the proprietorship agreement is required to pass the authentication that the customer possesses the same disc when a transcript is found (Stanek, Sorniotti, Androulaki, & Kencl, 2013). After the evidence, subsequent consumers with the same data will be handed a pointer coming from the server without expecting to change the same Platter.

Summary

A customer can transfer the encrypted information with a pointer from the waiter, which has to be decrypted by comparing information from the administrators with their concurrent keys. Therefore, simultaneous encryption allows the cloud to execute the de-duplication on code text, and the proof of proprietorship keeps the unsanctioned clients to go to paper.

While past de-duplication structures were efficient, they did not take into consideration the concepts of copy check that is significant in many devices (Stanek et al., 2013). When de-duplication is sanctioned, each client is given an implementation framework. Every record moved to the cloud is similarly determined by arrangements of profits to show which kind of consumers can do copy checks and fetch documents. Before giving the copy check command for the record of a customer, it requires the document and their specific profits and inputs. The customer can then trace a duplicate of this manuscript if there is a repeat of this information and a synchronized value stored in a swarm. For example, in a corporation, a large number of benefits are doled out to users (Storer, Greenan, Long, & Miller, 2008). Hence, to spare expenses and provide more productive management, data will be transferred to a capacity server supplier (S-CSP) that is located in a Public Cloud with indicated profits and the de-duplication procedure will be linked to keeping only duplicates of a similar book.

As an outcome of security considerations, some information will be encrypted and allow the copy check through representatives that have designated rewards to comprehend the entry control. Orthodox de-duplication contexts in view of simultaneous encryption, albeit voiding privacy to some level, they don’t back up the copy check with different profits.

Ultimately, no differential rewards have been seen in the framework particularly in the vocalized encryption method. It is by all means rejected on the off chance that demands to recognize both duplication and differential authorization copy check.

Chapter II: Background and Review of Literature

Introduction

This chapter provides the documentation utilized as a portion of study about safe de-duplication. Symmetric encryption used employs a typical mystery key κ to scramble and decode information.

Experts define cloud computing as the concept that employs both software and hardware to provide network service. It is an on-demand model used to access a collective group of configured computer resources, which can be quickly provisioned and freed with insignificant administration effort (Rahumed, Chen, Tang, Lee, & Lui, 2011). This type of calculating and storage solutions provides consumers with numerous competencies to store and procedure their data from third-party data centers, which are located far away from end users spread across the world.

The name "cloud computing" is obtained from the utilization of a cloud shaped symbol. It assigns remote services with a user's data, software, and computation power. It is made up of both hardware and software that can be obtained from the internet and even administered by a third party. Cloud Computing is of great importance as it gives access to high-end software applications and servers (Rahumed et al., 2011; Zhang, Zhou, Chen, Wang, & Ruan, 2011). It allows for the easy sharing of information between forms or departments on the same network.

Background Related to the Problem

The Asymmetric encryption plan is comprised of three basic capacities (Bellare, Keelveedhi, & Ristenpart, 2013a).

- $KeyGen_{SE}$ (1)! *K*; this is an algorithm that is used to come up with *κ* through the use of parameter 1.
- Enc_{SE} (*κ, M*)! *C*; this algorithm is responsible for hiding the secret of *κ* and *M* respectively. It also outputs the coded text *C*.
- Dec_{SE} (*κ, C*)! *M*; unlike the other algorithms, Dec_{SE} (*κ, C*)! *M* is used for decryption. It is used to take the secret *κ* and coded text *C*.

This encryption provides data classification in the proposed framework. A consumers or cloud provider acquires a concurrent key from every single distinctive info duplicate and deciphers the data identical to the joined key. Furthermore, the customer also deduces a label for the data duplicates, so that the label is used to identify the copies. From this, it is expected that the tag rightness property stays; for example, if two info replicas are similar, then their tags are alike, therefore, creating a need for a concept to differentiate them. For a client or consumer to know the difference, they are required to send the ticket to the server side to control if the vague replica has been instantly set aside (Bellare, Keelveedhi, & Ristenpart, 2013a). Both merged key, and the tag are put together unconventionally. The tag cannot be utilized to establish the link between United key and trade off information classification. The encoded info and the tag are passed out of the server. The tag is characterized by four simple capacities:

- KeyGenCE (M)! K. This is a key generation algorithm. It assists in illustrating data copy M that is convergent to K
- EncCE (K, M)! C is the symmetric encryption system
- DecCE (K, C)! M is the decryption algorithm

- TagGen (*M*)! T (M) is the tag generation algorithm that represents the original data copy M

Message-locked encryption (MLE) and secure de-duplication. MLE is the key that provides encryption and decryption, hence, gives the message that secures de-duplication using ROM secure tests. The test creates links with deterministic encryption and hash dimensions that connect data information, which is used globally to strategize different types of message sources. Research points out that MLE is a basic of both contemporary and past forms.

Security proofs of identity-based identification and signature schemes. This study provides security examination for a widening number of characters. It does so on the basis of known evidence. Using this information, it makes plans using the unambiguously or verifiably in current documentation (Bellare, Keelveedhi, & Ristenpart, 2013b). It also assists in empowering and improving the proposed security system.

A reverse de-duplication storage system optimized for reading to latest backups. While the proposed framework does not kill all the copies, it creates discontinuity, which in turn corrupts the read execution (Halevi, Harnik, Pinkas, & Shulman-Peleg, 2011). This system is, therefore, one that promotes perusal of the most current reinforcements of virtual machine (VM) pictures. It does so by using de-duplication.

While de-duplication removes copies of event, new data, Reverse de-duplication procedure, only removes that of old ones (Ng & Le, 2013). Therefore, it is possible to transfer the discontinuity of old information while at the same time retaining the format of new data. There have been several studies that have been undertaken to assess Reverse de-duplication procedure. One of the studies used a 12-week range certifiable VM picture that illustrated more

than 60 clients. The research showed that Reverse de-duplication procedure attained high de-duplication proficiency (Ng & Le, 2013).

Secure de-duplication with efficient and reliable convergent key management. One of the primary goals of the study is to address the problem of attaining effective and solid key administration with the absence of insecure de-duplication. Currently, there is a gauge approach in which all the customers have an autonomous expert key for scrambling the United keys and subcontracting them to the cloud. A standard key administration strategy produces a huge number of keys with the raising quantity of customers and forces users to steadfastly guard the expert keys. As a result, it is proposed that it is imperative to use Dekey (Stanek et al., 2013). This system does not give the clients the responsibility of carrying there keys. It instead securely distributes the united key shares over diverse servers (Stanek et al., 2013). Research even shows that Dekey is quite secure and has been used in using of the ramp mystery. The research even reveals that Dekey leads to a constrained overhead in sensible situations (Stanek et al., 2013).

Existing System

Data de-duplication framework is a type of a private cloud. It entails a proxy that permits users to safely carry out a duplicate check with different privileges (Xu, Chang, & Zhou, 2013). In the existing system, data proprietors can only get their data storage by employing the public cloud that is managed by a private cloud.

Disadvantages of Existing System

Traditional encryption though effective in giving security is not in line with the data de-duplication. Identical data copies of diverse client’s triggers code texts and making de-duplication impossible.

Proposed System

This research suggests an augmented system security; **e**specially providing a progressive arrangement to back resilient security by encoding the documents or data with disparity privilege keys.

Advantages of Proposed System

The first advantage is that the customer is allowed to undertake the replica check for files checked with consistent privileges. It also gives the unconventional scheme to aid a robust security by encoding the file with distinctive privilege keys (Zhang et al., 2011). Lowering the storage size of the tags for reliability checks and to boost the security of de-duplication and guard the data privacy.

Summary

This chapter gives the brief overview of the background work that has been done for this project. This also highlights the major elements of the problem statement, for example, encryption, and de-duplication. Message lock encryption, de-duplication and the problems related to this encryption, as well as the existing system and its advantages and disadvantages has been described in this chapter. Also, the advantages of the current proposed project is explained in brief.

Chapter III: Methodology

Introduction

To overcome the potential security issues of cloud computing, this paper proposes a different version of de-duplication framework that is aided by the official copy check. In this framework, an upgraded cloud application is used to solve the problem at hand. It will entail individual keys given to the clients deviously that are they are operated by a third party. This secures the sharing of keys among clients sustaining the profits associated with consumer key sharing. While using the framework, a client usually searches for information by sending a request to the network. To implement copy check for a document, the customer requests the record perfunctory from the isolated cloud server. The server will then establish a link between the customer's profile and the available data prior to providing a comparison between the document token to the node. The client can perform the approved copy check for this record with people in a public cloud before transferring this document. Taking into account the after effects of the copy check, the client either transfers this document.

Prior to the development of the de-duplication framework, characterizing a paired connection equals R = f ((p, p′) g hereafter. Given two benefits p and p′, say that p matches p′ if and just if R (p, p′) = 1 (Pietro & Sorniotti, 2012). This sort of a non-specific double connection definition could be instantiated taking into account the foundation of uses, for example, the basic progressive connection.

All the more unequivocally, in a ranked connection, p matches p′ if p is a larger amount of opportunity. For instance, in a hierarchical administrative framework, three progressive value levels. The first level is the Director, which is the topmost part. The second is project level,

which is at the lower level with the project lead. The level that occupies the base level is the Engineer. Clearly, in this modest illustration, the advantage of Director ties the benefits of Project Lead and Engineer (Bugiel et al., 2011). The suggested de-duplication framework as proposed is described hereafter [6].

Design of the Study

Asymmetric key Kpi for every pi 2 P will be chosen, and the arrangement of keys fkpi gpi∈P is transferred to the private cloud. An ID convention = (Evidence, Authenticate) is additionally characterized, where evidence and authenticate are the confirmation and check calculation correspondingly. Besides, every client U is accepted to have a mystery key SKU to carry out the recognizable proof with servers. It is accepted that client U has the benefits set PU. It additionally introduces a PoW Convention POW for the document possession evidence. The private cloud server will keep up a table which stores every client's open data pkU and its comparing profit set PU (Pietro & Sorniotti, 2012).

Security verifications for an expanding range of characters based on identifiable evidence and make plans characterized one or the other explicitly or verifiable in present writing. Hidden from these is a structure that from one perspective explains how these plans are determined and then again permits specific security checkups, in this manner understanding, improving, and binding together past work (Wilcox-O'Hearn & Warner, 2008). Likewise, it dissects a non-exclusive legend's development that unambiguously yields character based on identifiable proof and makes plans without irregular predictions.

Data Collection

Assume that an information administrator wants to move and communicate a record “F” to clients whose benefit belongs to the set PF + fpjg. The information administrator requires an interface coupled with isolated cloud prior to performing a copy check using the S-CSP. More so, the information manager or administrator needs to perform this task securely.

Input Design

The information configuration links the IS and the client. It involves creating the determination and approaches for data readiness, and these means are essential to establishing the interchange of data into an operational framework. This can be attained by using the PC to parse information from a printed record. It can also take place by direct entry of the information into the system. The plan of information interests on regulating the degree of information needed, governing the errors, sustaining a planned distance from adjournment, restraining from additional means and guarding the procedural basis (Ng, Wen, & Zhu, 2012). Data is planned in this manner to the point that it promotes security and usability. While undertaking input design, several things are taken into consideration:

- The type of data that is to be provided as information?
- The manner in which the data should be masterminded or corded?
- Discourse to manage the working faculty in providing information.
- Techniques for preparing input approvals and undertakings to take over when a mistake happens.

Goals:

1. Input Design is the way toward altering a consumer positioned portrayal of the contribution to a PC based system. It is significant in that it assists in avoiding mistakes in data input procedure. It also determines the right attitude to the management of acquiring the right data sourced from the automated system.
2. It is accomplished through coming up with concepts that simplifies the understanding of screens for the information segment undertake the expansive amount of data. The aim of planning information is to ensure that data passage is less difficult and is devoid of mistakes. The data segment screen is scheduled in a manner that every one of the data controls can be done.

When the information is being fed into the system, it will inspect for legitimacy.

Output Design

One of the output design is provision of accurate data. All systems operate on the fact that that the outcome of preparing data are passed to users and other systems via yields. In a yield outline, it is decided in the manner the information is queried to come up with a quick request and additionally the published copy yield. An industrious and intuitive yield plan develops the system's association to assist a user's basic leadership:

1. Choose approaches for showing information.
2. Make a record, report, or different formations that cover data delivered by the system.

The yield kind of an IS must achieve either of objectives below:

- Transfer data on past exercises, existing status or forecasts of the future.

- Signal dire incidents, open doors, issues, or notices.
- Initiate an action.
- Initiate an action.

Algorithm*:*

- FileTag (File). The procedure calculates the SHA-1 hash of the Folder as the document label.
- TokenReq (Tag, UserID). The procedure requires the Private Server for File Token cohort with the User ID and the File Tag.
- DupCheckReq (Token). The procedure petitions the Storage Server for the Duplicate Check of the File by transferring the file token obtained from the private server.
- ShareTokenReq (Tag, {Priv.}). The procedure needs the Private Server to create the Share File Token with Target Sharing Privilege Set and the File Tag.
- FileEncrypt (File). The procedure encodes the File with Convergent Encryption by employing 256-bit of the AES algorithm in cipher block chaining (CBC) mode, where the convergent key is from SHA-256 Hashing of the file.
- FileUploadReq (FileID, File, Token). The process uploads the File Data to the Storage Server if the file is Unique and the updates of the File Token is stored. Enactment of the Private Server comprises of matching request handlers for the token generation and upholds a key storage with HashMap.
- TokenGen (Tag, UserID). The procedure loads the related opportunity keys of the user and produces token with HMAC-SHA-1.

In this area, the lead performance assessment in light of the estimation on the actualized model arrangement of AnonyControl-F is done to the best of our insight.

Decrypts a document and re-scrambling it under various benefit trees is necessary. This toolbox depends on the CP-ABE toolbox, which is accessible on the mesh, and the entire framework is executed on a Linux framework with Intel i7 second Gen @ 2.7GHz and 2GB RAM. The calculation overhead brought about in the center calculations Setup, KeyGenerate, Encrypt and Decrypt under different conditions (Bellare et al., 2013a). It moreover actualized three comparative works under the same condition (the same security level and same environment) for this examination

This especially set a standard benefit of the document access and quantified an ideal chance to enter one benefit tree and count on its confirmation parameter. When all is said and done, the computation overhead of Li is more in comparison to others particularly in relation to their plan that includes many more exponentiations and bilinear mappings due to the obligation required (Bellare et al., 2013a).

The encryption and unscrambling under various document sizes did not indicate enormous contrasts when record sizes are substantial (≥20MB), in light of the fact that the run times are kept in line by the symmetric encryption (AES-256) (Bellare et al., 2013b). And last, run times are plotted on the grounds that the benefit creation is the extra procedure in design.

System Study

Feasibility study. The proposed task is to analyze or break down in these stages the feasibility of the system, and the proposal is unconventional with an enormously wide-ranging arrangement of options and cost estimates. Amongst system inspection, the plausibility

examination of the suggested system is to be done. This is to guarantee that the suggested framework is not a load to the organization but instead an advantage. This is then an assessment for the practicality of a proposed project.

This study deals about the weaknesses and strengths of the proposed project and discusses the threats, and opportunities in the environment. To determine the required resources to carry through with the success of the project, these studies are carried out to understand the cost required and value to be attained from the proposed project.

The basic design should provide the background information of the project, a brief explanation of product services, accounting statements, operation details, management and marketing investigation and strategies, financial status, legal standards and income tax responsibilities. These studies usually go before practical development and project enactment. These studies also evaluate the potential of the proposed project; the objective is the important factor for credibility for potential investment in the project. The feasibility study examines primary areas:

- Market Problems
- Practical and Organizational necessities
- Financial evaluation

These studies are carried out to get an overview of the proposed project and depending upon the results of these studies one can decide if this project is undertaken, and even if the product will provide the intended benefit to the users given the preferred alternatives. These are the areas that are considered while proposing a feasibility study.

The following are few things that are to be studied

- The preset structural system
- Shareholders, users, rules, functions, and goals.
- Problems with the current system
- Inconsistency and inadequacy in functionality
- Possible alternative solutions

An analysis is required to understand the client requirements and project outlines to carry a feasibility study, once the client requirements and project outline is presented a feasibility study outline is designed according to the requirements.

Types of feasibility. The feasibility study comprises analysis of a complete system; these studies have to be carried out in such a way that, they reflect the operational, economical, and technical feasibility of the system. There are four major feasibility studies that are to be carried in any system: Operational, Technical, Schedule, and Economic.

Operational feasibility. Operational feasibility defines the urgency and the importance of the proposed project. This includes the aspects of social issues, manpower, and certain operation-oriented issues. The current procedures and practices are taken into consideration while carrying out this study, as this will provide a clear understanding of organizational changes that are going to occur while implementing this study. The changes that are going to occur when the proposed project is executed will be defined very clearly by this study. The usual framework used for operation framework is called PIECES. PIECES is defined as: Performance, Information, Economy, Control, Efficiency, and Service.

Technical feasibility. The following research is carried out to inspect the expert viability, that is, the expert requirements of the system. This will trigger ranks of approval on the nearby

particular assets. It will also increase the amount of approval being put on the consumers. The formed system must have the requisite need as just insignificant or unacceptable variations are required for carrying out this system.

This aspect provides an understanding, if the proposed project is within the limit of current technology or not, is it available with given resource constraint. This includes the requirements of the organization such as:

- Devices that can input a huge volume of information in given time.
- Devices that can output an immense quantity of data in given time.
- Proposal of processing units can depend upon the type of processing unit proposed in the organization.

Schedule feasibility. Certain projects proposed are with deadlines, so it is important to consider if the deadlines are mandatory or desirable. It is important to understand if the proposed project needs to meet the deadlines. Depending upon the project proposed and the technical feasibility studies the deadlines can be changed for improvement of the project.

Economic feasibility. The research is carried out to determine the economical or monetary influence the system will have on the organization. Furthermore, it computes asset the organization has and the ability of it allowing innovative work the system needs. The costs must be legalized and accounted for. Along these lines, the created system also must meet the requirements of the financial plan, and this is to be attained on the basis that a huge part of the developments used is openly accessible. Just the altered items must be purchased.

This economic feasibility provides the benefits and costs of the proposed project. Benefits are classified as following:

- Monetary: when the value of money in dollars ($) can be calculated.
- Tangible: when the benefits are quantified, but the $ value is not calculated.
- Intangible: when neither of the above two scenarios applies, it is difficult to quantify.

Costs are classified as:

- Project related cost.
- Development and purchasing cost.
- Installation and conversion cost.
- Operational costs.
- Maintenance costs.

Social feasibility. This portion of the research is endowed with the responsibility of evaluating the degree of client's acknowledgment of the system. This will involve incorporating the way towards preparing the users to employ the system produced. Social feasibility must operate in a manner that the user does not in any way feel that they are being undermined by the proposed system. On the contrary, they should make the client feel that they are getting an improvement. The degree in which the client acknowledges the proposed system is dependent on the strategies used to teach and acquaint them. Therefore, using these strategies, their degree of certainty will be increased with the objectives of the system.

System Requirements

Hardware Requirements:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 MB.

- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 MB.

Software Requirements:

- Operating system: Windows XP/7.
- Coding Language: JAVA/J2EE
- IDE : Netbeans 7.4
- Database : MYSQL

Summary

This chapter explains the design of the proposed project, the object, and input and output design. This chapter also explains the software and hardware requirements and the types of feasibility studies that were covered in Chapter III.

Chapter IV: Implementation

Introduction

Implementation is the execution of a plan, idea, model, or design. It is a realization of technical specifications or algorithms for a program. Implementation exists for a specification or a standard. This chapter explains about the implementations of this proposed project.

Main Modules

- Cloud Service Provider
- Data Users Module
- Private Cloud Module
- Secure Deduplication System

Modules Description

Cloud service provider. A Cloud Service Provider module is a component that is established to provide data storage services, particularly in the public cloud. It also acts as a data outsourcing component. It also provides storage for the clients. In order to lower the storing cost, the module restricts the storage of repetitive information through duplication and maintains the original data (Rahumed et al., 2011). However, the module has a limited storage capacity and the calculation power.

Data users module. Customers usually use S-CSP to store data that they can retrieve later. In a storage framework backing de-duplication, the customer just sends exclusive data yet does not move any copied data to substitute the transfer, transmission capacity, which might be claimed by similar client or other clients. In the new de-duplication framework, each customer has delivered a plan of profits in the arrangement of the framework. Each document that is

provided is made secure by the use of shared encryption key. Therefore, it is of great benefit for the client to have an understanding of it and its benefits.

Private cloud module. Unlike the normal de-duplication design in cloud configuring, Private Cloud module is also one of the systems that are used to ensure client's cloud service security. In particular, as cataloging assets of data of the consumer side are narrowed, and the public cloud is not totally confided in, a private cloud can provide the data of the customer/proprietor with an implementation domain and foundation functioning as a crossing point amongst the client and the public cloud (Ng et al., 2012). The interface provided by the private cloud allows the user of cloud services ability to give out records and questions that are processed and stored privately.

Secure de-duplication system. While securing the de-duplication system, there are two main concepts to take into consideration; these concepts are outside enemy and inside foe. As shown in various researches, outside enemy is an internal foe with little or no benefit of access. When a user of cloud services has, a benefit denoted as p, it is important that the enemy does not generate and produce a significant copy token with whatsoever additional benefit p′ on any document F, where p does not coordinate p′. Moreover, it similarly demands that when the enemy fails to establish a solicitation of the token with its particular benefit from the private cloud server, it can't fashion and yield a substantial copy token with p on any F that has been examined.

Results

This is the log in page of the Administrator (Admin) account (Figure 1). In this page, the Admin will have to enter the given User Name and Password.

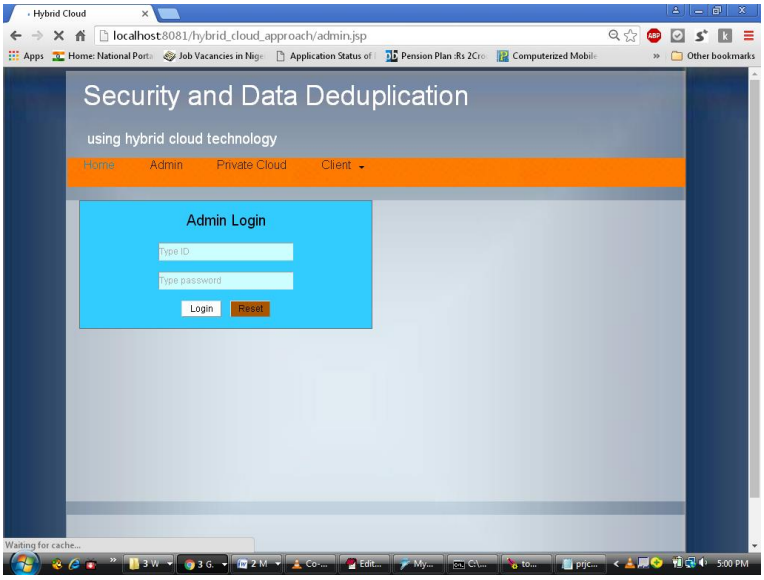

Figure 1. Admin Login Page

The screenshot shown in Figure 2, shows the dashboard page after a successful login of the admin account

Figure 2. Admin Logged in

The screenshot in Figure 3 gives the brief about Admin activity page, after a successful login.

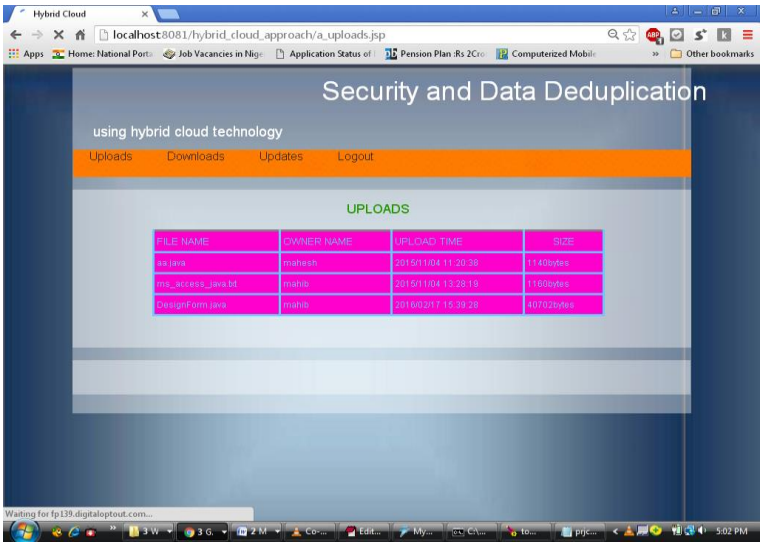

Figure 3. Upload Activities

The screenshot in Figure 4 shows the Admin operations where the Admin will be able to monitor the updates of different files uploaded by different users.

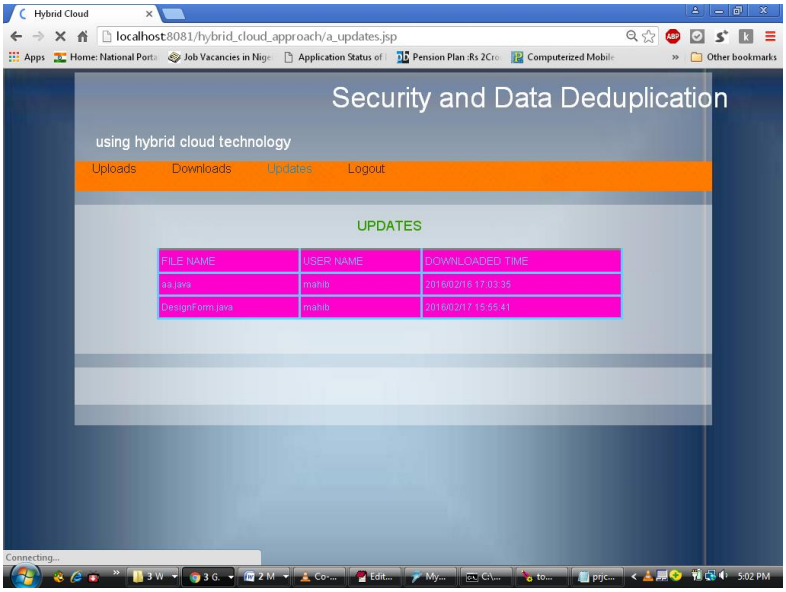

Figure 4. Update Activities

The screenshot in Figure 5 shows the download activity of the Admin where the Admin will be able to download the user's uploaded files.

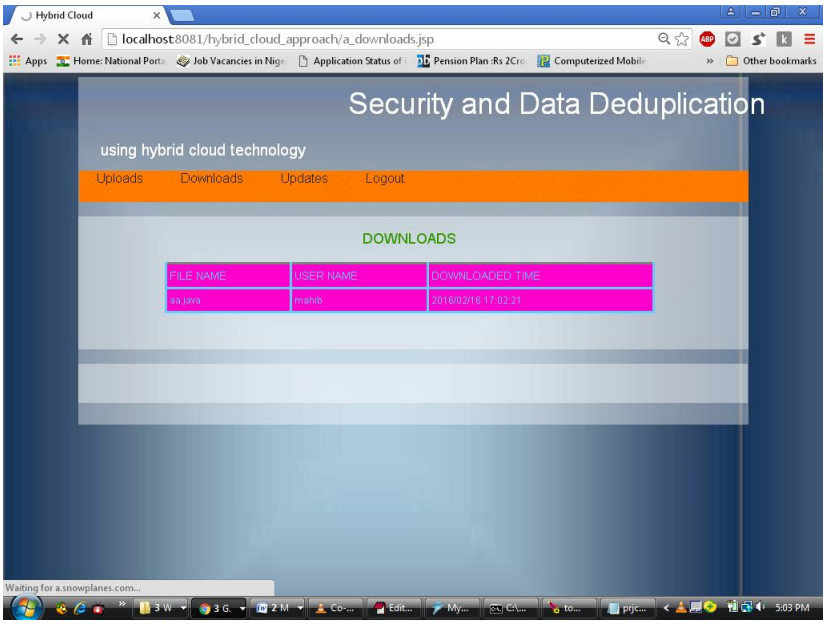

Figure 5. Download Activity

The screenshot in Figure 6 shows the Private cloud login page.

Figure 6. Private Cloud Login Page

The screenshot in Figure 7 shows the page after a successful login to the private cloud.

Figure 7. Private Cloud Logged On

The screenshot in Figure 8 gives a brief update of the system after logging on to the private cloud. The client details such as Name, User Name, E-Mail, Status and the Action will be visible.

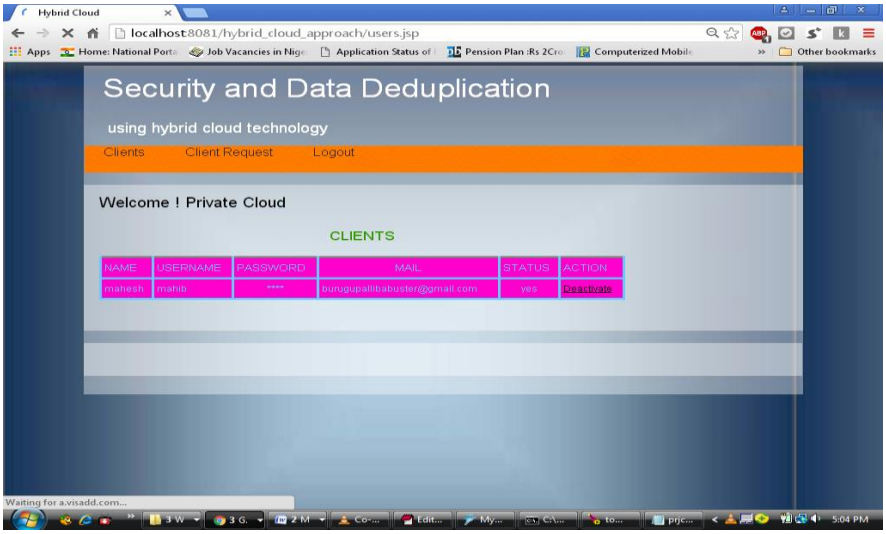

Figure 8. Clients Available in the Cloud

The screenshot in Figure 9 gives a brief about the client request. Once a client places a request, it appears in the private cloud and the request can be aprroved or denied by the Admin.

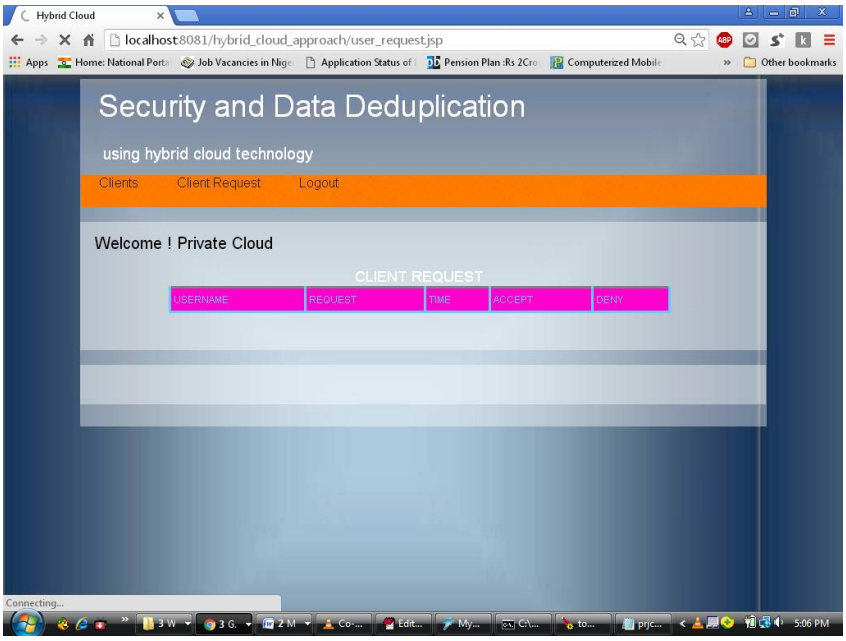

Figure 9. Client Request

The screenshot in Figure 10 shows the client registration page.

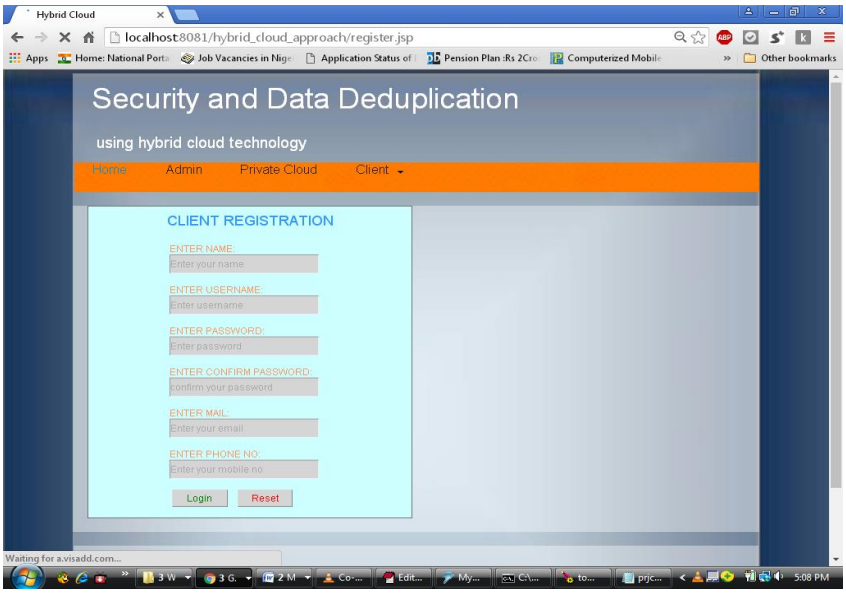

Figure 10. Client Registration Page

The screenshot in Figure 11 shows the login page of the client.

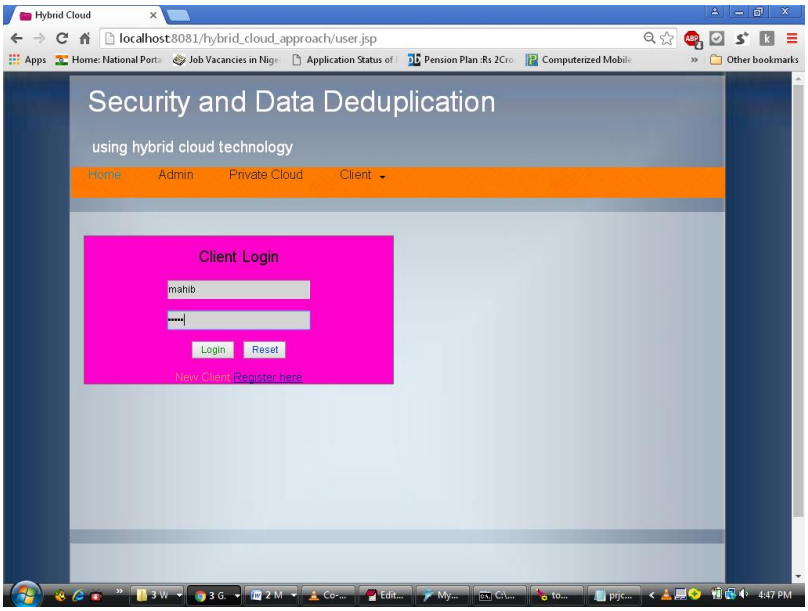

Figure 11. Client Login

After the client is registered and the request is approved, the client would get a token, the token will act as the secondary form of authentication as shown in Figure 12.

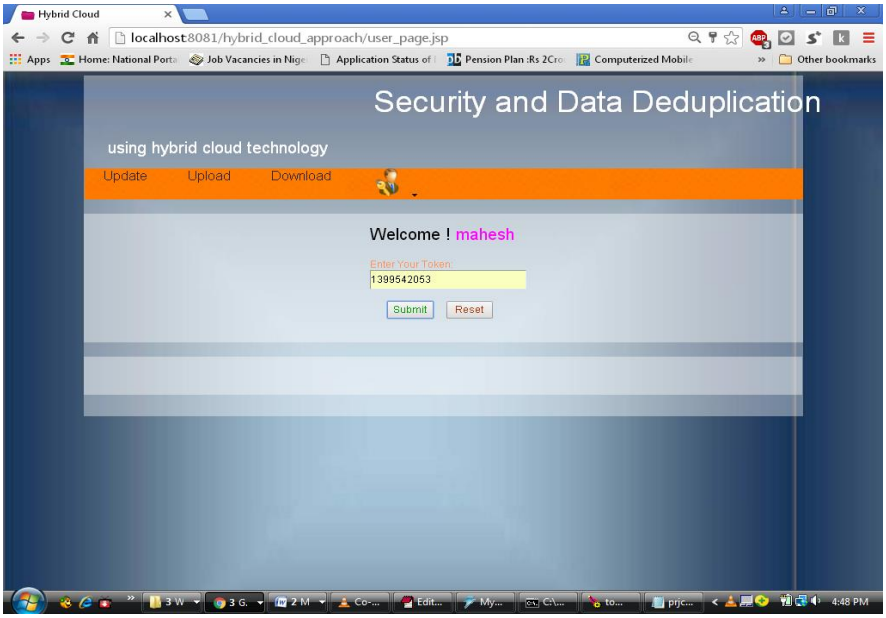

Figure 12. Token Verification

After the successful login, clients will be able to upload their data. The screenshot in Figure 13 shows the data upload page.

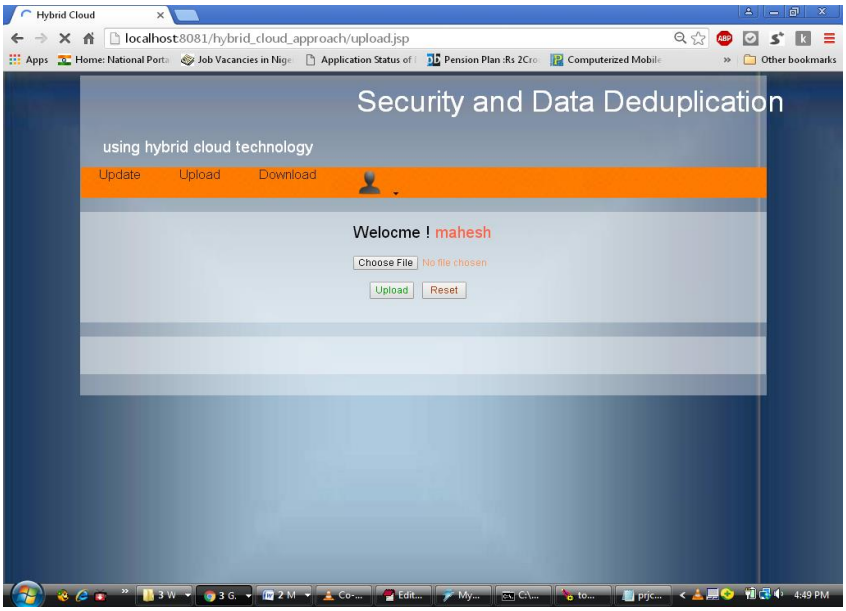

Figure 13. Client Logged In

The screenshot in Figure 14 gives the brief about the client activity and where the client can view his activities and manage them accordingly.

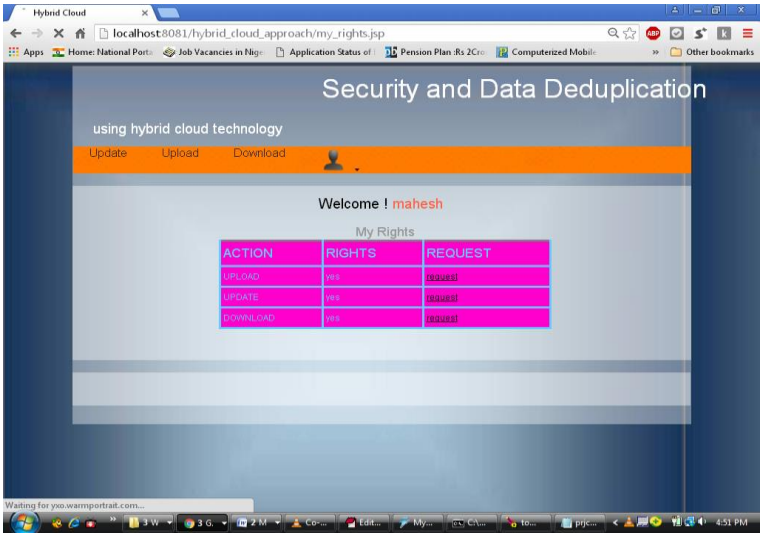

Figure 14. Client Rights

Summary

This chapter explains about the implementation of the system. This implementation is to explain about the plan and the model, it also explains the modules and the componets of the proposed project. The results of the project are described in brief and how they are going to look like with the screenshots of the web pages that are being developed. This is the overview of Chapter IV.

Chapter V: Software Environment

Introduction

This chapter explains the kind of software environment required for the proposed project to be executed in. The software and the programs that are to be used in this project have also been briefly explained.

Java Technology

The first technology to be used in implementing the proposed project is Java technology. Java is a programming language, which keeps running in the background on a platform called the Java Virtual Machine (JVM). This language is an abnormal state language that can be described by accompanying it with trendy expressions with most programming languages, either assembling or translating systems so that it can be run on a PC (Quinlan & Dorward, 2002). There are a number of languages used when running a Java program; these languages are:

- Simple
- Architecture Neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust

- Dynamic
- Secure

While the Java Programming language is common in many projects, it cannot be found in one that is both ordered and decoded. In the Java programming, there are two concepts compiler and interpretation. In the compiler, the interpretation of the message is done on a Java 32-bit platform. The platform then unconventionally cyphers and decodes the message by the use of the translator. The translator then analyses and runs every byte of code rules on the computer. The buildup occurs only once, nonetheless, the translation takes place each time the project is implemented. Figure 15 illustrates how the function operates.

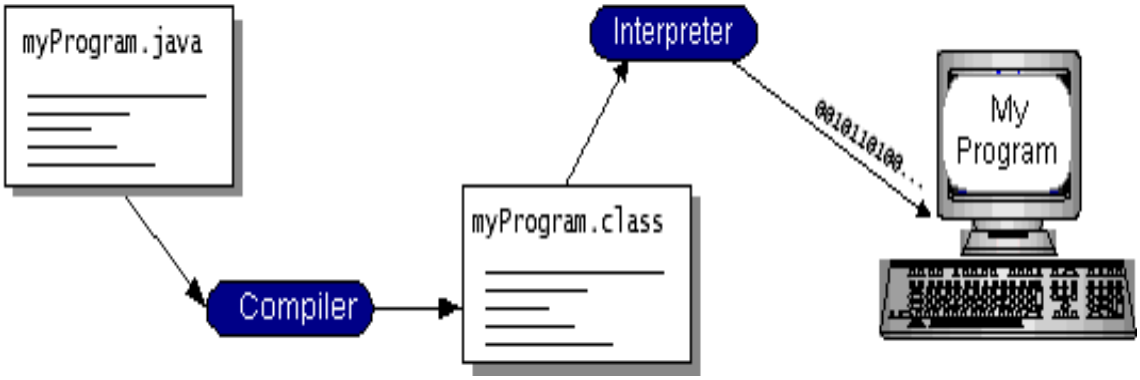

Figure 15. Java Compiler

Each Java Mediator either is an improvement device or internet program, which runs applets and is an implementation of the Java VM. Java bytecode is a “Compose once, Run anyplace” convenient code (Quinlan & Dorward, 2002). Thereafter, gather the proposed framework into bytecode on any platform that contains a Java Compiler, the byte code can then operate on any type of Java VM. This illustrates that a computer that has been installed with Java VM can run on several operating systems ranging from MacOC to Windows 2000.

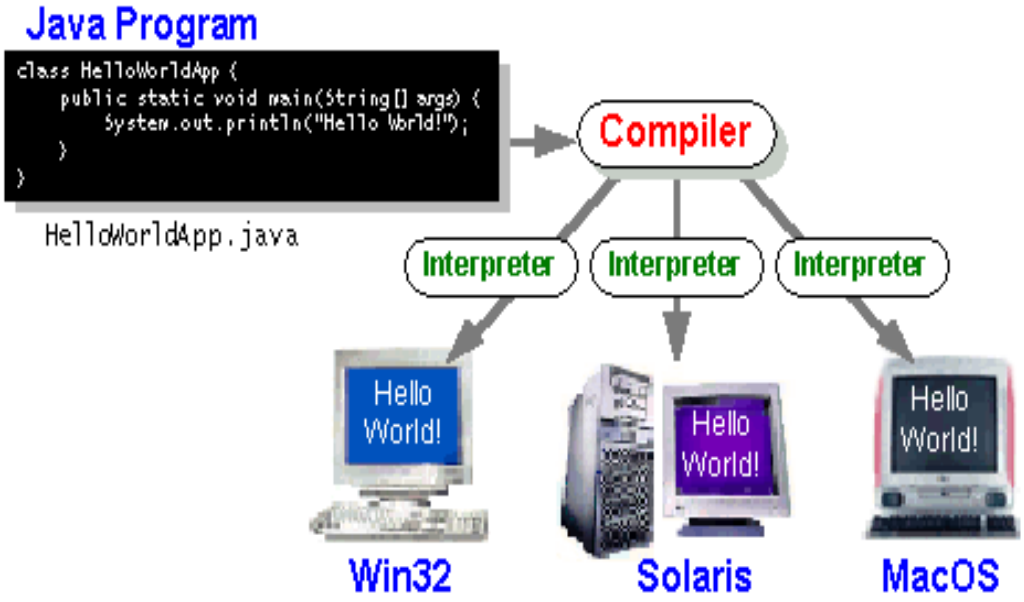

Figure 16. Java 2 SDK

The Java Platform

To better understand the Java platform, it is imperative to know what a platform is. Therefore, a platform is a tool or programming environment in which computer systems operate. They can range from Linus, Windows 2000 to MacOS. The Java platform is a tool that differs from other equipment-based platforms, for instance, it operates on other platforms. As a platform, it is characterized by two main parts namely:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

The Java API is a broad integration of prompt programs that have diverse competencies like Graphical User Interface (GUI) gadgets. It is put together into archive linked programs and interfaces that are called as packages.

Figure 17 describes a project that is operating on a Java platform. According to the figure, the two parts of Java platform do not operate together but separate the system from the platform.

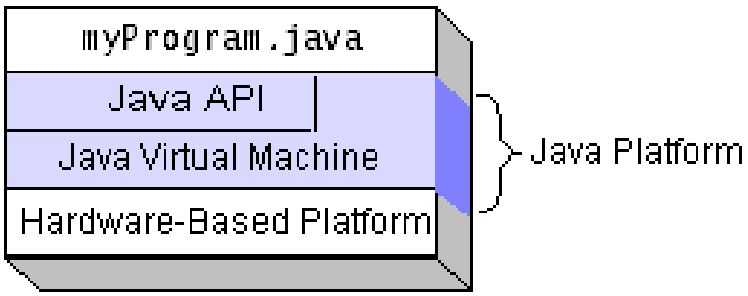

Figure 17. Java Functions

While operating on a Java platform, there is a concept known as native code. This code usually operates after assembling. It keeps operating on the equipment platform. It is more effective than the Java platform as the platform runs the code much slowly (Bugiel et al., 2011). The savvy compiler is a much more tuned mediator, and at the last moment, bytecode compilers convey execution near the local code without undermining conviviality.

Java Virtual Machine (JVM):

JVM is one of the components that allows computers to run or operate on a Java platform. It has three major ideas, they are:

- Specification. This is a document that formally describes the required JVM implementation. It has a single specification and ensures that implementation is interoperable.
- Implementation. JVM enactment is a program that satisfies the necessities of JVM conditions.

- Instance. JVM implementation is a procedure that implements a computer program compiled into Java bytecode.

JRE is a Java runtime environment, which is a software package, which contains a Java program, and it contains JVM implementation together with implementation of the Java class library.

JDK is a Java development kit, which contains tools for the Java program; this is provided by the Oracle Corporation free of charge.

JVM is an abstraction defined by a computer by a set of specifications; it is a language that can be expressed in terms of valid class files hosted by the JVM. Class files contain JVM instructions and symbol tables, and other ancillary information. The format of class file is operator independent, system independent, binary format represented by compiled classes and interfaces.

What Can Java Technology Do?

The programming language used in Java technology does not operate under specific operating system. Therefore, it is able to run on different operating systems and hardware. The most widely recognized sorts of Java programs are applets and applications (Bugiel et al., 2011). An applet is a project that sticks to a particular system that allows to keep operating in a Java authorized program.

The Java programming language is not only written for engaging applets for the interne, but also it is a proficient programming platform for using liberal API’s and can include various types of projects. Diverse platforms aim at varied classes of device and application domains. These application domains are:

- **Java Card:** This is a device that enables applets to operate firmly on smart cards and small memory technologies.
- **Java ME:** This is a Micro Edition technology, which runs with a set of outlines for tools with restricted storage, display, and power capacities. It is normally used in devices like printers and PDAs.
- **Java SE:** This is the contemporary version used for desktops, PC's, servers, and similar devices.
- **Java EE:** This is an Enterprise version, which is beneficial for Multi-tier client/server enterprise applications.

The Java platform is endowed with a number of programs that make it function effectively and carry out several functions. It has a Java Development Kit, which consists of the Java Compiler, which is responsible for changing source code into the Java bytecode. The Just in Time Compiler has been made effective by the favorable Java environment enabling the efficient in changing the intermediary bytecode to innate machine code. Besides having several components, the Java platform is characterized by a huge group of libraries. The most important parts are Java language compiler and the runtime environment. Other significant concepts are "Virtual Machine" which performs the Java bytecode program. The bytecode does not depend on the hardware or operating system. Unlike other type of programs, the Java ones can operate on different platforms. These Java programs are platform independent, and the JVM executes its operating platform. A huge portion of reusable code are given to the programmers to simplify their job, the Java platforms offer an all-inclusive set of libraries encompassing reusable

functions found in contemporary OS. In conclusion, the Java platform serves three functions within the libraries:

- Java libraries provide programmers a reusable set of libraries to perform common tasks.
- They offer a theoretical interface to the responsibilities that will be dependent on hardware and the operating system.
- In the event the software fails to cover all the features, the Java application executes and acts as a substitute or provides a reliable method to check for the presence of particular features.

Furthermore, the Java platform provides the following accompanying components:

- **The essentials:** Objects, strings, numbers, info and yield, information structures, system properties, date and time, et cetera.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization**: Help for constituting programs that can be restricted to users worldwide. Projects can naturally change to specific areas and are displayed in the correct language
- **Security**: It is provided the low level and abnormal states. It also entails electronic marks. Furthermore, it encompasses public and private key administration. Others are access control and endorsements.
- **Software components**: Also called JavaBeansTM. They are responsible for linking present component models.

- **Object serialization**: Permits inconsequential tirelessness and communication via Remote Method Invocation (RMI).

Java Database Connectivity (JDBC™). Gives unchanging entree to a large group of social databases. Furthermore, the Java platform contains APIs for 2D and 3D illustrations, openness, servers, cooperation, communication, discourse, liveliness, and that's just the beginning. Figure 18 portrays what is incorporated into the Java 2 SDK. Still, it is prone to improving projects and needs less effort than other programming languages.

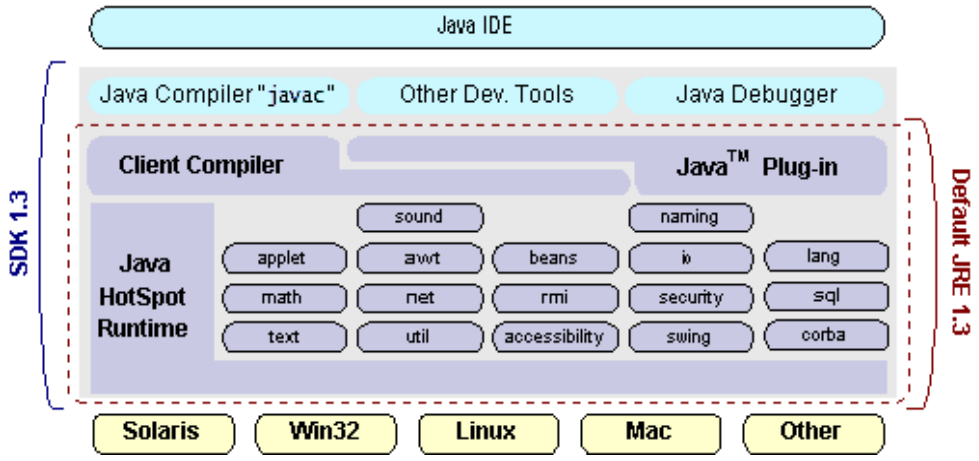

Figure 18. Java Platform

- Even though the language used in Java programming is object-oriented, it is not difficult to learn particularly when the programmers have knowledge of C++.
- Different projects illustrate that those that have been written using Java are much smaller than the ones written using C++.
- Compose better code: The Java programming language enables awesome coding practices, and its garbage collecting helps dodge memory spills.
- Develop programs more quickly: Development time might be cut in half versus composing the same project in C++.

- Write once, run anywhere: Since all Pure Java projects are amassed into machine-free bytecodes, they operate consistently on every Java platform.
- Appropriate and effective programming: To restructure applets effortlessly from the main server. Applets take advantage the element allowing new classes to be stacked "on the fly," devoid of recompiling the whole program (Bugiel et al., 2011).

ODBC and JDBC

ODBC (Open Database Connectivity). ODBC is a type of standard (API). It is effective for accessing database management systems. Its drivers can be used as translation layer amongst the program, and the DBMS accomplishes DBMS independence. Any application that is amenable to ODBC application is able to gain entry to every DBMS.

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application designers and database systems designers (Bugiel et al., 2011). In the past, software engineers were required to use restrictive languages while using ODBC. This is because the programming interface had not been standardized to fit with Windows projects.

Currently, ODBC has solved the issue of database systems becoming irrelevant from a coding standpoint. However, engineers that use the application still need several things to put across the fact that a project will begin with a database but will later change abruptly (Bugiel et al., 2011).

These ODBC systems were not introduced into systems with Windows 95. Instead, they are introduced while setting up different database applications. At the point where the ODBC symbol is setup, it uses a document known as ODBCINST.DLL. It is imaginable to control ODBC information sources through a standalone program known as ODBCADM.EXE.

However, there are 16-bit, and 32-bit variants of this system and each one maintains a diverse run down of the ODBC data system.

From a programming point of view, the beauty of ODBC is that it can be designed to use a similar arrangement of capacity calls to crossing point with any data source. The source code does not any way alter even if it converses with SQL server. As a result, it can only be accessed by known databases. From this concept, programmers are able to change even excel spreadsheets to information banks. ODBC is also used in a working system. Its function is to determine how the drivers are expected to converse. While undertaking this task, the stacking of its drivers are forthright.

In a client environment, the ODBC API is responsible for a huge part of the system problems for the use by a software engineer. Microsoft has reliably stated that the basic element in implementation is the attribute of the driver programming that is used.

The ease of access to good ODBC drivers has promoted an implausible organization as of late. Possibly not, but somewhat the compiler (or ODBC) provides the opportunity to make up cleaner programs, which implies that they will be complete sooner. In the interim, computers continue to get faster and faster consistently.

JDBC. Java Database Connectivity is an application-programming interface, which describes how the customer will access the database. This is a portion of the Java Standard version platform developed by Oracle Corporation, this provides query and update data in a database.

To create a database standard API for the platform, one of the companies came up with Java database connectivity otherwise known as JDBC. JDBC provides a non-exclusive SQL

database access system that offers a steady crossing point to types of RDBMS. It is accomplished using “Module” database availability or drivers.

Unlike other programming packages, JDBC has many objectives that have promoted the progress of API. Together with other commentator input, the objectives have established the JDBC class into a good library that can be used to design and construct database applications in Java. Moreover, the objectives are critical as the given knowledge with reference to certain class. Therefore, it should be known that the objectives have different functions that are created by eight outlines categorized into the following groups.

SQL Level API. The founders felt that the main objective is to give a description of SQL interface Java. Irrespective, of the fact that the objectives do not operate at all levels, it functions more on the low level. Otherwise, it is at adequately high level of the application for software technicians to use it promptly. Attaining this aim considers any possible future apparatus sellers to “yield” JDBC code. It also hides numerous of JDBC’s intricacies from the end clients.

SQL conformance. SQL sentence structure is different as it shifts from database seller to dealer. With the aim of strengthening a large variety of sellers, JDBC allows any review; the explanation then goes through it to the vital database driver. This allows the JDBC to deal with non-standard questions, in a manner that is suitable for its customers.

JDBC must be executed on top of common file interfaces

To utilize existing ODBC level divers, JDBC SQL API must “sit” on top of other regular SQL Level API. Being on top allows the interface to make interpretation JDBC calls to ODBC and vice versa.

JDBC must also offer a Java interface that is in line with other forms of Java system. Because of the Java's acknowledgment in the customer group this far, the developers have the perception that they ought not to wander from the current alignment of the central Java framework.

Another common objective is to keep it simple. These goals show up in most of the products in the outline post goals. The configuration of JDBC is designed to be exceptionally straightforward that takes only one technique into consideration for finishing the assignment for each instrument. Authorizing copy practicality just helps to confound the customers of an API.

The other objective is using strong and static typing wherever possible. Solid writing takes into consideration more mistakes checking to carried out at the gathering time; likewise, fewer blunders show up during the runtime.

Furthermore, the Java platform aims at keeping the common cases simple. As a common rule, many developers use different SQL calls. These calls are SELECTs and INSERTs. Others are DELETEs and UPDATEs. Notwithstanding, more complex SQL vocalizations must be plausible.

Java has two things—a programming language and a platform. This is an abnormal state programming language that is a greater part of the accompanying process. This is irregular in every Java project with combinations of both ordered and translated programs. With an interpretation of this program to an intermediate of the road language called Java bytecode, the platform independent code direction takes place and kept operation on the computer. Gathering takes place just once, however, understanding occurs each time the system is implemented. Figure 19 helps to understand how Java functions.

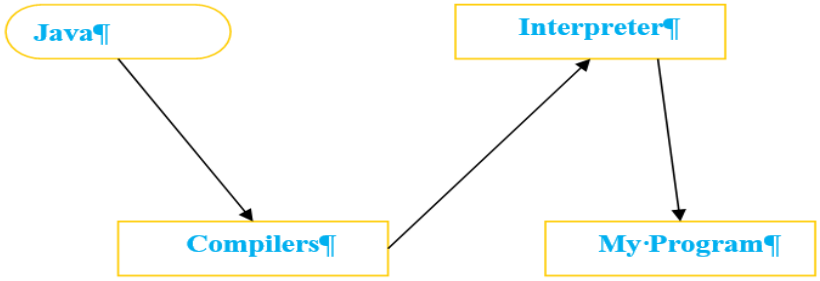

Figure 19. How Java Functions

The Java bytecode as the machine code directions for Java virtual machine (Java VM). Each Java translato either a Java improvement instrument that can function on Java applets, is a usage of Java VM. The Java VM can be actualized in the equipment. It functions in a Java program platform. It changes the Java program to bytecodes, which are then responsible for running any use of Java JM.

Networking

TCP/IP stack. The TCP/IP stack is shorter than the OSI one:

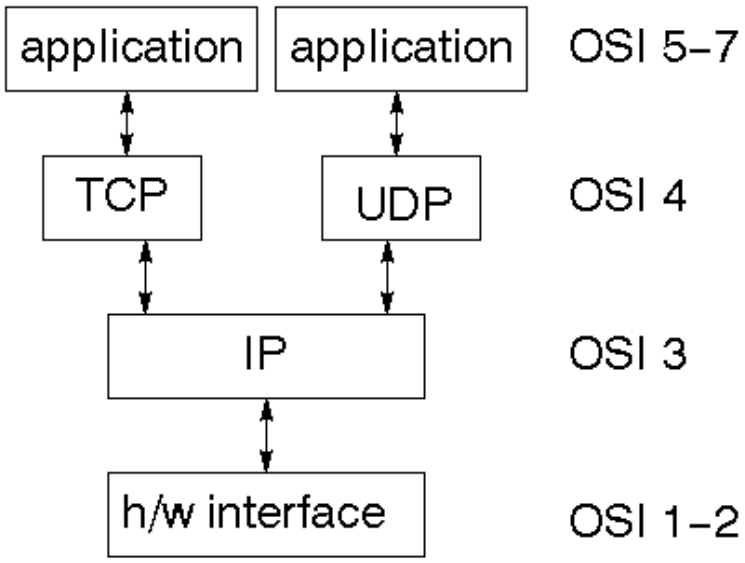

Figure 20. TCP/IP Stack

IP datagram's. The IP layer offers a fickle data movement system that requires no connection. It takes into consideration each datagram self-governing of the others. The higher layers of the TCP/IP stack must supply any relationship between the datagram. The IP layer supplies a checksum that consolidates its specific header. The header joins the source and ultimate locations.

The IP layer is also responsible for directing of packets via the web. It is furthermore accountable for isolating sweeping datagrams into humbler ones for conduction and reconstructing them at the destination.

UDP. Like the IP layers, the UDP does not require a connection to its protocol. This makes it possibly risky. Its task is to add IP header a checksum for substance of the datagram and port numbers.

TCP. The TCP protocol supplies a basis to provide a strong affiliation arrangement tradition on top of the IP protocol. It provides a virtual circuit that 2 endpoints can utilize to pass on and form a connection.

Internet addresses. To use an admin, it should have the capability to ascertain it. The web uses a location plan for machines to make it easy to locate. This place is a 32-bit figure broken into a network and node portion, which provides the IP address. This encrypts a system ID, which fits into diverse classes according to the extent of the system address.

Network address. Class A utilizations 8 bits for the network address with 24 bits left over from other tending Addressing. Class B utilizes 16-bit network tending to. Class C utilizes 24-bit network tending to, and class D utilizes each of the 32 bits.

Subnet address. Inside, the UNIX network is isolated from sub-networks. Building 11 is right now on one subnetwork and uses 10-bit tending to, permitting 1024 distinct hosts.

Host address. Eight bits are at least utilized for host addresses inside subnet. This places a breaking point of 256 machines that can be on the subnet.

Total address.

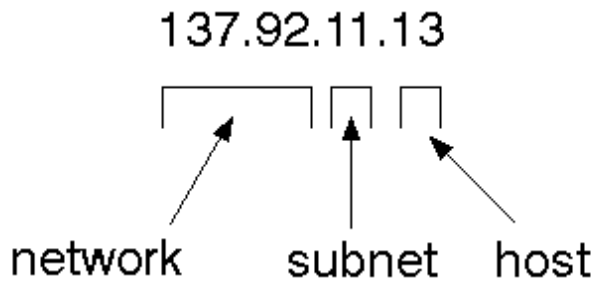

Figure 21. Total Address

The 32-bit location is typically composed of four whole numbers isolated by periods.

Port addresses. A service is found on a host and is distinguished by its port; this is a 16-bit number. To make a connection to a server, send it to the port for that service which is running on the host. This is not area that is always straightforward! Some of these ports may not always be "notable."

Socket. When talking about on sockets, it is imperative to mention on the concept of attachment, which is described as an information structured, stored to carry out network linking. It is created by using the cell attachment. It gives back an entire figure that resembles a record descriptor. When using Windows, it can be used with Read or Write file, for example, #include <sys/types.h>. In this concept, the "family" will be illustrated as AF_INET for IP interchanges. The convention, on the other hand, is denoted as zero. Lastly, sort will depend on if TCP or UDP

protocol is used. Two processes wanting to link over a network make an attachment to each. These are like two ends of a pipe. However, the real pipe is not existent.

JFree chart. JFreeChart is a Java outline library that assists engineers to show proficient quality graphs in their use. This is an "Open source" free programming. It follows the rules and regulations provided by the GNU Lesser General Public License (LGPL), which grants an exclusive license to use the programming language.

Map visualization. These are the Charts which demonstrate the values that identify within geographical regions. A few illustrations include the population density of the United States and the pay per capita for each country in Europe. Perform additional (to JFreeChart) highlight for intelligent time planning charts to demonstrate a diverse control that reveals a slight form of entire the time arrangement information, with a sliding "viewpoint" box that permits the choosing of division in preparation info to illustrate in the fundamental diagram.

Dashboards. There is as of now a considerable measure of interest for dashboard views. Develop a flexible dashboard tool that backs a subclass of JFreeChart outline sorts easily communicated effortlessly by means of Java Web Start and an applet.

Property editors. The property supervisor system in JFreeChart is responsible for a small subsection of the properties that can be a set of graphs. It gives prominent end client control over the presence of the diagram.

J2ME (Java 2 Micro Edition). This is the micro edition of the Java platform made specifically for fixed systems. The gadgets for which this software is designed are industrial software and Mobile phones. They also function well in set-top boxes.

In June 1999 at the Java One Developer Conference, Sun Microsystems characterized J2ME as “an exceptionally upgraded Java run-time environment. It focused on several shop items like pages and PDAs. It also looked at screen telephones and advanced set-top boxes. This brings the expediency of the Java language to smaller devices allowing portable remote devices to share similar applications. With this Microsoft has tuned the Java stage for consumer goods that fuse or depend on small registering devices. This Java ME offers a flexible application setting for applications operating on the embedded technologies like mobile phones. This entails Flexible user interfaces and strong security. It also comprises of support for networked and offline applications which can be transferred with dynamism.

General J2ME architecture.

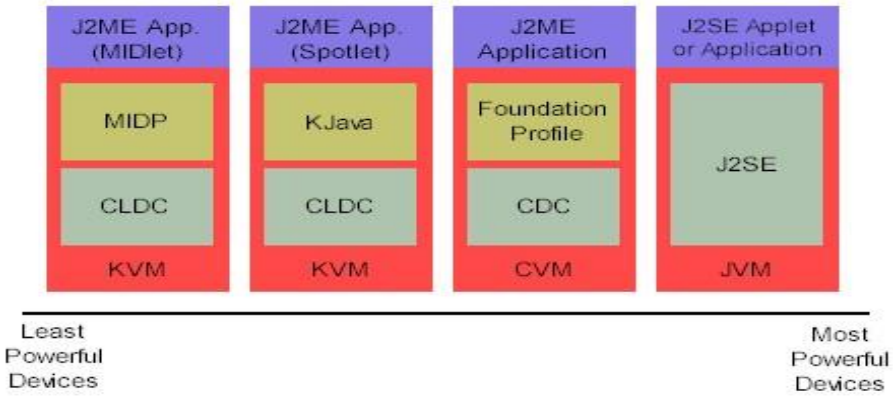

Figure 22. General J2ME Architecture

J2ME utilizes arrangements to alter the Java Runtime Environment (JRE). As a JRE, J2ME have included a stop that makes a decision on the JVM to be used. It also entails a profile that describes the application by including space for specific classes. The design illustrates the essential run-time setting as a planning of center classes and a specifically JVM that keeps operating on specific sorts of devices. The profile personifies the device; mainly, it supplements

a specific class to the J2ME structure to describe particular uses for devices. We will spread outlines top to bottom in the associated realistic example, which describes the link between the various virtual machines, arrangements, and profiles. It attracts a parallel with the J2SE API and its Java virtual machine.

Developing J2ME applications. While creating applications for smaller devices, one should investigate whether the compiler is required when utilizing J2SE to gather a J2ME application and then an investigation about bundling and send the pre-verification plays in this procedure.

Design considerations for small devices. Coming up with applications to be used in small technologies involves considering certain processes particularly in the outline phase. It is usually advisable to provide an outline before undertaking the initiative of coding for a small gadget. Redoing the code once you have abandoned to contemplate on the majority of the "gotchas" prior to developing the application can be an excruciating and time-consuming process. Following are some configuration process to take into consideration:

- Keep it basic. Dismiss unessential mechanisms, possibly making those elements a changed, non-compulsory application.
- Smaller is better. This is among the easiest decisions to be made by a designer. Smaller applications use a lesser amount of memory on the device and need petite formation duration. Take into consideration establishing Java applications as packed Java Archive (container) documents.
- Lessen runtime memory use. To decrease the amount of memory used at the runtime, utilize scalar sorts set up of article sorts. Additionally, do not depend on garbage

collector. This will assist in solving the issue related to memory effectively by establishing object references to invalid when they have completed using them.

- One of the other techniques that are used to decrease run-time memory is to use sluggish instantiation. Diverse techniques for decreasing in general and highpoint memory utilization of little devices are to release resources quickly, recycle protests, and dodge exemptions.

***Configuration overview*.** The setup describes the vital run-time environment as a preparation of center classes and a specific JVM that keep operating on specific types of devices. Two arrangements exist for J2ME:

- **Connected Limited Device Configuration (CLDC).** CLDC is utilized with the KVM in 16-bit or 32-bit devices with restricted levels of memory. This is the setup utilized for developing small J2ME applications. Its size constraints make CLDC additionally fascinating and testing (from an improved perspective) than CDC. CLDC is likewise the design that will be used for building up and drawing the instrument application. A case of a small remote device running small applications is a Palm handheld PC.
- **Connected device configuration (CDS).** CDC is utilized with (CVM) and is used in 32-bit models needing over 2 MB of memory, for instance, a Net TV box.

J2ME profiles. A profile characterizes the kind of device that is bolstered. For instance, the mobile information device profile (MIDP) characterizes classes for PDAs. It adds particular space classes to the J2ME arrangement to characterize the utilization of comparable devices.

Two outlines have been characterized for J2ME. They founded on CLDC. These classes are Java and MIDP. These two are connected with CLDC. They are also linked to smaller devices. The outlines are based on arrangements. As the profiles are specific to the span of the technological tool on which an application run's, specific profiles are connected with specific setups.

- **Profile 1: KJava.** KJava is Sun's restrictive profile. It is made up of the KJava API. Its profile is based on top of the CLDC design. Its virtual machine, KVM, acknowledges similar bytecodes and class record design as the great J2SE virtual machine. KJava has a Sun particular API whose function is to operate on the Palm OS. Its API has an incredible arrangement in a similar manner as the J2SE Abstract Windowing Toolkit (AWT). In any case, its fundamental bundle is com.sun.kjava.
- **Profile 2: MIDP.** MIDP is designed for cell phones, for example, mobile phones, and pagers. While acting as KJava, the application operates on the concept of CLDC and provides a standard run-time setting that permits new applications and administrations to be sent powerfully on end client gadgets. MIDP is a typical, industry-standard outline for cell phones that are not reliant on a particular merchant. It is a far-reaching and fully established for versatile application developments. MIDP has the accompanying packages, the initial three of which are center CLDC packages, in addition to three MIDP-particular packages.
  - ❖ java.lang
  - ❖ java.io
  - ❖ java.util
  - ❖ javax.microedition.io
  - ❖ javax.microedition.lcdui
  - ❖ javax.microedition.midlet
  - ❖ javax.microedition.rms

Tomcat 6.0 Web Server

This is an open-source web server made by the Apache Group (Figure 23). It acts as a servlet container utilized in the standard reference implementation for JavaServer Pages technologies, which were made by Sun under the Java Community Process. They only support web tools. Its server, on the other hand, supports business components like BEAs WebLogic to come up with an internet application that has JSP/servlet install any web server.

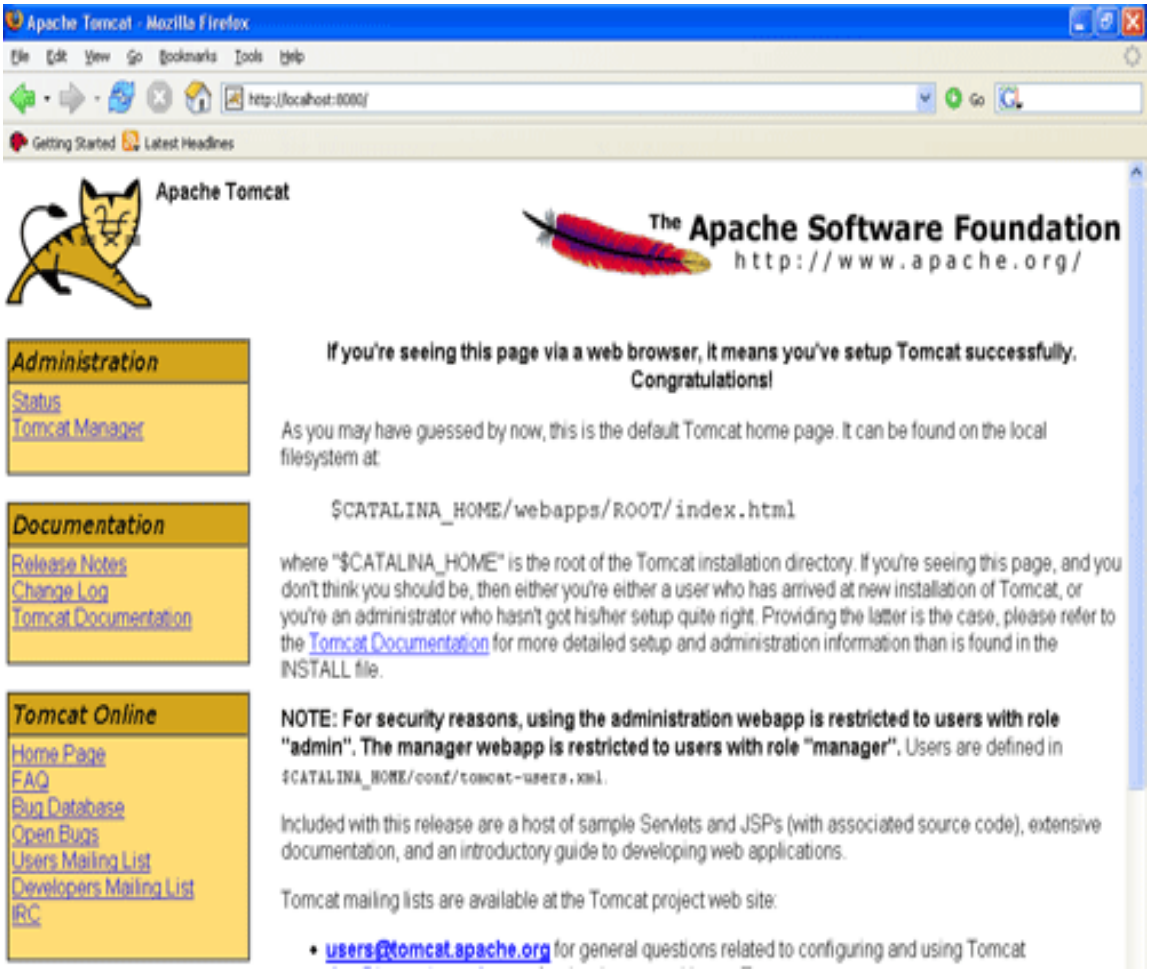

Figure 23. Tomcat Web Pages

Summary

This chapter explained about the software requirements that have been used to execute the current project. The project is based on Java and the Java platform, and that has been explained briefly with the results and examples like the Tomcat web page.

Chapter VI: System Design

Introduction

This chapter explains about the system design. For every project, a system design is very important, as it is a project that is being executed to solve a problem. The design should be reliable and user-friendly to be used by the end users.

System Architecture

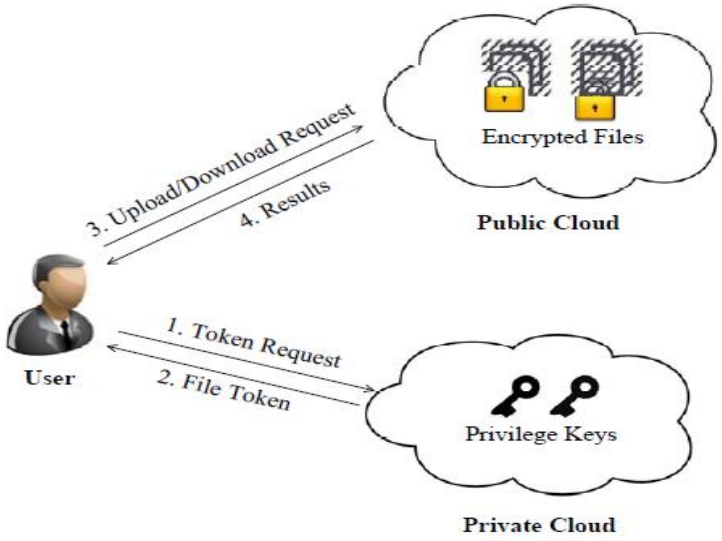

Figure 24. System Architecture

Data Flow Diagram

- The diagram is represented as a bubble flow chart. It acts as a graphical representation of a system that has input, processing, and output.
- Though being common, it is greatly confused. It is considered as a system procedure that processes. It is also an external entity that interconnects with the system and information flows in the system.

- The diagram is effective in providing an illustration of a system, particularly in abstraction. It is usually portioned into levels that symbolize the rising information flow. It also represents the function details. It illustrates how information flows in the system and the manner it is changed by a series of transformations (Figure 25).

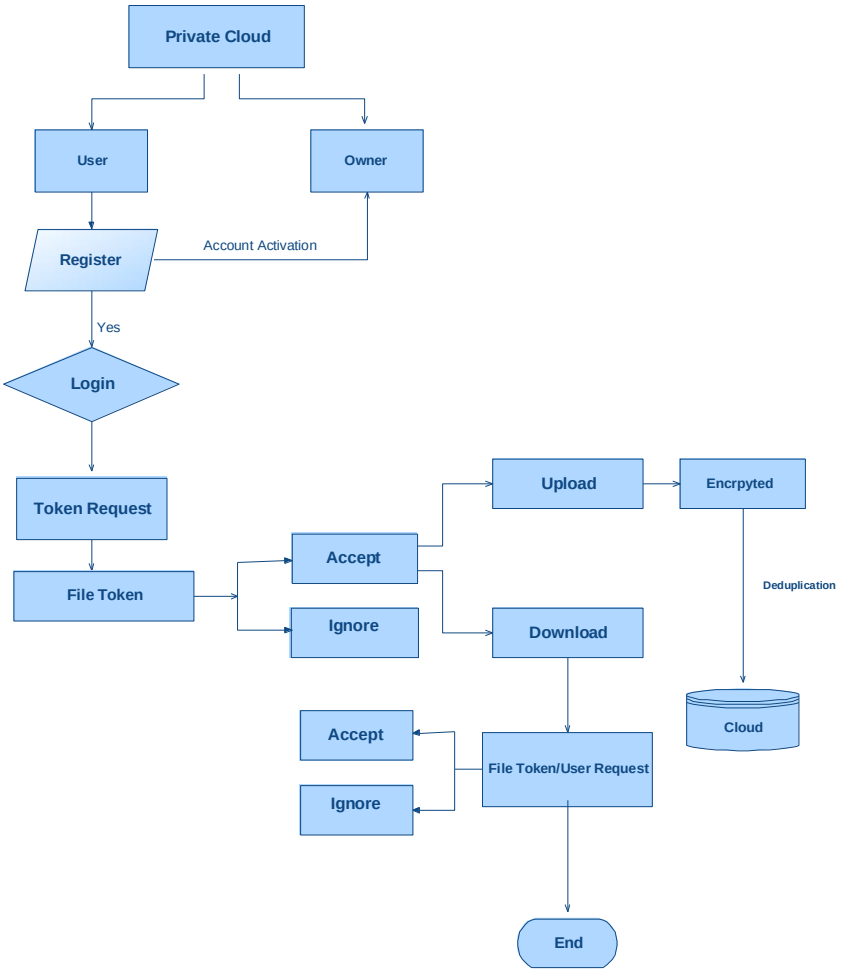

Figure 25. Data Flow Diagram

UML Diagrams

UML in full is unified modeling language. It is a standardized universal modeling language in object-oriented software engineering. The object management group manages the standard. The main objective of the language is to be a standardized language for developing the

different versions of object-oriented computer software. Currently, UML involves two main mechanisms—a Meta-model and notation. In most cases, it uses graphical notations to show the design of software projects. Its goals are:

1. Give consumers a product that is ready-to-use. It is also an expressive visual modeling language that can construct and come up and exchange meaningful models.
2. Give extensibility and specialty devices to lengthen the fundamental ideas.
3. They aim at providing autonomous programming languages and the development procedures.
4. They also give an official basis for comprehending the modeling language.
5. It encourages the development of the object-oriented device market.
6. It aims purposes at giving support to the design and making of high-level concepts like collaborations.

Use Case Diagram

In UML, this diagram is a form of representation that is described and developed from a use case analysis. Its main objective is to illustrate giving a graphical representation of functionality is provided by a system. It shows how the system function are carried out by a particular actor whose tasks are shown in the Figure 26.

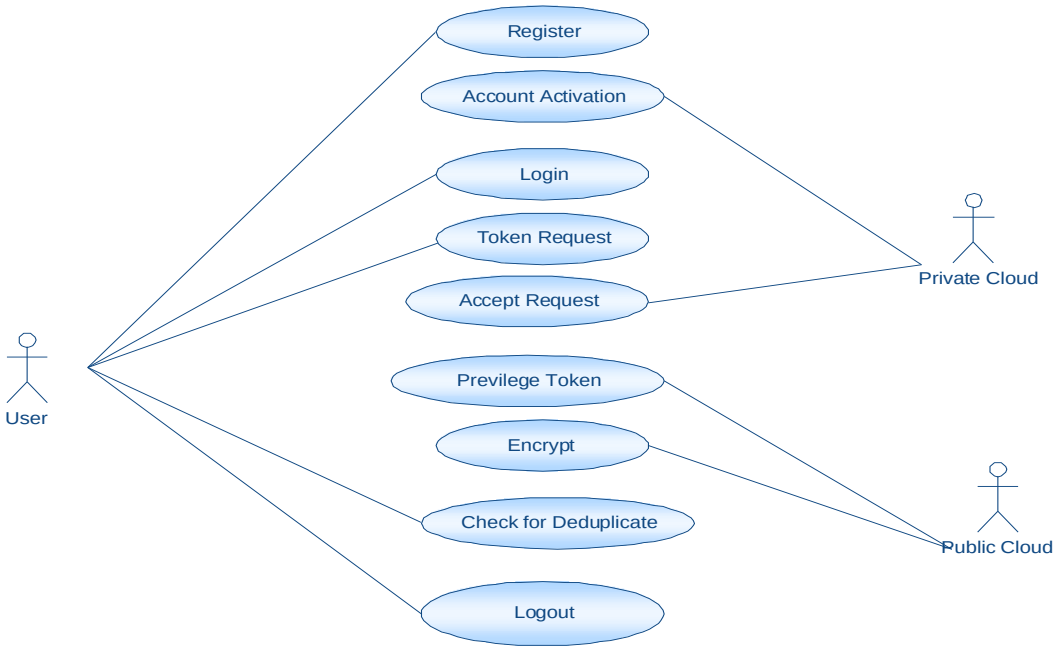

Figure 26. Case Diagram

Class Diagram

In UML, this diagram is responsible for giving a description of the structure of a system that illustrates system's classes, characteristics, processes, and the relations between the classes. It illuminates which class has info (Figure 27).

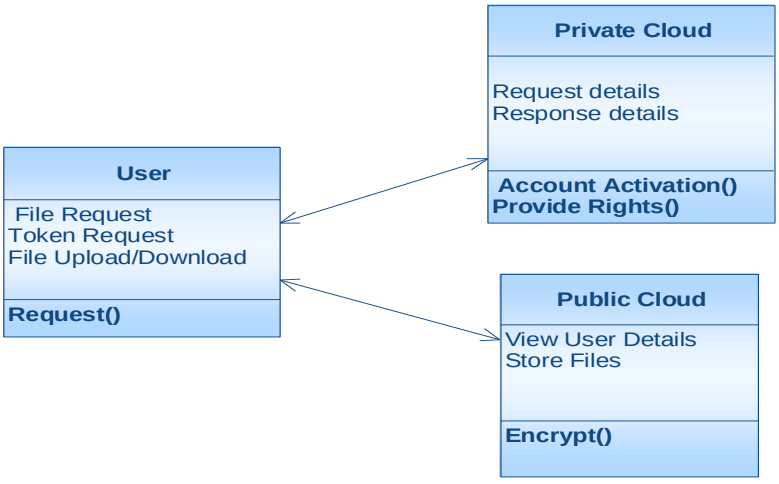

Figure 27. Class Diagram

Sequence Diagram

In the UML, the diagram is a form communication diagram that illustrates how the procedure functions with one another and in what sequence. It is a buildup of a Message Sequence Chart. They are sometimes known as event diagrams (Figure 28).

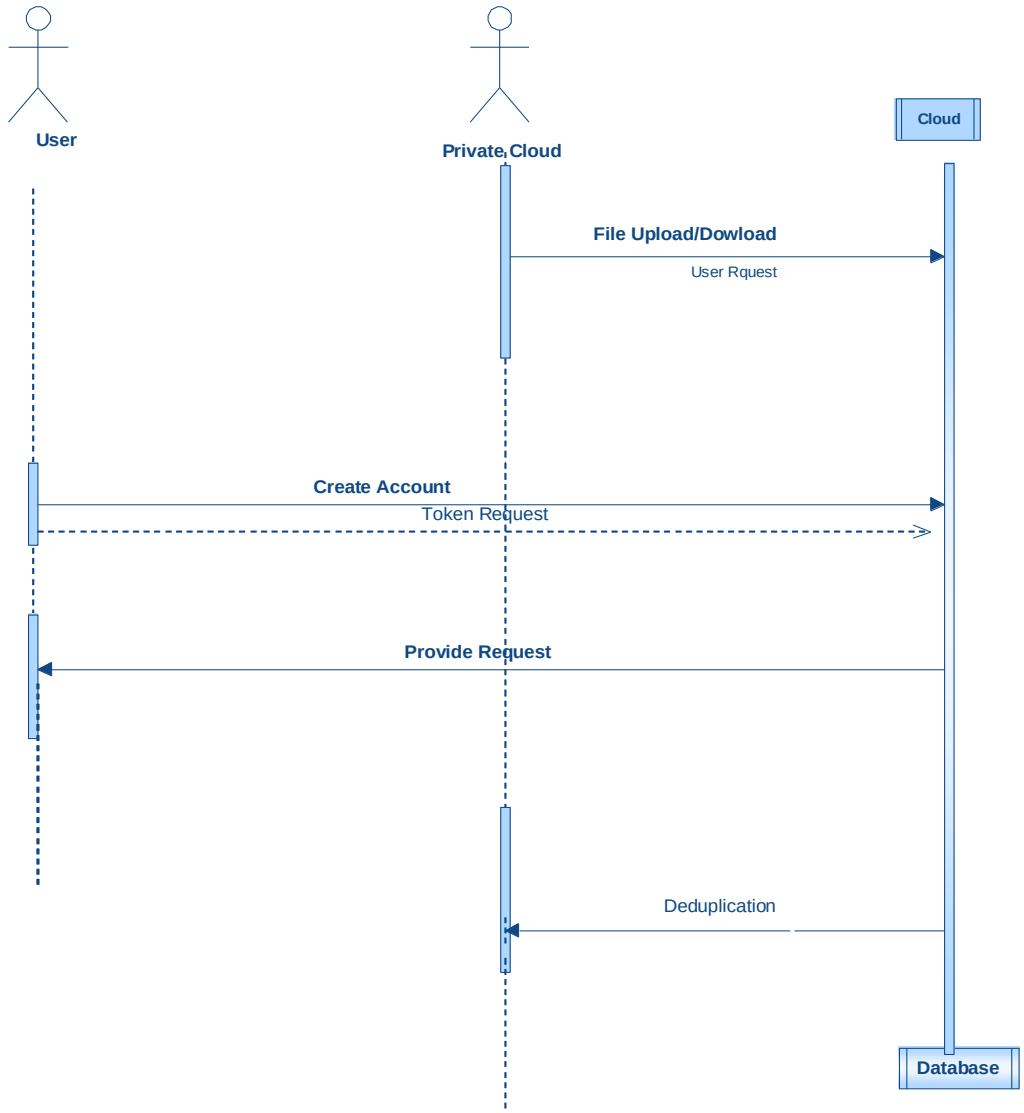

Figure 28. Sequence Diagram

Activity Diagram

The diagram in Figure 29 is a representation of the workflows of stepwise activities with the aid of selection, repetition, and concurrency. In UML, this diagram is normally utilized can be used define the enterprise and the operational step-by-step workflows of components in the system. It displays the entire flow of control.

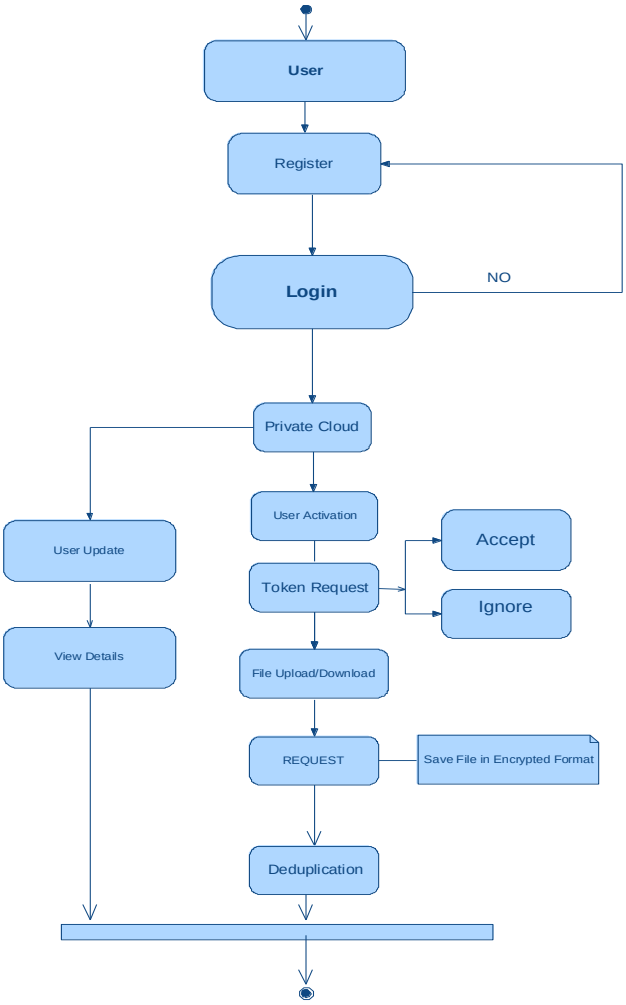

Figure 29. Activity Diagram

Summary

Different types of system designs have been explained in this chapter. System design is the process of describing and designing the project according to the user requirements. In the data processing industry system design plays a crucial role, it helps in building a modular system by standardizing hardware and software.

Chapter VII: System Testing

Introduction

System testing is to identify any potential errors. These testing attempts are performed in every possible way to find out every single minor and a major flaw in a work product. Testing provides an opportunity to inspect the effectiveness of sections, sub-congregations, assemblies and/or finished product software systems, and whether it lives up to the requirements and the customer demands and does not fall flat in a prohibited manner.

Types of System Testing

There are different ways of testing, and every testing method addresses a particular flaw. The following are the forms of tests:

- Unit Testing
- Integration Testing
- Functional Test
- System Test
- White Box Test
- Black Box Test

Test Strategy and Approach

The field-testing will be carried out substantially, and practical tests will be comprised of points of interest.

Test objectives:

- All field entries must work legally.
- Pages must be actuated from the well-known link networks.

- The entry section screen, messages, and reactions must not be deferred.

Features to be tested:

- Authenticate that the sections are of the correct configuration
- No copy sections ought to be permitted

All connections must take the customers to the correct locations.

Unit Testing

Unit testing is usually an automated process in which the smallest part, which is called a unit, is scrutinized and tested individually for proper operation. The purpose of this testing is for proper validation of software performance as it was designed.

A unit is the tiniest testable portion of a piece of software, which may have multiple inputs but a single output. In a procedural programming language, a unit might be an individual program, function or process. In an object-oriented programming language, the smallest unit is a procedure. It belongs to three classes; base, super, abstract classes. This is a first level testing which is also called primary testing. This generally has led to the main feature of combined code and the unit test period of the product life cycle, which typically has two unique stages.

Integration Testing

This is software testing in which all the individual units are combined as groups and then tested. The major aim of this testing is to identify the interactions amongst units and groups of units.

This integration testing is classified as Component Integration Testing and System Integration Testing.

Component integration testing. This software testing tests the interactions between individual components.

System Integration testing. This tests the interactions between the systems and packages. The importance of the incorporation testing is to authenticate practical, performance, and consistency requirements placed on major designed items. This integration technique is carried out in accordance with the software development life cycle (SDLC). The reliance on this testing are scheduled for incorporation testing, approach and chosen of tools utilized for incorporation.

Diverse forms of integration techniques are:

- Big-Bang
- Top-down
- Bottom-up
- Mixed (Sandwich)
- Risky-Hardest
- Collaboration integration
- Backbone integration
- Layer integration
- Client- server integration
- Distributed services integration
- High-frequency integration

Big bang integration. The modules that have been developed are integrated to come up with an entire software system making it efficient in time-saving. It also makes the testing process effective. Since the testing is a kind of "Usage model testing," it can be used in software and software testing. The testing runs users like capacities in a joined user like environment. The goal of this technique is to identify the difficulties brought about by a collaboration of components in the setting.

Top down testing. In this type of testing, the top integrated modules are tested. This is then followed by the step-by-step testing of the interrelated modules.

***Bottom up testing*.** When undertaking this testing, the lower levels testing are first tested to pave the way for the high levels. This process repeats up until the top of the hierarchy is tested. This technique assists to know the levels of software developed. It also makes it easier to come up with a percentage figure for the testing.

Mixed testing.This is also called Sandwich testing. It combines both top and bottom testing.

Functional testing. Functional testing depends upon the specifications of software components under test. This is a type of quality assurance testing; it describes what a system does, this tests a slice of the functionality of a whole system. It authenticates a program by inspecting it against design documents or specifications. This testing involves the following steps:

- Identifying the functions of software
- Creating an input database depending on their functional specifications
- Output determination on functional specifications
- Test case execution
- Actual and expected output comparison
- Check if the software fulfills the customer needs

The functional testing is of following types:

1. Smoke testing. Smoke testing is the preliminary test performed in the software testing; in this, simple testing failures are identified. This is also called an Intake test.

2. Sanity testing. Sanity testing is a basic test to evaluate results of calculation, which can be possibly true.
3. Regression testing. Regression testing is done for a previously developed software, it is done to test if the software still works accurately even after the interphase change.
4. Usability testing. Usability testing is done on a user interphase to evaluate the product in user interphase.

Functional tests systematically show that volumes tested are reachable as indicated by the trade and specific basics, system records, and customer guides. Functional testing is centered on the following items:

- Valid Input: recognized classes of valid input must be acknowledged.
- Invalid Input: well-known classes of invalid input must be vetoed.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedures: interfacing systems or process must be entreated.

System testing. System testing is a process that involves the evaluation of both software and hardware to determine if they are in line with requirements provided. It gives an assurance that the whole framework meets the required necessities. It tests a setup to assure known and unexpected results. A case of system testing is the design situated framework integration test. System testing is dependent on the process illustrations and streams, accentuating pre-driven process networks and integration focuses.

White box testing. This is a type of test in which the product analyzer has knowledge of the inward workings, framework, and language of the product. It is the reason that is used to test

zones that can't become two from a finding level. A static White box testing is a technical review; by conducting a technical review, early defects of software are spotted. Major characteristics of a technical review:

- These reviews are documented, and technical specialists identify the defects as a part of review process.
- This process does not involve management participation.
- It is led by trained moderators, not an author.
- A final report is prepared with a list of issues that need to be addressed.

Black box testing.This is a type of testing that involves evaluation of the system with no knowledge of its inward workings and framework. It is different from the other tests, therefore, it must have a conclusive archive. Its difference comes from the fact that the product being tested is taken as a discovery. The test gives information sources and reacts to yield without taking into consideration the manner in which the product functions. Common black box testing methods are:

- Decision table testing
- All-pairs testing
- Equivalence partitioning
- Boundary value analysis
- Cause effect graph
- Error guessing
- State transition testing
- Domain analysis
- Combining technique

Integration testing. Programming software integration testing is the increasing integration testing more than two coordinated programming sections on a solitary stage to come up with interface deserts. The purpose of the integration test is to watch that segments or

programming software applications, for example, segments in a product framework for one stage up programming applications at the organization level to collaborate without mistakes.

Test Results: The tests mentioned were all successful. None had a defect.

Acceptance testing. After the tests have been carried out, the user must accept it. This is the most important part of the test, as a result, it needs a great deal of participation particularly by the end user. It also makes sure that the system is in accordance with the stipulated requirements.

Test Results: The tests mentioned were all successful. None had a defect.

Summary

In this chapter, different types of system testing are mentioned briefly. System testing is performed to understand the functional specifications of a system and system requirements of the system. This testing is done to evaluate the system with its specific requirements. Different software testing is being performed by the proposed software, and the results are explained briefly in this chapter.

Chapter VIII: Conclusion

Introduction

In this chapter, the entire paper has been explained in short with a conclusion of why and how this study tried to prove to solve the problem statement. The possible future study and the discussion of what possible study can be done have been explained here.

Conclusion

In this study, the concept of sanctioned data de-duplication was suggested to guarantee data security. Other that assuring on data security, the framework includes variance profits of customers in the copy check. It also showed a few new de-duplication advances aiding approved copy check in a half-breed cloud design. In this design, the copy check tokens of information are established using the private cloud server with private keys. The security evaluation displays that plans are protected as far as inside and outside case assaults designated in the projected security model. As evidence of an idea objectified a trial product of a suggested and permitted copy check plan and lead tested experiments on the prototype. It also demonstrated the accepted copy check plan brings about insignificant overhead juxtaposed with the United encryption and system exchange.

Future Work

The data de-duplication technique used in this research paper is implemented in due course. For future work, the file that is uploaded to the cloud can be checked for integrity with the files that already exist in the cloud

References

Anderson, P., & Zhang, L. (2010). Fast and secure laptop backups with encrypted de-duplication. In *Proceedings of USENIX LISA, 2010*. San Jose, CA.

Bellare, M., Keelveedhi, S., & Ristenpart, T. (2013a). Message-locked encryption and secure deduplication. In *Proceedings of EUROCRYPT,* Athens, Greece, 296-312.

Bellare, M., Keelveedhi, S., & Ristenpartr, T. (2013b). Server-aided encryption of deduplicated stores. Symposium conducted at the *USENIX Security Symposium*., Washington, DC.

Bellare, M., Namprempre, C., & Neven, G. (2009). Security proofs for identity-based identification and signature schemes. *J. Cryptology, 22*(1):1-61.

Bugiel, S., Nurnberger, S., Sadeghi, A., & Schneider, T. (2011). Twin clouds: An architecture for secure cloud computing. In *Workshop on Cryptography and Security in Clouds (WCSC).*

Halevi, S., Harnik, D., Pinkas, B., & Shulman-Peleg, A. (2011). Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, & V. Shmatikov, (Eds.), *ACM Conference on Computer and Communications Security*, 491-500.

Li, J., Chen, X., Li, M., Li, J., Lee, P., & Lou, W. (2013). Secure deduplication with efficient and reliable convergent key management. In *IEEE Transactions on Parallel and Distributed Systems.*

Ng, C., & Lee, P. (2013). *Revdedup: A reverse deduplication storage system optimized for reads to latest backups*. Retrieved from http://www.cse.cuhk.edu.hk/~pclee/www/pubs/apsys13.pdf

Ng, W. K., Wen, Y., & Zhu, H. (2012). Private data deduplication protocols in cloud storage. In S. Ossowski & P. Lecca, (Eds.), *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, Riva, Italy, 441-446.

Pietro, R. D., & Sorniotti, A. (2012). Boosting efficiency and security in proof of ownership for deduplication. In H. Y. Youm and Y. Won (Eds.), *ACM Symposium on Information, Computer and Communications Security*, Seoul, Republic Korea, 81-72.

Quinlan, S., & Dorward, S. (2002). *Venti: A new approach to archival storage*. Retrieved from https://www.usenix.org/legacy/events/fast02/quinlan/quinlan_html/

Rahumed, H. C. H., Chen, Y., Tang, P., Lee, C., & Lui, J. C. S. (2011). A secure cloud backup system with assured deletion and version control. *In 3rd International Workshop on Security in Cloud Computing,* Shanghai, China.

Stanek, J., Sorniotti, A., Androulaki, E., & Kencl, L. (2013). *A secure data deduplication scheme for cloud storage*. Retrieved from http://ale.sopit.net/pdf/dedup.pdf

Storer, M. W., Greenan, K., Long, D. D. E., & Miller, El L. (2008). Secure data deduplication. In *Proceedings of StorageSS*, Alexandria, VA.

Wilcox-O-Hearn, Z., & Warner, B. (2008). Tahoe: The least-authority file system. In *Proceedings of ACM StorageSS,* Alexandria, VA.

Xu, J., Chang, E-C., & Zhou, J. (2013). *Weak leakage-resilient client-side deduplication of encrypted data in cloud storage.* Retrieved from https://www.comp.nus.edu.sg/~changec/publications/2013_asiaccs.pdf

Zhang, K., Zhou, X., Chen, Y., Wang, X., & Ruan, Y. (2011). Sedic: Privacy aware data-intensive computing on hybrid clouds. In *Proceedings of the 18th ACM conference on Computer and communications security*, New York, NY, 515-526.