**SYNOPSIS**

The use of cloud computing has increased rapidly in many organizations. Cloud computing provides many benefits in terms of low cost and accessibility of data.

Ensuring the security of cloud computing is a major factor in the cloud computing environment, as users often store sensitive information with cloud storage providers but these providers may be untrusted.

Distributing data over different cloud storage providers (CSPs) automatically provides users with a certain degree of information leakage control, for no single point of attack can leak all the information. However, unplanned distribution of data chunks can lead to high information disclosure even while using multiple clouds.

Dealing with “single cloud” providers is predicted to become less popular with customers due to risks of service availability failure and the possibility of malicious insiders in the single cloud. A movement towards “multi-clouds”, or in other words, “interludes” or “cloud-of-clouds” has emerged recently. This paper surveys recent research related to single and multi-cloud security and addresses possible solutions.

It is found that the research into the use of multi-cloud providers to maintain security has received less attention from the research community than hasthe use of single clouds. This work aims to promote the use of multi-clouds due to its ability to reduce security risks that affect the cloud computing user.

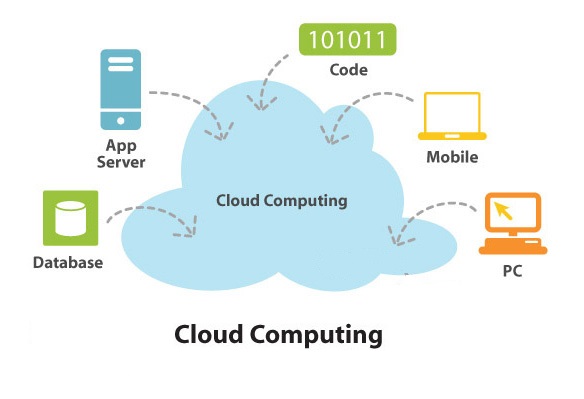
**INTRODUCTION**

Today, we have an ability to utilize scalable, distributed computing environments within the confines of the Internet, a practice known as cloud computing. In this new world of computing, users are universally required to accept the underlying premise of trust. Within the cloud computing world, the virtual environment lets users’ access computing power that exceeds that contained within their own physical worlds. Cloud computing is the process of providing computer facilities via internet. And it’s provided us better and efficient way to access information in timely manner and also increases storage of capacity for user in.

Cloud computing enables a new business model that supports on-demand, pay for-use, and economies-of-scale IT services over the Internet. The Internet cloud works as a service factory built around virtualized data centers.1 Cloud platforms are dynamically built through virtualization with provisioned hardware, software, networks, and datasets. The idea is to migrate desktop computing to a service-oriented platform using virtual server clusters at data centers. However, a lack of trust between cloud users and providers has hindered the universal acceptance of clouds as outsourced computing services. To promote multitenancy, we must design the cloud ecosystem to be secure, rust worthy, and dependable. In reality, trust is a social problem, not a purely technical issue. However, we believe that technology can enhance trust, justice, reputation, credibility, and assurance in Internet applications. To increase the adoption of Web and cloud services, cloud service providers (CSPs) must first establish trust and security to alleviate the worries of a large number of users.

Cloud users are anxious about whether data-center owners will misuse the system by arbitrarily using private datasets or releasing sensitive data to a third party Without permission. Cloud security is deployed to provide full of protection between data owner and service provider. To address these issues, we propose a reputation-based trust-management scheme augmented with data coloring and software watermarking.

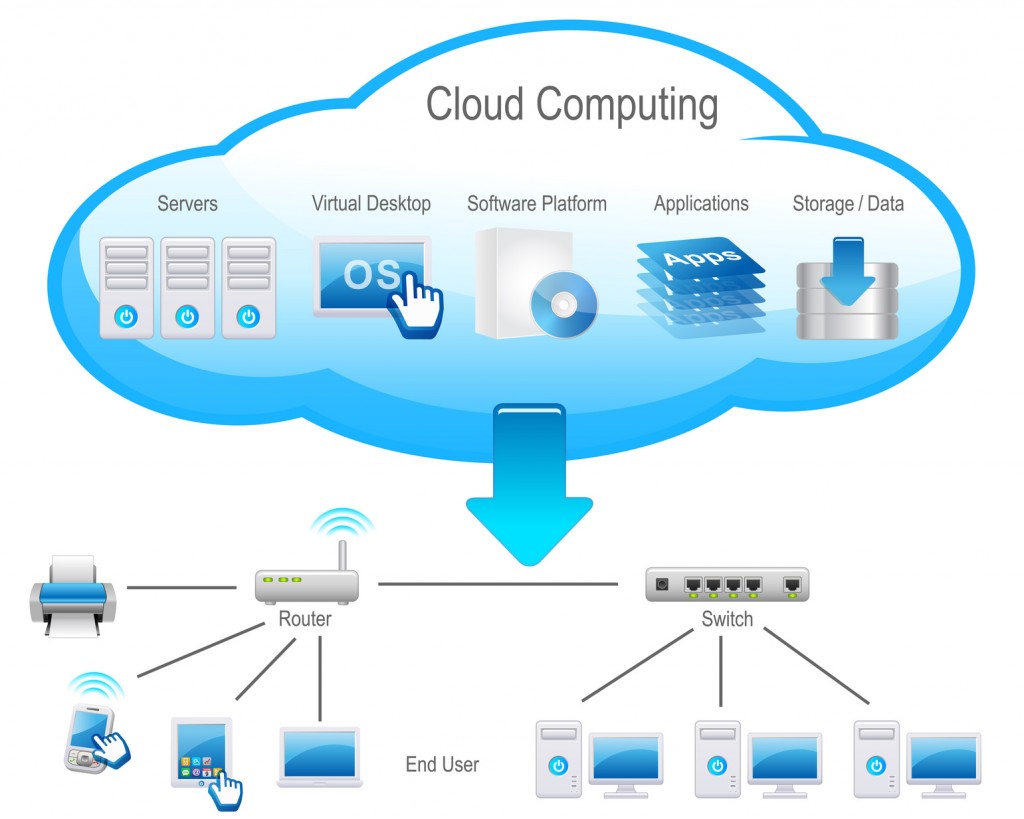
The Cloud Security Alliance 5 has identified a few critical issues for trusted cloud computing, and several recent works discuss general issues on cloud security and privacy. Public and private clouds demand different levels of security enforcement. We can distinguish among different service-level agreements (SLAs) by their variable degree of shared responsibility between cloud providers and users. Critical security issues include data integrity, user confidentiality, and trust among providers, individual users, and user groups. The three most popular cloud service models have varying security demands; the infrastructure-as-a-service (IaaS) model sits at the innermost implementation layer, which is extended to form the platform-as-a-service (PaaS) layer by adding OS and middleware support. PaaS further extends to the software as-a-service (SaaS) model by creating applications on data, content, and metadata using special APIs. This implies that SaaS demands all protection functions at all levels. At the other extreme, IaaS demands protection mainly at the networking, trusted computing, and compute/storage levels, whereas PaaS embodies the IaaS support plus additional protection at the resource-management level.

****

**Cloud Structure**

**Cloud Services:**

The cloud security grouping has identified a few vital issues for trusted cloud computing, and several recent works discuss general issues on cloud security and privacy. Public and private clouds order different levels of security enforcement. We can differentiate among different service-level agreements (SLAs) by their variable degree of shared responsibility of Security concerns which contains data integrity, user confidentiality, and trust among providers, individual users, and user groups. The three security demands should be varied from the three cloud service models that are described in below the infrastructure-as-a-service model sits at the innermost implementation layer, which is expanded to form the platform-as-a service (PaaS) layer by adding OS and middleware support. Pass further extends to the software-as-a-service (SaaS) model by creating applications on data, content, and meta-data using special A PIs. This implies that SaaS demands all protection functions at all levels.



**Cloud Services**

**Securing transportation as a service**

The Iaas model is works to compute networking and data storage, other resources in a virtualized environment. Amazon’s Elastic Compute cloud is one of good example of Iaas, at the cloud infrastructure level, CSP can implement network security with intrusion-detection systems, firewalls, antivirus programs, distributed denial-of-service suspicion and so on. Securing policy as a service. Cloud platforms built in Iaas with system integration and virtualization middle-ware. And these platforms can be used to users for implementing user-built software applications onto the cloud infrastructure using provider-supported programming languages and software tools like Java, Python and Dot net.

**Cloud Suppliers and Reported Services**

Hardware and software features are presents in cloud security. Three main security requirements is used in cloud computing demands such as confidentiality, integrity and availability. Report is the process of maintaining user communicated details on database. And these details are viewed only by an authenticated user in cloud environments

**Data reliability and confidentiality Protection**

Data integrity is the ability of cloud provider keep data safe from unauthorized person or hackers. Confidentiality is essentially the way the cloud provider insures that the data is secured from unauthorized access. The measures that the cloud provider uses to insure that this goal is met include physical isolation and cryptology. Cloud computing is a public network, which brings a set of complicated challenges for the provider to produce isolation for the customer. Physical isolation is accomplished by using virtual local area networks and middle boxes. The second method the provider uses is cryptology, which essentially encrypts the data before it is placed into the cloud. These two methods are standard measures that are used to secure data in the cloud.

Many tools are available for constructing cloud applications on large datasets and it’s provided by cloud software environment to desired users. Let’s following features are presents in security and privacy such as. Fine-grained access control to conserve data integrity and deter intruders or hackers a method to stop ISPs or CSPs from attacking user privacy. CSPs that struggle against spyware and web bug. We can improve some of these features with cloud reputation systems and more efficient identity management systems some features are Cloud resource can access security protocol like http and secure socket layer. Fine grained access control to protect data integrity and data attacker.

**Trusted Cloud Computing over Data Centers**

Security aware cloud architecture and this used to identify the protection mechanisms needed. Intruder detection action is should be implemented by using these architecture.

**Cloud protection Infrastructure**

Trusted and dependable cloud architecture helps protect network attack by launching trusted operational sectors for different cloud applications. The difficulties in security agreement that CSPs protect all data center servers and storage areas. Our architecture protects VM checking from software based attacks and upholder data and information from robbery, fraud and natural failures. It provides strong authentication and authorized access to sensitive data and on-demand services. We had several design objectives for a trusted and dependable cloud when creating our architecture.

**OVERVIEW**

**Benefits of cloud computing**

Cloud computing has a range of defining features (which make a general definition elusive5), namely

* Hardware (computers, storage devices) is owned by the cloud computing provider, not by the user who interacts with it via the internet.
* The use of hardware is dynamically optimized across a network of computers, so that the exact location of data or processes, as well as the information which piece of hardware is actually serving a particular user at a given moment, does not in principle have to concern the user, even though it may have an important bearing on the applicable legal environment.
* Cloud providers often move their users' workloads around (e.g. from one computer to another or from one data centre to another) to optimize the use of available hardware;
* The remote hardware stores and processes data and makes it available, e.g. through applications so that a company could use its cloud-based computing in just the same way as consumers already today use their webmail accounts.
* Organizations and individuals can access their content, and use their software when and where they need it, e.g. on desktop computers, laptops, tablets and smart phones. Actions for ICT Employment", annex to the Commission Communication "Towards a job-rich recovery", COM (2012) 173 final. The organizations may worry about business continuity in the case of service disruption whereas individuals may have concerns about what happens with their personal information. Such worries slow down the overall speed of adoption of cloud computing. 5 Many such definitions are highly abstract: One well-known definition speaks of "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources … that can be rapidly provisioned and released with minimal effort or service provider interaction" NIST (2009), US National Institute for Standards and Technology.
  + A cloud set-up consists of layers: hardware, middleware or platform, and application software. Standardizations are important especially at the middle layer because it enables developers to address a wide range of potential customers and gives users choice.
  + Users normally pay by usage, avoiding the large upfront and fixed costs necessary to set up and operate sophisticated computing equipment.

At the same time, users can very easily modify the amount of hardware they use bring new storage capacity online in a matter of seconds with a few mouse clicks. Consumers can use cloud services to store information (e.g. pictures or e-mail) and to use software. The social networks, streamed video and music, and games. Organizations, including public administrations, can use cloud services to successively replace internally run data centers and information and communication technology (ICT) departments. Companies can use cloud services to quickly test and scale up what they offer to their customers because they can do so without investing in and building physical infrastructures. Overall, cloud computing represents a further industrialization (standardization, scaling-up, wide-spread availability) of the provision of computing power ("utility computing") in the same way as power plants industrialized the provision of electrical power. Thanks to standardized interfaces (the equivalent to electrical power plugs) users can leave the details (how to build, power, run and secure a data center) to experts who achieve much better economies of scale (by serving many users) than individual users ever could. Moreover, cloud services offer very large economies of scale meaning that go-it-alone efforts at national level are unlikely to deliver optimal cost efficiencies. The benefits of adopting cloud computing can be illustrated by a 2011 survey for the Commission which shows that as a result of the adoption of cloud computing 80% of organizations reduce costs by 10-20%. Other benefits include enhanced mobile working (46%), productivity (41%), standardization (35%), as well as new business opportunities (33%) and markets (32%).6 All available economic studies also confirm the importance of cloud computing which is expected to grow rapidly worldwide.7 The unprecedented increase of data flow and processing of information over the Internet has a significant environmental impact through energy and water consumption, and greenhouse gas emissions. Cloud computing can help mitigate these problems thanks to more efficient use of hardware as well as, more specifically, by building data centers to use low-energy servers and green energy.8 For example, according to some estimates, large companies in the US could save $12.3 billion annually in energy consumption by adopting cloud computing.

**SYSTEM ANALYSIS**

**EXISTING SYSTEM**

The use of cloud computing has increased rapidly in many organizations. Cloud providers should address privacy and security issues as a matter of high and urgent priority. Distributing data over different cloud storage providers. (CSPs) automatically provides users with a certain degree of information leakage control, for no single point of attack can leak all the information. However, unplanned distribution of data chunks can lead to high information disclosure even while using multiple clouds. To optimize the information leakage, we presented the Store, an information leakage aware storage system in the multi-cloud. An effective storage plan generation algorithm based on clustering for distributing data chunks with minimal information leakage across multiple clouds.

**Problem description in existing system:**

* The loss of service availability has caused many problems for a large number of customers recently.
* Single cloud providers are predicted to become less popular with customers due to risks of service availability failure.
* The data stored in the cloud may suffer from damage during transition operations from or to the cloud storage provider.
* When multiple clients use cloud storage or when multiple devices are synchronized by one user, it is difficult to address the data corruption issue.

**PROPOSED SYSTEM**

In multiple cloud system we improve security while transfer the data from one end to another. Attackers may hack our message during communication between two users. To avoid this theft we perform encryption. By using this technique we convert our original message into another format. In receiver end, we perform decryption to get the original data. Moving from single clouds or inner-clouds to multicourse is reasonable and important for many reasons. Services of single clouds are still subject to outage. In addition, showed that over 80% of company management fear security threats and loss of control of data and systems. Assumes that the main purpose of moving to interludes is to improve what was offered in single clouds by distributing reliability, trust, and security among multiple cloud providers. In addition, reliable distributed storage which utilizes a subset of BFT techniques was suggested to be used in multi-clouds. A number of recent studies in this area have built protocols for inter clouds. RACS for instance, utilizes RAID-like techniques that are normally used by disks and file systems, but for multiple cloud storage. Assume that to avoid “vender lock-in”, distributing a user’s data among multiple clouds is a helpful solution. This replication also decreases the cost of switching providers and offers better fault tolerance. Therefore, the storage load will be spread among several providers.

Authentication

Choose the data file

Send selected data

Encryption

Decryption

Retrieve original Data

**BLOCK DIAGRAM**

**Advantages**

* Small and medium companies use cloud computing services for various reasons, including because these services provide fast access to their applications and reduce their infrastructure costs.
* Cloud computing provides many benefits in terms of low cost and accessibility of data.
* The secret sharing algorithm to reduce the risk of data intrusion and the loss of service availability in the cloud and ensure data integrity.
* Solution can avoid data corruption caused by some components in the cloud.

**REQUIREMENT ANALYSIS**

**HARDWARE REQUIREMENTS**

PROCESSOR : core2duo

CLOCK SPEED : 3.0 GHZ

RAM SIZE : 2 GB

HARD DISK CAPACITY : 500 GB

MONITOR TYPE : 15 INCH COLOR MONITOR

KEYBOARD TYPE : INTERNET KEYBOARD

**SOFTWARE REQUIREMENTS**

OPERATING SYSTEM : WINDOWS XP PROFESSIONAL, 7

ENVIRONMENT : JAVA

BACKEND : Microsoft SQL SERVER 2005.

**SYSTEM DEVELOPMENT**

**JAVA INTRODUCTION:**

Java is an object-oriented programming language developed by Sun Microsystems and it is also a powerful internet programming language. Java is a high-level programming language which has the following features:

* Object oriented
* Portable
* Architecture-neutral
* High-performance
* Multithreaded
* Robust
* Secure

Java is an efficient application programming language. It has APIs to support the GUI based application development. The following features of java, makes it more suitable for implementing this project.

Initially the language was called as “OAK” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform independent language that could be used to create software to be embedded in various consumer electronic devices.

* Java is programmer’s language.
* Java is cohesive and consistent.
* Except for those constraints imposed by the internet environment, Java gives the programmer, full control.

**Types of Java Program**

**Applications**

An application is a program that runs on our computer under the operating system of that computer. It is more or less like on creating using C or C++. Java’s ability to create Applets makes it important.

**FEATURES**

**Security**

Every time you that the download a “normal” program, you are risking a viral infection. Prior to java, most users did not download executable programs frequently. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a “firewall” between a network application and your computer.

**Portability**

For programs to be dynamically downloaded to all the various types of platforms connected to the internet, some means of generating portable executable code is needed. As you will see, the same mechanism that helps ensure security also helps create portability. Indeed, java’s solution to these two problems is both elegant and efficient.

**The Byte code**

The key that allows the java to solve the security and portability problems is that the output of java compiler is byte code. Byte code is a highly optimized set of instructions designed to be executed by the java run – time system, which is called the java virtual machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

**Java Virtual Machine (JVM)**

Beyond the language, there is the java virtual machine. The java virtual machine is an important element of the java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and the byte code verification makes sure that the code that has been generated by the compiler will not corrupt the machine it is loaded on. Byte code verification takes place at the end of the compilation process to make sure that it is accurate and correct. So byte code verification is integral to the compiling and executing of java code.

**Java Architecture**

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the java virtual machine, which is then interpreted on each platform by the run-time environment. Java is dynamic system, able to load code when needed from a machine in the same room or across the planet.

**Compilation of code**

When you compile the code, the java compiler creates machine code (called byte code) for a hypothetical machine called java virtual machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

**Platform Independent**

Platform independence, that means the ability of a program to move easily from one computer system to another. It is one of the most significant advantages that Java has over other programming languages. Java is platform independent at both the source and the binary level.

**Networking Classes in the JDK**

Through the classes in java.net, Java programs can use TCP or UDP to communicate over the Internet. The Socket and Server Socket classes all use TCP to communicate over the network.

**Advantages**

* By using Java, one program can be run on many different platforms. This means that you do not need to put your efforts on developing a different version of software for each platform.
* There are many programmers who can understand and write code in Java, so that many people can participate in developing open source software.
* In many cases, a Java virtual machine can prevent an incorrectly written application program from causing problems to the rest of your computing environment.
* Macros
* Context-sensitive help and the Answer Wizard

**Microsoft Windows**

Windows XP is the phenomenally successful and ubiquitous operating system software by Microsoft Corporation. It is a multitasking operating system well equipped with a Graphical User Interface (GUI).

A task bar, always easily accessible on screen has buttons listing the currently running applications, allowing the user to easily switch between them. Windows XP is supported to make personal computers and laptops easier to learn and exploit.

**SYSTEM DESIGN**

**INPUT DESIGN**

Input Design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system. Needless to say, therefore, that the input data is the life blood of a system and have to be analyzed and designed with utmost care and consideration. The decisions made during the input design are

* To provide cost effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is understand by the user

System analysis decide the following input design details like, what data to input, what medium to use, how the data should be arranged or coded, data items and transactions needs validations to detect errors and at last the dialogue to guide user in providing input.

Input data of a system may not be necessarily be raw data captured in the system from scratch. These can also be the output of another system or sub system.. The inputs involve identifying the data needed, specifying the characteristics of each data item, capturing and preparing data for computer processing and ensuring correctness of data.

**OUTPUT DESIGN**

Input Design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system. Needless to say, therefore, that the input data is the life blood of a system and have to be analyzed and designed with utmost care and consideration. The decisions made during the input design are

* To provide cost effective method of input.
* To achieve the highest possible level of accuracy.

System analysis decide the following input design details like, what data to input, what medium to use, how the data should be arranged or coded, data items and transactions needs validations to detect errors and at last the dialogue to guide user in providing input. Input data of a system may not be necessarily be raw data captured in the system from scratch. These can also be the output of another system or sub system.. The inputs involve identifying the data needed, specifying the characteristics of each data item, capturing and preparing data for computer processing and ensuring correctness of data.

* Mobility Support
* XML Web service Support

**DATA FLOW DIAGRAM**

GETTING AUTHENTICATION

SELECTE DATA TO BE SEND

ENCRYPTION

DECRYPTION

RETRIEVE ORIGINAL DATA

**Architecture of Transfer Data between client and server**

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

**Project Executive Summary**

This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. If appropriate, include the information discussed in the subsequent sections in the summary.

**System Overview**

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements trace ability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

**Design Constraints**

This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design.

**Input design**

Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blue print for the document phase.

Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified.

Design will explain software components in detail. This will help the implementation of the system. Moreover, this will guide the further changes in the system to satisfy the future requirements.

**OUTPUT DESIGN**

Form is a tool with a message; it is the physical carrier of data or information. It also can constitute authority for actions. In the form design files are used to do each module. The following are list of forms used in this project:

**Main Form**

Contains option for viewing face from data base. The system retrieves the images stored in the folder called train and test folder, which is available in bin folder of your application.

**View database form:**

This form retrieves face available in the train folder. It is just for viewing purpose for the user.

**Recognition form:** This form provides option for loading input data from particular folder. Then user has to click button which leads the application for training to gain knowledge as it is of the implementation form learning algorithm.

**SYSTEM TESTING**

**Testing**

This paper presently is used to the multi agent based gride resource discovering in the based on the three type of agent providing of the papers 1.Request agent 2. Mapper agent 3.Schedule agent used in the project request agent purpose enter the user details job id, job name, storage capacity, arrival time, service time in collectively of the data. Mapper agent catch the requirement on the identification of the particular details selectively of the processing of the genetic algorithm apply the information Processing has undergone major improvements in the past two decades in both hardware and software. Hardware has decreased in size and price, while providing more and faster processing power. Software has become easier to use, while providing increased capabilities.

There is an abundance of products available to assist both end-users and software developers in their work. Software testing, however, has not progressed significantly. It is still largely a manual process conducted as an art rather than a methodology. It is almost an accepted practice to release software that contains defects. Software that is not thoroughly tested is released for production. This is true for both off-the-shelf software products and custom applications. Software vendor and in-house systems developers release an initial system and then deliver fixes to the code. They continue delivering fixes until they create a new system and stop supporting the old one. The user is then forced to convert to the new system, which again will require fixes. In-house systems developers generally do not provide any better level of support. They require the users to submit Incident Reports specifying the system defects. The Incident Reports are then assigned a priority and the defects are fixed as time and budgets permit.

**Importance of testing**

Testing is difficult. It requires knowledge of the application and the system architecture. The majority of the preparation work is tedious. The test conditions, test data, and expected results are generally created manually. System testing is also one of the final activities before the system is released for production. There is always pressure to complete systems testing promptly to meet the deadline. Nevertheless, systems testing are important. In mainframe when the system is distributed to multiple sites, any errors or omissions in the system will affect several groups of users.

Any savings realized in downsizing the application will be negated by costs to correct software errors and reprocess information. Software developers must deliver reliable and secure systems that satisfy the user’s requirements. A key item in successful systems testing is developing a testing methodology rather than relying on individual style of the test practitioner. The systems testing effort must follow a defined strategy. It must have an objective, a scope, and an approach. Testing is not an art; it is a skill that can be taught.

Testing is generally associated with the execution of programs. The emphasis is on the outcome of the testing, rather than what is tested and how it’s tested. Testing is not a one-step activity; execute the test. It requires planning and design. The tests should be reviewed prior to execution to verify their accuracy and completeness. They must be documented and saved for reuse. System testing is the most extensive testing of the system. It requires more manpower and machine processing time than any other testing level. It is therefore the most expensive testing level. It is critical process in the system development. It verifies that the system performs the business requirements accurately, completely, and within the required performance limits. It must be thorough, controlled and managed.

**TYPES OF TESTING**

Software development has several levels of testing.

1. Unit Testing
2. Integration Testing
3. Systems Testing
4. Acceptance Testing

## Unit Testing

The first level of testing is called unit testing which is done during the development of the system. Unit testing is essential for verification of the code produced during the coding phase. Errors were been noted down and corrected immediately. It is performed by the programmer. It uses the program specifications and the program itself as its source. Thus, our modules are individually tested here. There is no formal documentation required for unit-testing program.

## Integration Testing

The second level of testing includes integration testing. Here different dependent modules are assembled and tested for any bugs that may surface due to the integration of modules. Thus, the administrator module and various visa immigration modules are tested here.

**Systems Testing**

The third level of testing includes systems testing. Systems testing verify that the system performs the business functions while meeting the specified performance requirements. It is performed by a team consisting of software technicians and users. It uses the Systems Requirements document, the System Architectural Design and Detailed Design Documents, and the Information Systems Department standards as its sources. Documentation is recorded and saved for systems testing.

## Acceptance Testing

The final level of testing is the acceptance testing. Acceptance testing provides the users with assurance that the system is ready for production use; it is performed by the users. It uses the System Requirements document as its source. There is no formal documentation required for acceptance testing.

Systems testing are the major testing effort of the project. It is the functional testing of the application and is concerned with following,

1. Quality/standards compliance
2. Business requirements
3. Performance capabilities
4. Operational capabilities

Below are defined a few test cases which have been implemented for the various screens. The outputs have been registered and the required changes have been incorporated.

**SOURCE CODE**

**access.aspx**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class access : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(ConfigurationManager.AppSettings["ConnectionString"]);

SqlDataAdapter da;

DataSet ds;

protected void Page\_Load(object sender, EventArgs e)

{

if (con.State == ConnectionState.Closed)

{

con.Open();

}

if (Page.IsPostBack != true)

{

Getcustomers();

}

}

private void Getcustomers()

{

da = new SqlDataAdapter("select userid,uname,status,cdate from reg", con);

ds = new DataSet();

da.Fill(ds, "reg");

GridView1.DataSource = ds.Tables["reg"].DefaultView;

GridView1.DataBind();

}

protected void GridView1\_PageIndexChanging(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

Getcustomers();

}

protected void GridView1\_RowCommand(object sender, GridViewCommandEventArgs e)

{

if (e.CommandName == "Artist")

{

da = new SqlDataAdapter("select status from reg where userid=" + Convert.ToInt32(e.CommandArgument.ToString()) + " ", con);

ds = new DataSet();

da.Fill(ds, "reg");

if (ds.Tables.Count > 0 && ds.Tables["reg"].Rows.Count > 0)

{

if (ds.Tables["reg"].Rows[0][0].ToString() == "Activate")

{

da = new SqlDataAdapter("update reg set status='Deactivate' where userid=" + Convert.ToInt32(e.CommandArgument.ToString()) + " ", con);

int n = da.SelectCommand.ExecuteNonQuery();

if (n == 1)

{

Getcustomers();

}

}

else if (ds.Tables["reg"].Rows[0][0].ToString() == "Deactivate")

{

da = new SqlDataAdapter("update reg set status='Activate' where userid=" + Convert.ToInt32(e.CommandArgument.ToString()) + " ", con);

int n = da.SelectCommand.ExecuteNonQuery();

if (n == 1)

{

Getcustomers();

} } } } }

protected void GridView1\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

Label uid = new Label();

uid = (Label)GridView1.Rows[e.RowIndex].Cells[1].FindControl("userid");

if (uid.Text != "")

{

da = new SqlDataAdapter("delete from reg where userid=" + Convert.ToInt32(uid.Text) + " ", con);

int res = da.SelectCommand.ExecuteNonQuery();

if (res == 1)

{

Getcustomers();

}

}

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

**accessright.aspx**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class accessright : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(ConfigurationManager.AppSettings["ConnectionString"]);

SqlDataAdapter da;

DataSet ds;

protected void Page\_Load(object sender, EventArgs e)

{

if (con.State == ConnectionState.Closed)

{

con.Open();

}

if (Page.IsPostBack != true)

{

Getcustomers();

}

}

private void Getcustomers()

{

da = new SqlDataAdapter("select userid,uname,status,cdate from reg", con);

ds = new DataSet();

da.Fill(ds, "reg");

GridView1.DataSource = ds.Tables["reg"].DefaultView;

GridView1.DataBind();

}

protected void GridView1\_PageIndexChanging(object sender, GridViewPageEventArgs e)

{

GridView1.PageIndex = e.NewPageIndex;

Getcustomers();

}

protected void GridView1\_RowCommand(object sender, GridViewCommandEventArgs e)

{

if (e.CommandName == "Artist")

{

da = new SqlDataAdapter("select status from reg where userid=" + Convert.ToInt32(e.CommandArgument.ToString()) + " ", con);

ds = new DataSet();

da.Fill(ds, "reg");

if (ds.Tables.Count > 0 && ds.Tables["reg"].Rows.Count > 0)

{

if (ds.Tables["reg"].Rows[0][0].ToString() == "Activate")

{

da = new SqlDataAdapter("update reg set status='Deactivate' where userid=" + Convert.ToInt32(e.CommandArgument.ToString()) + " ", con);

int n = da.SelectCommand.ExecuteNonQuery();

if (n == 1)

{

Getcustomers();

}

}

else if (ds.Tables["reg"].Rows[0][0].ToString() == "Deactivate")

{

da = new SqlDataAdapter("update reg set status='Activate' where userid=" + Convert.ToInt32(e.CommandArgument.ToString()) + " ", con);

int n = da.SelectCommand.ExecuteNonQuery();

if (n == 1)

{ Getcustomers();

} } } }

}

protected void GridView1\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

Label uid = new Label();

uid = (Label)GridView1.Rows[e.RowIndex].Cells[1].FindControl("userid");

if (uid.Text != "")

{

da = new SqlDataAdapter("delete from reg where userid=" + Convert.ToInt32(uid.Text) + " ", con);

int res = da.SelectCommand.ExecuteNonQuery();

if (res == 1)

{

Getcustomers();

}

}

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

**alluser.aspx**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Default3 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void GridView1\_RowCommand(object sender, GridViewCommandEventArgs e)

{ }}

**Cloudadmin.aspx**

using System;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Data.SqlClient;

public partial class Default2 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

if (TextBox1.Text == "")

{

string myStringVariable1 = string.Empty;

myStringVariable1 = "Enter Cloud ID";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + myStringVariable1 + "');", true);

}

else

{

if (TextBox2.Text == "")

{

string myStringVariable1 = string.Empty;

myStringVariable1 = "Enter Cloud Password";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + myStringVariable1 + "');", true);

}

else

{

if (TextBox1.Text == "cloud" && TextBox2.Text == "cloud")

{

Session["adminid"] = TextBox1.Text;

Response.Redirect("cloudhome.aspx");

}

else

{

string myStringVariable1 = string.Empty;

myStringVariable1 = "Enter ID/Password Correctly.";

ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + myStringVariable1 + "');", true);

}

}

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

TextBox1.Text = "";

TextBox2.Text = "";

}

}

**Upload file**

using System;

using System.Collections;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Security.Cryptography;

using System.IO;

using System.Net;

using System.Net.Mail;

using System.Data.SqlClient;

using System.Text;

public partial class Default2 : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(ConfigurationManager.AppSettings["ConnectionString"]);

Class1 cs = new Class1();

Cryptography cr = new Cryptography();

FileInfo info;

//static

string mydocpath, fuploadstatus = "Finish";

string filename, filenamemain;

string tocken, binary1, binary, splitfilesize, tokenstatus;

long value;

string gMailAccount = "customerservice404@gmail.com";

string to;

string subject = "Token Keys";

string message;

string pwd1, sub;

string id;

private object size;

protected void Page\_Load(object sender, EventArgs e)

{

Label4.Text = Convert.ToString(cs.idgeneration1());

id = (string)Session["id"];

Label8.Text = (string)Session["FileName"];

filename = Label8.Text;

if (!IsPostBack)

{

for (int i = 0; i < 3; i++)

{

if (i == 0)

{

mydocpath = Request.PhysicalApplicationPath + "/UploadProcess/";

filenamemain = filename;

read1();

GenTocken();

//Label12.Text = tocken;

firstblockcontentlength();

}

}

}

}

private void firstblockcontentlength()

{

mydocpath = Request.PhysicalApplicationPath + "/UploadProcess/";

filenamemain = mydocpath + filename;

info = new FileInfo(filenamemain);

value = info.Length;

splitfilesize = Convert.ToString(value);

//HiddenField4.Value = Convert.ToString(splitfilesize);

GenBinary();

//HiddenField1.Value = Convert.ToString(binary1);

}

private void GenBinary()

{

binary1 = "";

for (int i = 0; i < splitfilesize.Length; i++)

{

char c = splitfilesize[i];

binary = Convert.ToString(c, 2);

binary1 = binary1 + binary;

}

}

private void GenTocken()

{

Random val = new Random();

int rno = val.Next(12345, 54321);

tocken = Convert.ToString(rno);

}

private void read1()

{

foreach (string txtName in Directory.GetFiles(mydocpath, filenamemain))

{

using (StreamReader sr = new StreamReader(txtName))

{

String line;

line = sr.ReadToEnd();

Console.WriteLine(line);

TextBox1.Text = line;

//TextBox2.Text = line;

//TextBox3.Text = line;

}

}

}

protected void Button1\_Click1(object sender, EventArgs e)

{

using (SqlConnection con = new SqlConnection(ConfigurationManager.AppSettings["ConnectionString"]))

{

using (SqlCommand cmd = new SqlCommand())

{

// cmd.CommandText = "insert into upload(id,fileid,name,title,date,filename,filetype,filedata) values(@id,@fileid,@name,@title,@date,@filename,@filetype,@filedata)";

cmd.CommandText = "insert into encrypt(fid,filename,document,fileencrypt,trapdoor,filetype)values(@fid,@filename,@document,@fileencrypt,@trapdoor,@filetype)";

cmd.Parameters.AddWithValue("@fid", Label4.Text);

cmd.Parameters.AddWithValue("@filename", Label8.Text);

cmd.Parameters.AddWithValue("@document", TextBox1.Text);

cmd.Parameters.AddWithValue("@trapdoor", TextBox3.Text);

cmd.Parameters.AddWithValue("@fileencrypt", TextBox2.Text);

// cmd.Parameters.AddWithValue("@reencrypt", TextBox3.Text);

cmd.Parameters.AddWithValue("@filetype", "application/word");

// cmd.Parameters.AddWithValue("@cdate", "getdate()");

//cmd.Parameters.AddWithValue("@filedata", size);

cmd.Connection = con;

con.Open();

cmd.ExecuteNonQuery();

con.Close();

Response.Redirect("success.aspx");

}

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

GenTocken();

// Label12.Text = cr.Encrypt(tocken);

//Image2.ImageUrl = "Images/blockicon.png";

TextBox2.Text = cr.Encrypt(TextBox1.Text);

TextBox2.ForeColor = System.Drawing.ColorTranslator.FromHtml("#DF0101");

TextBox2.BackColor = System.Drawing.ColorTranslator.FromHtml("#FEDADB");

}

protected void Button3\_Click(object sender, EventArgs e)

{

string Encryptfile = HttpUtility.UrlEncode(Encrypt(TextBox3.Text.Trim()));

//string username = HttpUtility.UrlEncode(Encrypt(txtName.Text.Trim()));

// Response.Redirect("~/DecryptQueryString.aspx?userid=" + userid + "&uname=" + username + "");

Console.WriteLine(Encryptfile);

TextBox3.Text = Encryptfile;

}

public static string Encrypt(string inputText)

{

string encryptionkey = "SDFRFTGER$^%\*HJHK";

byte[] keybytes = Encoding.ASCII.GetBytes(encryptionkey.Length.ToString());

RijndaelManaged rijndaelCipher = new RijndaelManaged();

byte[] plainText = Encoding.Unicode.GetBytes(inputText);

PasswordDeriveBytes pwdbytes = new PasswordDeriveBytes(encryptionkey, keybytes);

using (ICryptoTransform encryptrans = rijndaelCipher.CreateEncryptor(pwdbytes.GetBytes(32), pwdbytes.GetBytes(16)))

{

using (MemoryStream mstrm = new MemoryStream())

{

using (CryptoStream cryptstm = new CryptoStream(mstrm, encryptrans, CryptoStreamMode.Write))

{

cryptstm.Write(plainText, 0, plainText.Length);

cryptstm.Close();

return Convert.ToBase64String(mstrm.ToArray());

}

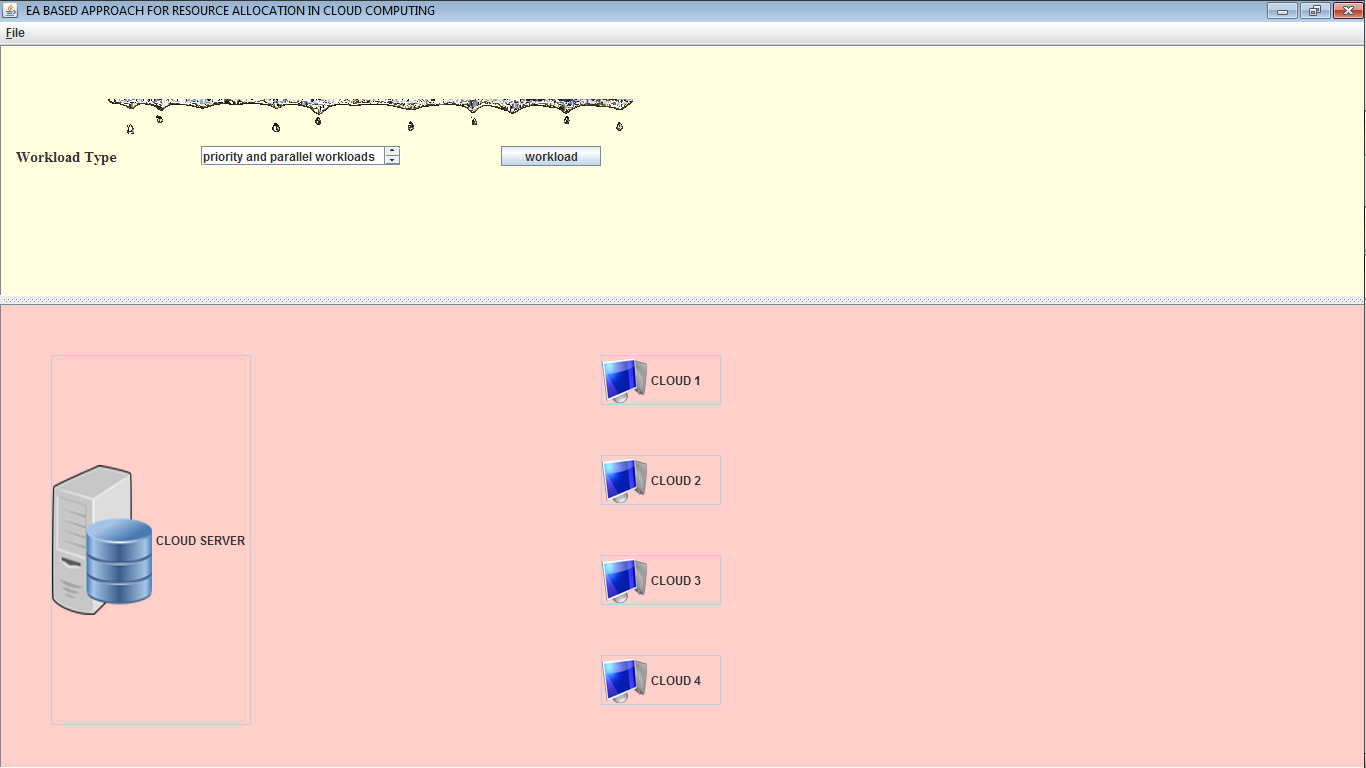
}

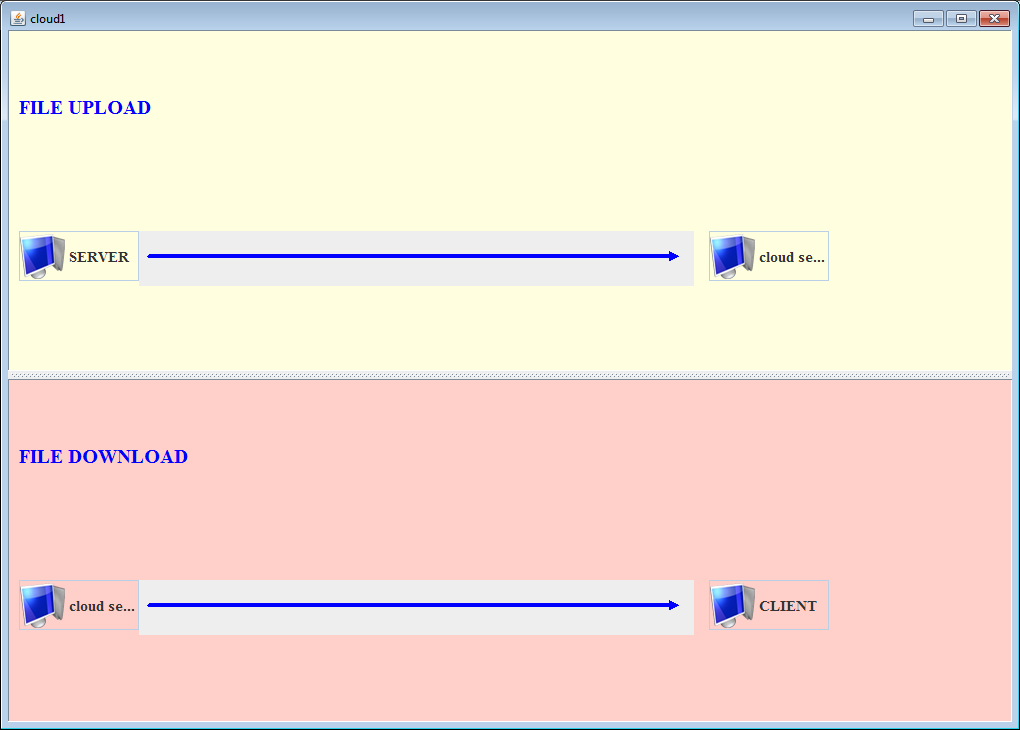
}

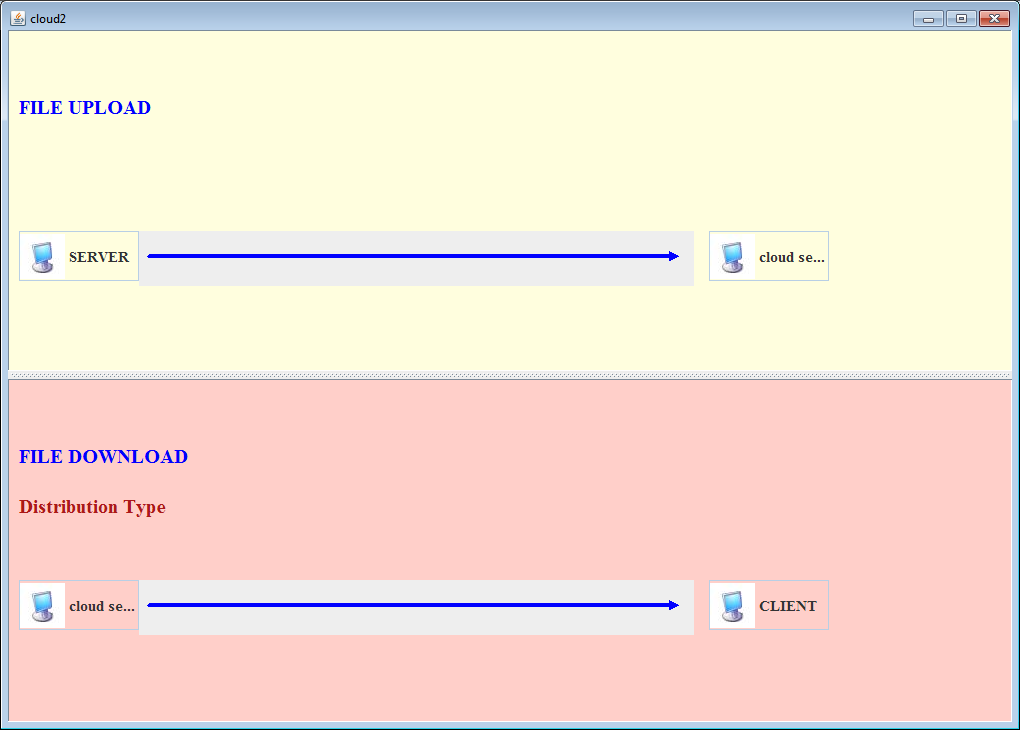
}

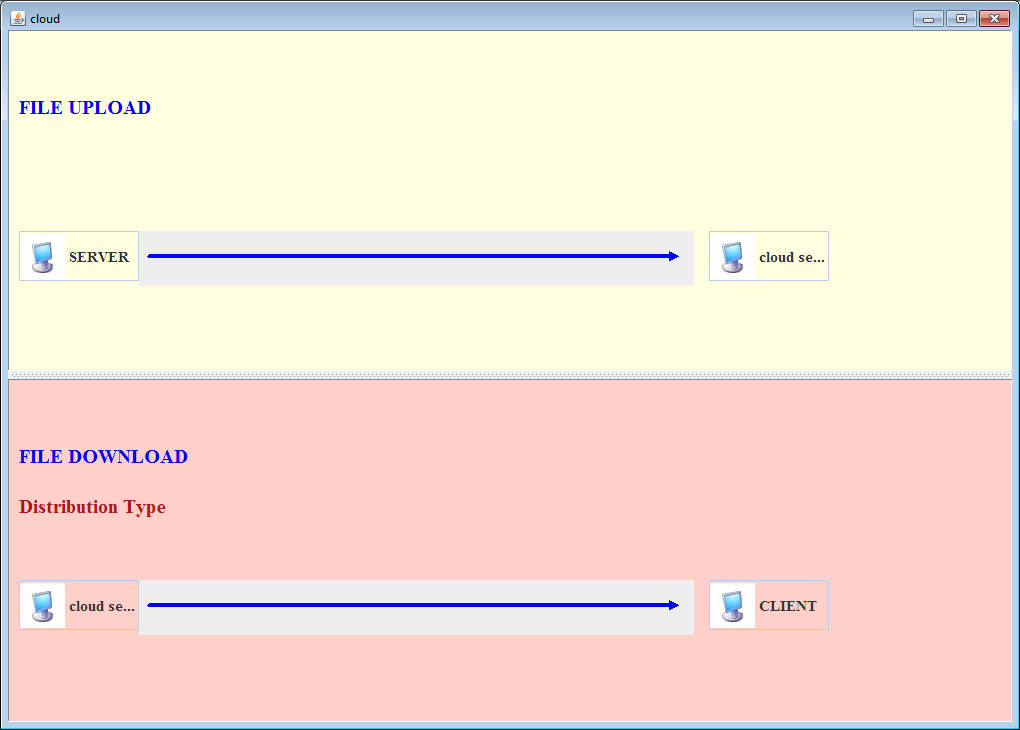
}

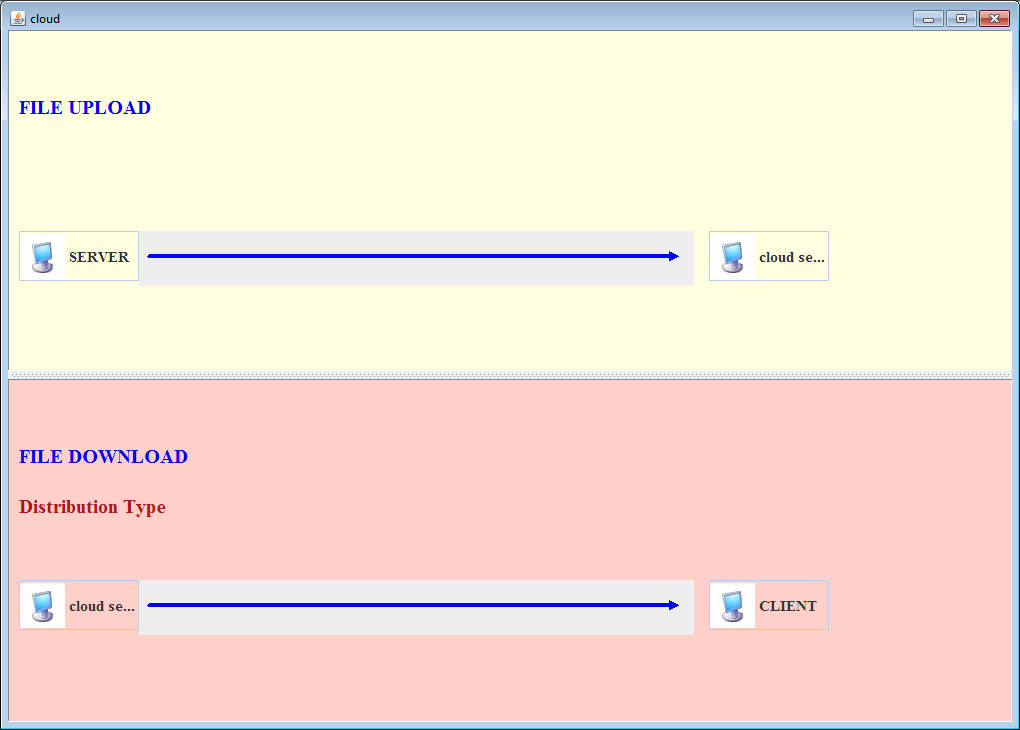
**SCREEN SHOTS**

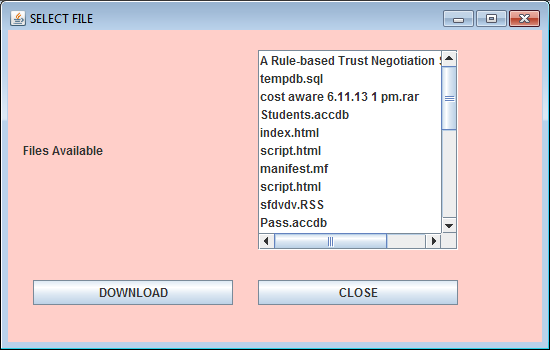


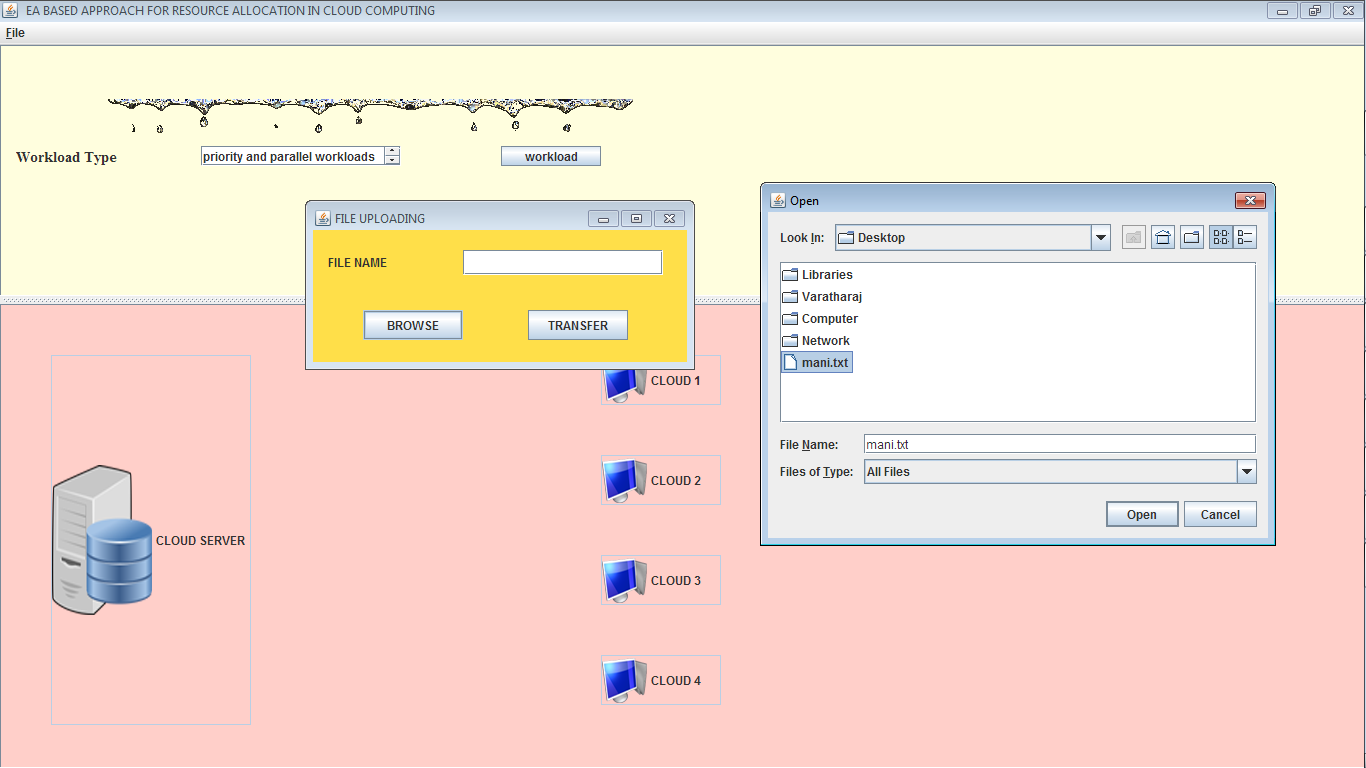


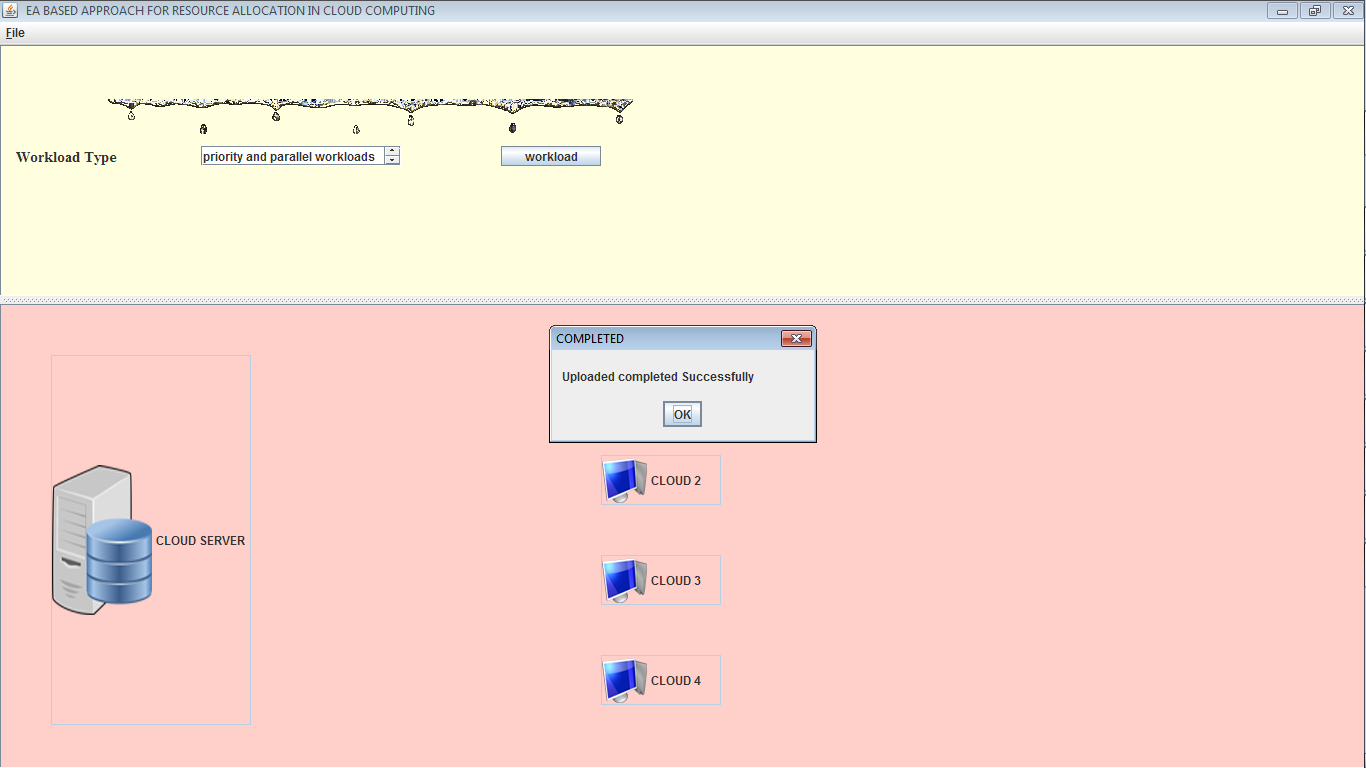


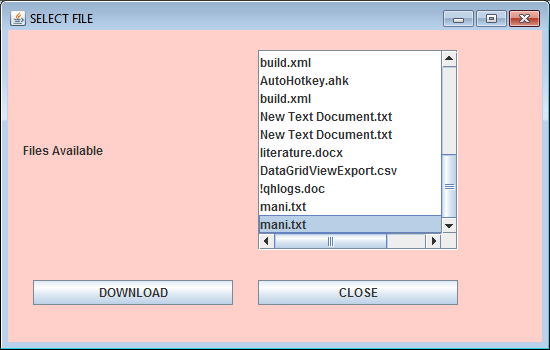












**CONCLUTION**

This document contains the in depth analysis of wireless networks and all the details that are required to form an understanding of Network simulation. The document is a report of the investigation phase of a project which is the development of a web based network simulator that allows the simulation and understanding of networks. This document gives the aim and the objectives of the project and the steps in which those objectives can be achieved All of the essential knowledge for the development for a network simulator is provided in a clear and concise manner. Once the entire project is completed it will provide users with a web based network simulator that is open source thus allowing users to use the source code to modify the application to suit their needs or use it in the creation of their own application. This application is intended to be used for educational purposes, thus it will have a user friendly and simplistic approach to network simulation.

The System implementation phase consists of the following steps:

* Testing the developed software with sample data.
* Correction of any errors if identified.
* Creating the files of the system with actual data.
* Making necessary changes to the system to find out errors.
* Training of user personnel.

The system has been tested with sample data, changes are made to the user requirements and run in parallel with the existing system to find out the discrepancies. The user has also been appraised how to run the system during the training period. The Multi cloud data transfer between the client and server to set the secret key to transfer from the data. It is using the secret key to encrypt the data to transfer the client to the server. The data received from the client side to decrypt the data then only view all the information from the available network.

**FEATURE ENHANCEMENT**

It is clear that although the use of cloud computing has rapidly increased cloud computing security is still considered the major issue in the cloud computing environment. Customers do not want to lose their private information as a result of malicious insiders in the cloud. In addition, the loss of service availability has caused many problems for a large number of customers recently. Furthermore, data intrusion leads to many problems for the users of cloud computing. The purpose of this work is to survey the recent research on single clouds and multi-clouds to address the security risks and solutions.

**BIBLIOGRAPHY**

1. Above the Clouds: A View of Cloud Computing Michael Armbrust, rmando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia *UC Berkeley Reliable Adaptive Distributed systems Laboratory (RAD Lab)*
2. Cloud computing Availability: Multi-clouds for Big data Service Sangdo Lee1, Hyoungyill Park1, Yongtae Shin1Department of Computing, Soongsil University, 511 Sangdo-Dong, Dongjak-Gu, Seoul, Korea { piterlee, harrypark58}@gmail.com, [shin@ssu.ac.kr](mailto:shin@ssu.ac.kr).
3. Cloud Computing and Grid Computing 360-Degree Compared 1,2,3Ian Foster, 4Yong Zhao, 1Ioan Raicu, 5Shiyong Lu *foster@mcs.anl.gov, yozha@microsoft.com, iraicu@cs.uchicago.edu,* [*shiyong@wayne.edu*](mailto:shiyong@wayne.edu)1 Department of Computer Science, University of Chicago, Chicago, IL, USA.
4. A Multi-Level Security Model for Partitioning Workflows over Federated Clouds September, 2011.
5. Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon PARC {rchow,pgolle,mjakobss,eshi,staddon} @parc.com.
6. Cloud Federation Tobias Kurze\_, Markus Klemsy, David Bermbachy, Alexander Lenkz, Stefan Taiy and Marcel Kunze\_ \_Steinbuch Centre for Computing (SCC) Karlsruhe titute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany Email: fkurze, [marcel.kunzeg@kit.edu](mailto:marcel.kunzeg@kit.edu) yInstitute of Applied Informatics and Formal Description Methods (AIFB) Karlsruhe Institute of Technology (KIT), Kaiserstrasse 12, 76131 Karlsruhe, Germany Email: fmarkus.klems, david.bermbach, [stefan.taig@kit.edu](mailto:stefan.taig@kit.edu) ZFZI Forschungszentrum Informatik Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany Email: [flenkg@fzi.de](mailto:flenkg@fzi.de).
7. Communication from the commission to the european Parliament, the council, the european economic and social Committee and the committee of the regions Unleashing the potential of cloud computing in Europe (text with eea relevance) {swd(2012) 271 final}.
8. Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing Alexandru Iosup, *Member, IEEE,* Simon Ostermann,Nezih Yigitbasi, *Member, IEEE,* Radu Prodan, *Member, IEEE,* Thomas Fahringer, *Member, IEEE,* and Dick Epema, *Member, IEEE.*
9. K. Birman, G. Chockler and R. van Renesse,"Toward a cloud computing research agenda", SIGACT News, 40, 2009, pp. 68-80.
10. K.D. Bowers, A. Juels and A. Opera, "HAIL: A high-availability and integrity layer for cloud storage", CCS'09: Proc. 16th ACM Conf. on Computer and communications security, 2009, pp. 187-198.
11. C. Cachin, R. Haas and M. Vukolic, "Dependable storage in the Intercloud", Research Report Z, 3783, 2010.
12. C. Cachin, I. Keidar and A. Shraer, "Trusting the cloud", ACM SIGACT News, 40, 2009, pp. 81-86. C. Cachin and S. Tessaro, "Optimal resilience for erasure-coded Byzantine distributed storage", DISC:Proc. 19thIntl.Conf. on Distributed Computing, 2005, pp. 497-498.
13. M. Castro and B. Liskov, "Practical Byzantine fault tolerance", Operating Systems Review, 33, 1998, pp. 173-186.
14. G. Chockler, R. Guerraoui, I. Keidar and M. Vukolic, "Reliable distributed storage", Computer, 42, 2009, pp. 60-67.
15. Clavister, "Security in the cloud", Clavister White Paper, 2008.
16. A.J. Feldman, W.P. Zeller, M.J. Freedman and E.W. Felten, "SPORC: Group collaboration using untrusted cloud resources", OSDI, October2010, pp. 1-14.
17. S.L. Garfinkel, "Email-based identification and authentication: An alternative to PKI?"
18. IEEE Security and Privacy, 1(6), 2003, pp. 20-26.
19. S.L. Garfinkel, "An evaluation of amazon’s grid computing services: EC2, S3, and SQS", Technical Report TR-08-07, Computer Science Group, Harvard University, Citeseer, 2007, pp. 1-15.
20. E. . . Goh, H. Shacham, N. Modadugu and D. Boneh, "SiRiUS: Securing remote untrusted storage",NDSS: Proc. Network and Distributed System Security Symposium, 2003, pp. 131–145.
21. A. Juels and B.S. Kaliski Jr, "PORs: Proofs of retrievability for large files", CCS '07: Proc. 14th ACM Conf. on Computer and communications security, 2007, pp. 584-597.