**CHAPTER 1- INTRODUCTION**

Cloud computing has powerful attractions for the organization. It offers instant access to an infinitely flexible computing resource and the ability to make major cost savings through outsourcing. Yet for many organisations, the final barrier to adopting Cloud computing is whether it is sufficiently secure.

In the jungle of multi-tenant data, one need to trust the cloud provider that their information will not be exposed..

As with most SaaS offerings, the applications offering are constantly being revised, a fact which raises more security issues for customers. Companies need to know, for instance, whether a software change might actually alter its security settings. For every update we review the security requirements for every user in the system

However, according to Datamonitor's Trifković, the cloud is still very much a new frontier with very little in the way of specific standards for security or data privacy. In many ways he says that cloud computing is in a similar position to where the recording industry found itself when it was trying to combat peer-to-peer file sharing with copyright laws created in the age of analogue.

Many are concerned that cloud computing remains at such an embryonic stage that the imposition of strict standards could do more harm than good.

**Security – The main Barrier in cloud computing**

One of the world's largest technology companies, Google, has invested a lot of money into the cloud space, where it recognises that having a reputation for security is a key determinant of success. "Security is built into the DNA of our products," says a company spokesperson. "Google practices a defense-in-depth security strategy, by architecting security into our people, process and technologies".

[Google was forced to make an embarrassing apology](http://www.computerweekly.com/Articles/2009/02/24/234988/google-mail-collapses.htm) in February when its Gmail servicecollapsed in Europe, while Salesforce.com is still smarting from a phishing attack in 2007 which duped a staff member into revealing passwords.

While cloud service providers face similar security issues as other sorts of organisations,analysts warn that [the cloud is becoming particularly attractive to cyber crooks](http://www.computerweekly.com/Articles/2009/01/20/234325/cyber-criminals-to-favour-financial-scams-in-2009-says.htm).

The richer the pot of data, the more cloud service providers need to do to protect it

**Local law and jurisdiction where data is held**

Possibly even more pressing an issue than standards in this new frontier is the emerging question of jurisdiction. Data that might be secure in one country may not be secure in another. In many cases though, users of cloud services don't know where their information is held. Currently in the process of trying to harmonise the data laws of its member states, the EU favours very strict protection of privacy, while in America laws such as the US Patriot Act invest government and other agencies with virtually limitless powers to access information including that belonging to companies.

### Cloud computing is facing trouble in seeping within the US federal lines because the US Federation doubts the security of their confidential data logs.

**Cloud computing** is surely taking the world by rage but as far as the **US government** is concerned, they still have their reservations in completely using **cloud** for their data management. **Vivek Kundra**, the White House’s chief information officer was a strong advocate of **cloud computing** but it appears that the US government will wait for another couple of years before entrusting their data to **cloud**.

He believes that **cloud computing** can open horizons that would prove to be wondrous for the IT industry. However, the **US** **government’s** priority is security and for this they are hesitant in allotting the **cloud computing** companies, the door keys of their data centers which are highly confidential. They say that **Contractors like Amazon**, **Google** and **Lockheed Martin** are known suppliers of **cloud** but despite that, the liability of getting their confident data in danger, still exists. These three companies have made it big by providing cloud services in several spheres of life. But the **US federation** does not want to go under pressure and bear any violent consequences later.

An example of the **US** security leak is the attack on **Pentagon** when 24,000 confidential files were hacked. Such discrepancy occurred in a manual situation however; on net too they will be able to make it work one way or the other.

“When done with the proper considerations and planning, **cloud computing** will be a very effective and efficient tool,” Ms. Takai said.

* 1. **. RATIONALE:**

To study the existing security architecture, specific standards for security or data privacy in cloud computing & propose a security concept for more consistent security and monitoring of cloud services specifically for safe delivery & modification of client data.

* 1. **. PROBLEM DEFINITION AND PROPOSED SOLUTION:**

**Problem Definition:**

Cloud computing puts your data outside of your organization. Also when we use a cloud computing service, we are limiting yourself to the amount of advanced security tools that you can put on the system. There are also other issues to consider. We have little control over how much audit information is collected. For example, we likely do not have access to failed log-in attempts, so we cannot proactively look for attack reconnaissance. Likewise, while we may maintain ownership of your own data, we do not likely own all of the access log data. That potentially creates legal problems. For example, if someone does illicitly access our information, we might need to get a court order to see where they are coming from. If however we maintained your data internally, you would have instant access to all of this information.

The problem was that the users or the clients are not satisfied with the data security on cloud service provider. Clients want more security of data when they upload their confidential data on cloud. However, if our organization with a great deal of [intellectual property](http://www.csoonline.com/article/204600), believe that our data is valuable, and intend to implement more than basic security measures, we probably need to maintain your own data infrastructure. We can however review cloud computing providers and see if they allow for the implementation of the security countermeasures we believe are necessary.

The Cloud computing security concept we propose will hand over ultimate control to the customer in order to meet their requirements for authentication, authorisation & encryption of resources available through cloud service provider.

**Proposed Solution:**

* Requirement of an Operating System which will support the architecture of Cloud Computing. If we have such an OS which is made for the Cloud Computing then it will be easy to run Cloud Apps on that OS.
* Use of random of Encryption Algorithms so that the **intruder or hacker** find it difficult to peep into clients personal Data
* Providing security at CSP (Cloud Service Provider) level

**1.3. SYSTEM SYNTHESIS:**

Synthesis is the process in which we begin from principles and [proceed to] build up theorems and problems,while analysis is the process in which we begin with a given conclusion or proposed problem and seek the principles by which we may demonstrate the conclusion or solve the problem

Cloud computing has powerful attractions for the organisation. It offers instant access to an infinitely flexible computing resource and the ability to make major cost savings through outsourcing. Yet for many organisations, the final barrier to adopting Cloud computing is whether it is sufficiently secure.

SaaS (software as a service) and PaaS (platform as a service) providers all trumpet the robustness of their systems, often claiming that security in the cloud is tighter than in most enterprises. But the simple fact is that every security system that has ever been breached was once thought infallible.

However, according to Datamonitor's Trifković, the cloud is still very much a new frontier with very little in the way of specific standards for security or data privacy. In many ways he says that cloud computing is in a similar position to where the recording industry found itself when it was trying to combat peer-to-peer file sharing with copyright laws created in the age of analogue.

Many are concerned that cloud computing remains at such an embryonic stage that the imposition of strict standards could do more harm than good.

First, identify the business processes and information requirements associated with those business processes and then build your security services so that they adequately protect these business services from realistic threats and attack vectors - regardless of delivery model.

The challenge then is to identify appropriate technologies and processes to meet these architectural requirements in the implementation across various delivery models to ensure a consistent level of control. It is equally important to ascertain the levels of assurance required of these identified controls, as this may rule out certain delivery models, subject to business risk tolerances. The other challenge is to use a lightweight approach that does not stifle the flexibility and speed of implementation associated with delivery "as a service".

For too long security has been seen as a blocker and detached from the business, security architecture is the means to bridge this gap and help organisations to take advantage of new technologies within risk tolerances that the business understands and accepts.

* 1. **. REPORT ORGANIZATION**
* **Chapter 1** will provide the introduction of the project.
* **Chapter 2** will provide the overall description of the literature review of the application.
* **Chapter 3** will contain the Process Model Adopted for the application in which the sub headings are

Analysis. Design, Implementation and Testing.

* **Chapter 4** will contain Result of the application.
* **Chapter 5** will contain Conclusion and future work/limitation for the application.

**CHAPTER 2 - LITERATURE REVIEW**

Research papers and literatures on **“**Overview and Analysis of Cloud Computing Research and Application” by Yizeng Chen , Xingui Li and Fangning Chen (School of Management Shanghai University, SHU Shanghai, China). This paper aims to the current cloud computing research and application and analyzes the characteristic of cloud computing.[9]

**“**Ensuring Security-The last barrier to Cloud Adoption (Cable & wireless worldwide) in March 2011.This White Paper examines the perceived risks, assesses whether they are justified, and the technology and measures that can make the Cloud’s ‘virtual’ security a reality.” [8]

“Cloud computing security threats and responses” by Farzad Sabahi Faculty of Computer Engineering in Azad University, Iran.This paper summarizes availability, and security issues for cloud computing (RAS issues), and propose feasible and available solutions for some of them**.[10]**

”Cloud security issues” by BalachandraReddyKandukuri,Ramakrishna Paturi V from Advanced Software Technologies International Institute of Information Technology, Pune proposed that there has to be a standardized way to prepare the SLA irrespective to the providers. This can help some of the enterprises to look forward in using the cloud services. In this paper, they put forward some security issues that have to be included in SLA(Service Level Agreement).[11]

After the concept of Cloud computing--the Revolutionary development and progress in IT industry-- was born in Google, the IT software and services vendors, universities, enterprises, etc, join in the process of research and application of cloud computing.

Cloud computing technology is not an innovation, but the integration of past technology, even also the future of the software industry model (software 10.0). On the definition of cloud computing has not yet formed a unified understanding in academia, but what has generally been accepted is that cloud computing is the developing result of Grid Computing, Distributed Computing, Parallel Computing, Utility Computing, Network Storage Technologies, Virtualization,Load Balance, etc traditional computer technology. It is designed to provide users with cloud architecture nodes and via the Internet or intranet to integrate the number of relatively low-cost computer entities into one system with powerful computing capabilities. With the help of software of architecture (SOA), SaaS, PaaS, IaaS, management service supply (MSS) and other advanced software models, this powerful Cloud computing capability distributed to the cloud users’ (individual or corporate) hands.

The core vision of Cloud Computing is to continuously improve the compute power of "cloud", thereby reducing the processing burden and the cost of cloud users, ultimately to simplify the cloud users into an simple Input and Output devices (I\O Infrastructure), and to enjoy the powerful "cloud" computing capability On-Demand.

These issues which discussed in the project are the main reasons that cause many enterprises which have a plane to migrate to cloud prefer using cloud for less sensitive data and store important data in their own local machine.

**2.1. ARCHITECTURE USED:**

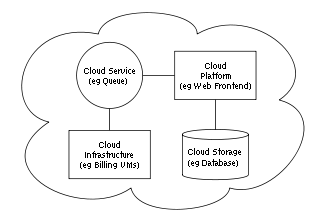
Cloud architecture, the [systems architecture](http://en.wikipedia.org/wiki/Systems_architecture) of the [software systems](http://en.wikipedia.org/wiki/Software_systems) involved in the delivery of cloud computing, typically involves multiple cloud components communicating with each other over a [loose coupling](http://en.wikipedia.org/wiki/Loose_coupling) mechanism such as a messaging queue. Elastic provision implies intelligence in the use of tight or loose coupling as applied to mechanisms such as these and others.

### The Intercloud

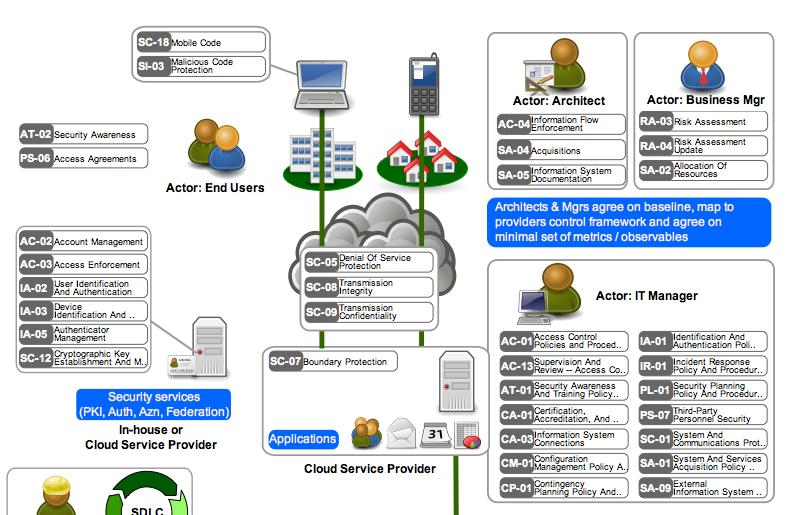
 The Intercloud is an interconnected global "cloud of clouds" and an extension of the [Internet](http://en.wikipedia.org/wiki/Internet) "network of networks" on which it is based.

### Cloud engineering

**Cloud engineering** is the application of [engineering](http://en.wikipedia.org/wiki/Engineering) disciplines to cloud computing. It brings a systematic approach to the high level concerns of commercialisation, standardisation, and governance in conceiving, developing, operating and maintaining cloud computing systems. It is a multidisciplinary method encompassing contributions from diverse areas such as [systems](http://en.wikipedia.org/wiki/Systems_engineering), [software](http://en.wikipedia.org/wiki/Software_engineering), [web](http://en.wikipedia.org/wiki/Web_engineering), [performance](http://en.wikipedia.org/wiki/Performance_engineering),[information](http://en.wikipedia.org/wiki/Information_engineering), [security](http://en.wikipedia.org/wiki/Security_engineering), [platform](http://en.wikipedia.org/wiki/Platform_engineering), [risk](http://en.wikipedia.org/wiki/Risk_analysis_(engineering)), and [quality](http://en.wikipedia.org/wiki/Quality_control) engineering.

****

**Fig. 1(Cloud Architecture)**

****

**Fig.2 (Security Architecture)**

**2.2. EXISTING SYSTEM:**

Until it is there are nevertheless a [handful of existing web standards](http://www.computerweekly.com/Articles/2008/12/16/233939/security-zone-promoting-accountability-through-isoiec-27001-27002-formerly-isoiec.htm) which companies in the cloud should know about. Chief among these is “[ISO27001](http://www.27001-online.com/)”, which is designed to provide the foundations for third party audit, and implements OECD principles governing security of information and network systems. The [SAS70](http://www.sas70.com/) auditing standard is also used by cloud service providers.

Say the words "security architecture" to many security professionals and they will automatically start thinking about firewalls and the placement of network intrusion detection sensors. Presently there are many architecture in the security elements of an enterprise architecture. Enterprise architecture frameworks such as Zachman and TOGAF are relatively weak in their treatment of security, but that does not mean that the underlying concepts and drivers of enterprise architecture are not equally applicable to security, or that such frameworks are worthless from a security perspective.

If you have built a system that meets the security requirements (sourced from the business through standard requirements analysis and an agreed risk analysis), and that abides by all applicable principles and service contracts, then the system is adequately 'secure' almost by definition - subject to appropriate assurance activities.

Does that mean that the system can't be hacked? No. But it should mean that if it *is* hacked, this is a risk that has been factored into the architecture and accepted by the business.

**2.2.1. TOOLS AND TECHNOLOGIES:**

**2.2.1.1. JAVA**

**Introduction**

Java is a programming language originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture.

**History**

James Gosling initiated the Java language project in June 1991 for use in one of his many set-top box projects. Gosling aimed to implement a virtual machine and a language that had a familiar C/C++ style of notation. Sun released the first public implementation as Java 1.0 in 1995. It promised "Write Once, Run Anywhere" (WORA), providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowednetwork- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular.[4]

**Principles**

There were five primary goals in the creation of the Java language:

1. It should be "simple, object oriented and familiar".

2. It should be "robust and secure".

3. It should be "architecture neutral and portable".

4. It should execute with "high performance".

5. It should be "interpreted, threaded, and dynamic".[4]

**Java Platform**

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any supported hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to platform-specific machine code. Java bytecode instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets. [4]

**Automatic memory management**

Java uses an automatic garbage collector to manage memory in the object lifecycle. The programmer determines when objects are created, and the Java runtime is responsible for recovering the memory once objects are no longer in use. Once no references to an object remain, the unreachable memory becomes eligible to be freed automatically by the garbage collector. One of the ideas behind Java's automatic memory management model is that programmers be spared the burden of having to perform manual memory management.

**Syntax**

The syntax of Java is largely derived from C++. Unlike C++, which combines the syntax for structured, generic, and object-oriented programming, Java was built almost exclusively as an object oriented language. All code is written inside a class and everything is an object, with the exception of the intrinsic data types (ordinal and real numbers, boolean values, and characters), which are not classes for performance reasons. Java uses the same commenting methods as C++. There are two different styles of comment: a single line style marked with two forward slashes, and a multiple line style opened with a forward slash asterisk (/\*) and closed with an asterisk forward slash (\*/).[4]

**2.2.1.2. HTML**

HTML, which stands for [Hyper Text](http://en.wikipedia.org/wiki/HyperText) Markup Language, is the predominant [markup language](http://en.wikipedia.org/wiki/Markup_language) for [web pages](http://en.wikipedia.org/wiki/Web_page). HTML is the basic building-blocks of webpages.

HTML is written in the form of [HTML elements](http://en.wikipedia.org/wiki/HTML_element) consisting of tags, enclosed in [angle brackets](http://en.wikipedia.org/wiki/Bracket#Angle_brackets_or_chevrons_.E2.9F.A8_.E2.9F.A9) (like <html>), within the web page content. HTML tags normally come in pairs like <h1> and </h1>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags).

The purpose of a [web browser](http://en.wikipedia.org/wiki/Web_browser) is to read HTML documents and compose them into visual or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows [images and objects](http://en.wikipedia.org/wiki/HTML_element#Images_and_objects) to be embedded and can be used to create [interactive forms](http://en.wikipedia.org/wiki/HTML_element#Forms). It provides a means to create [structured documents](http://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](http://en.wikipedia.org/wiki/Semantic) for text such as headings, paragraphs, lists, links, quotes and other items. It can embed [scripts](http://en.wikipedia.org/wiki/Scripting_language) in languages such as [JavaScript](http://en.wikipedia.org/wiki/JavaScript) which affect the behavior of HTML webpages.

Web browsers can also refer to [Cascading Style Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) to define the appearance and layout of text and other material. The [W3C](http://en.wikipedia.org/wiki/W3C), maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentational HTML markup.[3]

**2.2.1.3. JAVA SERVLETS**

A Servlet is a [Java class](http://en.wikipedia.org/wiki/Java_class) in [Java EE](http://en.wikipedia.org/wiki/Java_EE) that conforms to the Java Servlet API, a protocol by which a Java class may respond to [HTTP](http://en.wikipedia.org/wiki/HTTP) requests. They are not tied to a specific client-server protocol, but are most often used with this protocol. The word "Servlet" is often used in the meaning of "HTTP Servlet". Thus, a [software developer](http://en.wikipedia.org/wiki/Software_developer) may use a servlet to add [dynamic content](http://en.wikipedia.org/wiki/Dynamic_web_page) to a [Web server](http://en.wikipedia.org/wiki/Web_server) using the [Java platform](http://en.wikipedia.org/wiki/Java_platform). The generated content is commonly [HTML](http://en.wikipedia.org/wiki/HTML), but may be other data such as [XML](http://en.wikipedia.org/wiki/XML). Servlets are the [Java](http://en.wikipedia.org/wiki/Java_%28software_platform%29) counterpart to non-Java dynamic Web content technologies such as [CGI](http://en.wikipedia.org/wiki/Common_Gateway_Interface) and [ASP.NET](http://en.wikipedia.org/wiki/Active_Server_Pages). Servlets can maintain [state](http://en.wikipedia.org/wiki/State_%28computer_science%29) in [session](http://en.wikipedia.org/wiki/Session_%28computer_science%29) variables across many server transactions by using [HTTP cookies](http://en.wikipedia.org/wiki/HTTP_cookie), or [URL rewriting](http://en.wikipedia.org/wiki/URL_rewriting).[3]

The servlet [API](http://en.wikipedia.org/wiki/Application_programming_interface), contained in the [Java package](http://en.wikipedia.org/wiki/Java_package) hierarchy [javax.servlet](http://java.sun.com/javaee/6/docs/api/javax/servlet/package-summary.html), defines the expected interactions of a [Web container](http://en.wikipedia.org/wiki/Web_container) and a servlet. A Web container is essentially the component of a Web server that interacts with the servlets. The Web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

A [Servlet](http://java.sun.com/javaee/6/docs/api/javax/servlet/Servlet.html) is an [object](http://en.wikipedia.org/wiki/Object_%28computer_science%29) that receives a request and generates a response based on that request. The basicservlet package defines Java objects to represent servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environment. The package [javax.servlet http](http://java.sun.com/javaee/6/docs/api/javax/servlet/http/package-summary.html) defines [HTTP](http://en.wikipedia.org/wiki/HTTP)-specific subclasses of the generic servlet elements, including session management objects that track multiple requests and responses between the Web server and a client. Servlets may be packaged in a [WAR file](http://en.wikipedia.org/wiki/WAR_%28Sun_file_format%29) as a [Web application](http://en.wikipedia.org/wiki/Web_application).

Servlets can be generated automatically from [JavaServer Pages](http://en.wikipedia.org/wiki/JavaServer_Pages) (JSP) by the [JavaServer Pages compiler](http://en.wikipedia.org/wiki/JavaServer_Pages_compiler). The difference between Servlets and JSP is that Servlets typically embed HTML inside Java code, while JSPs embed Java code in HTML. While the direct usage of Servlets to generate HTML (as shown in the example below) is relatively rare nowadays, the higher level MVC web framework in Java EE ([JSF](http://en.wikipedia.org/wiki/Java_Server_Faces)) still explicitly uses the Servlet technology for the low level request/response handling via the FacesServlet. A somewhat older usage is to use servlets in conjunction with JSPs in a pattern called "[Model 2](http://en.wikipedia.org/wiki/Model_2)", which is a flavour of the [model-view-controller](http://en.wikipedia.org/wiki/Model-view-controller) pattern.[3]

**Life cycle of a servlet**

1. The container calls the no-arg constructor.

2. The Web container calls the init() method. This method initializes the servlet and must

be called before life of a servlet, the init() method is called only once.

3. After initialization, the servlet can service client requests. Each [request](http://en.wikipedia.org/wiki/HTTP_request#Request_message) is serviced in its

own separate thread. The Web container calls the service() method of the servlet for

every request.

The service() method determines the kind of request being made and dispatches it to an appropriate method to handle the request. The developer of the servlet must provide an implementation for these methods. If a request for a method that is not implemented by the servlet is made, the method of the parent class is called, typically resulting in an error being returned to the requester.

4. Finally, the Web container calls the destroy() method that takes the servlet out of

service. The destroy() method, like init(), is called only once in the lifecycle of a servlet.

Servlets are most often used to:

* Process or store data that was submitted from an HTML form.
* Provide dynamic content such as the results of a database query
* Manage state information that does not exist in the stateless HTTP protocol, such as filling the articles into the shopping cart of the appropriate customer.

**CHAPTER 3: PROCESS MODEL ADOPTED**

**3.1. ANALYSIS:**

Based on the analysis regarding the project the different phases are given below :

**3.1.1. REQUIRMENT ANALYSIS**

**Requirements analysis** in [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering) and [software engineering](http://en.wikipedia.org/wiki/Software_engineering), encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting [requirements](http://en.wikipedia.org/wiki/Requirements) of the various users.

Requirements analysis is critical to the success of a development project.

The aim of the project was to develop a software which gives more data security at user level when the user save his data on the cloud with the help of cloud service provider.

**3.1.2. CONCEPT PLANNING**

The important task in creating a software product is extracting the [requirements](http://en.wikipedia.org/wiki/Requirement) or [requirements analysis](http://en.wikipedia.org/wiki/Requirements_analysis). Customers typically have an abstract idea of what they want as an end result.

Once the general requirements are gathered from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document.

**3.1.2.1. PROJECT RESOURCES**

**Cost:**

It is a project or a piece of technology; it’s a mission-critical business system and a significant investment that requires proper planning.

It is far too expensive to leave to guess work, intuition, or a single-minded purpose such as communications or HR. In order to truly realize the full value of the investment the it must have a proper plan that accounts for all aspects of the business – including HR, IT, communications, operations, finance, etc. A proper plan begins with a thorough assessment.

**Duration:**

It is more than a project or a piece of technology; it’s a mission-critical business system. Therefore, it may take a considerable time for successful completion.

**Manpower and assessment resources:**

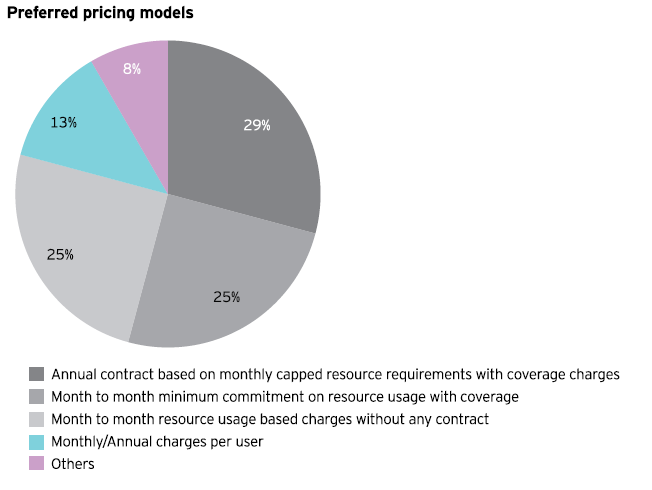
Before undertaking any website plan or build, an extensive needs or business requirements assessment is necessary to identify, develop, prioritize, and document goals and current practices.

The assessment should include stakeholder interviews and input, as well as user research, and possibly stakeholder workshops. When building a leading-edge website, a detailed strategic blueprint can be crafted with the acquired data and knowledge including:

* Creative
* Information architecture
* Technology

It is recommended that any organization consider engaging a third-party or consultant to conduct the assessment. While the cost may be prohibitive for organizations with tight budgets, a third-party may be more successful in gathering sensitive opinions and feedback as a third-party, unlike stakeholders, have no personal attachment or stake in the intranet and do not have any political agendas.

It is important to gather the needs and requirements of stakeholder and users, at the risk of failure; a representative sampling of user opinions is crucial to gathering an accurate reading on user needs and requirements.

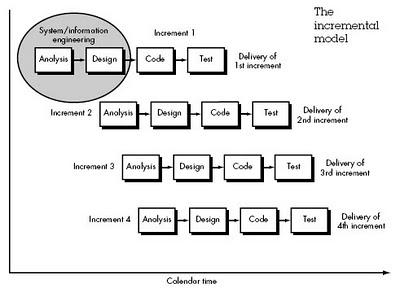
****

**Fig.3**

**3.1.4. ARCHITECTURAL SPECIFICATION**

**MODEL USED**

**Incremental Model**

****

Incremental model is an evolution of waterfall model. The product is designed, implemented, integrated and tested as a series of incremental builds.

It is a popular model software evolution used many commercial software companies and system vendor.Incremental software development model may be applicable to projects where:

* Software Requirements are well defined, but realization may be delayed.
* The basic software functionality are required early.

**ADVANTAGES AND DISADVANTAGES**

**Advantages**

* Generates working software quickly and early during thesoftware life cycle.
* More flexible – less costly to change scope and requirements.
* Easier to test and debug during a smaller iteration.
* Easier to manage risk because risky pieces are identified and handled during its iteration.
* Each iteration is an easily managed milestone.

**Disadvantages**

* Each phase of an iteration is rigid and do not overlap each other
* Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.

**REASON FOR USING ‘INCREMENTAL MODEL’**

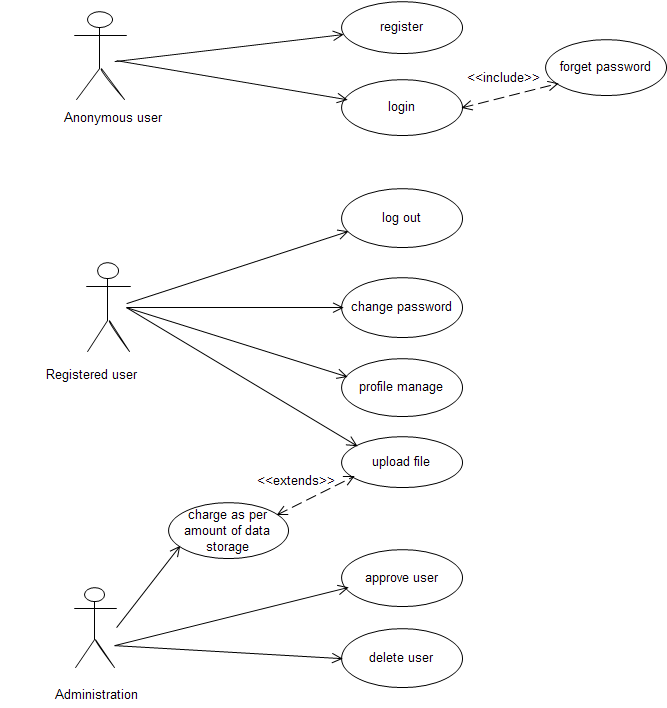
**T**he [**incremental**](http://en.wikipedia.org/wiki/Incremental)**build model** is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements.

The product is decomposed into a number of components, each of which are designed and built separately (termed as builds). Each component is delivered to the client when it is complete. This allows partial utilisation of product and avoids a long development time. It also creates a large initial capital outlay with the subsequent long wait avoided. This model of development also helps ease the traumatic effect of introducing completely new system all at once. There are, overall, few problems with this model.

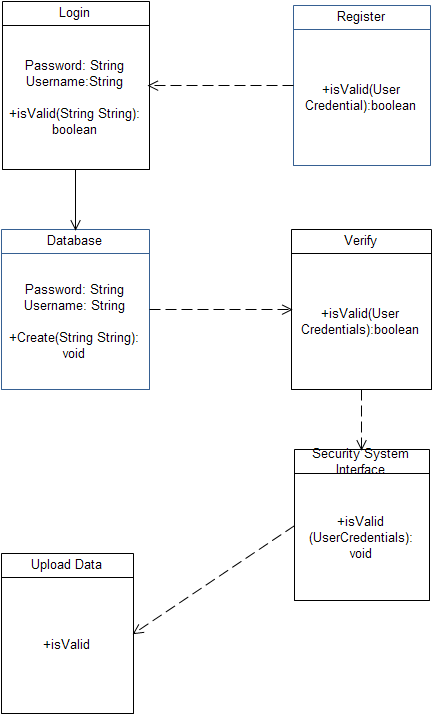
Therefore,we are using this model for our project.

**3.2. DESIGN**

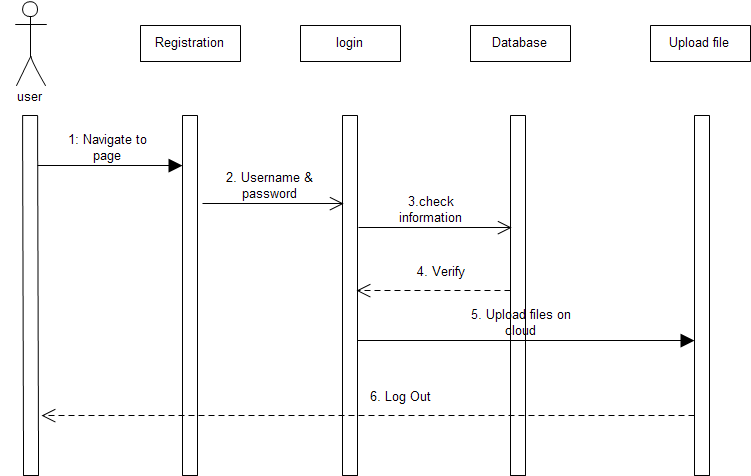
**3.2.2. Use Case Diagram**

****

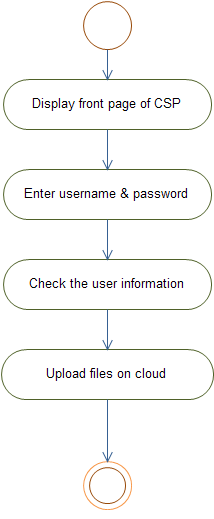
**3.2.3. Class Diagram**

****

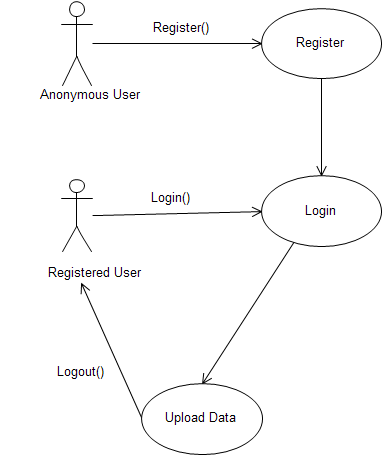
**3.2.4. Sequence Diagram**

****

**3.2.5. Activity Diagram**

****

**3.2.5. Collabration Diagram**

****

**3.3. IMPLEMANTATION AND TESTING**

**3.3.1. Testing Strategy Adopted**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet-undiscovered error.A successful test is one that uncovers an as-yet undiscovered error.

If the testing is conducted successfully (according to the objectives stated above) then it will uncover errors in the software.

As a secondary benefits, testing demonstrates that software functions appear to be working according to specification, that behavioral and performance requirements appear to have been met.

Data collected as testing is conducted provide a good indication of software reliability and indication of quality

**3.3.2. Testing Principles**

Davis [DAV95] suggests a set of testin g principles that have been adapted in our project: -

* All tests should be traceable to customer requirements:

The objectives of software testing is to uncover errors, it follows that the most severe defects (from the customer’s point of view) are those that program to fail to meet its requirements.

* Tests should be planned long before testing begins:

Test planning can begin as soon as the requirements model is complete. Detailed definition of test cases can begin as soon as the design model has been solidified. Therefore, all tests can plan and designed before any code has been generated.

* The Pareto principle applies to software testing:

Stated simply, the Pareto principle implies that 80 percent of all errors uncovered during testing will likely be traceable to 20 percent of all program components. The problem, of course is to isolate the suspect components and to thoroughly test them.

* Testing should begin “in the small’ and progress toward testing “in the large”:

The first tests planned and executed generally focus on individual components. As testing progress, focus shifts in an attempt to find errors in integrated clusters of components and ultimately in the entire system.

* Exhaustive testing is not possible:

The no. of path permutations for even a moderately sized program is exceptionally large. For this reason, it is impossible to execute every combination of paths during testing. It is possible, however to adequately cover program logic and to ensure that all conditions in the component-level design have been exercised.

To be most effective, testing should be conducted by an independent third party: The testing has the highest probability of finding errors therefore the software engineer who created the system is not the best person to conduct all tests for the software.

**3.3.3. Unit or Component Testing**

In unit testing the individual components are tested independently to ensure their quality. The focus is to uncover the errors in design and implementation. We done the unit testing on our project and recover all the errors which are occurring during the testing.

**3.3.4. Integration Testing**

In integration testing a group of components are tested together to ensure their quality of their integration unit. The objective is to take unit tested components and build a program structure that has been dictated by software design. The integration testing can be carried out using two approaches-

* The non – incremental integration
* Incremental integration

**3.3.5. Validation Testing**

The integrated software is tested based on requirements to ensure that the desired product is obtained. In validation testing the main focus is to uncover error in

* System input/output
* System function and information data
* System interfaces with external parts
* User interface
* System behaviour and performance.

**3.3.6. User Acceptance Testing**

The Acceptance testing is a kind of testing conducted to ensure that the software works correctly in the user work environment. This software is working properly and giving the required output.

**3.3.7. Test Cases**

**Login Module**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case No.** | **Action** | **Expected Result** | **Success** | **Comments** |
| 1.0 | Do not enter any of username or password fields and press the Button ‘Login’ | Error message is displayed. Please re- login. | Yes | The action was successfully tested. |
| 1.1 | Click the submit button with invalid username or password. | Either the username or password is incorrect. Please check and re- login | Yes | The action was successfully tested. |
| 1.2 | Enter valid username and password in the password field. | Navigation page displayed on the user login and user can begin trading. | Yes | The action was successfully tested. |

**Registration Module**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case No.** | **Action** | **Expected Result** | **Success** | **Comments** |
| 1.0 | Do not left the fields empty. | Error message is displayed. | Yes | The action was successfully tested. |
| 1.1 | Click the submit button with already existing username. | Choose another username. | Yes | The action was successfully tested. |
| 1.2 | Enter the date of birth in correct format as given. | It will not be accepted. | Yes | The action was successfully tested. |

**CHAPTER 4 RESULT**

* Requirement of an Operating System which will support the architecture of Cloud Computing. If we have such an OS which is made for the Cloud Computing then it will be easy to run Cloud Apps on that OS.
* Encryption and Decryption of document file at client side so that after uploading it on cloud the Cloud Service Provider itself cannot access the file.

**CHAPTER 5- CONCLUSION AND FUTURE WORK/ LIMITATIONS**

**Conclusion :**

First, identify the business processes and information requirements associated with those business processes and then build your security services so that they adequately protect these business services from realistic threats and attack vectors - regardless of delivery model.

The challenge then is to identify appropriate technologies and processes to meet these architectural requirements in the implementation across various delivery models to ensure a consistent level of control. It is equally important to ascertain the levels of assurance required of these identified controls, as this may rule out certain delivery models, subject to business risk tolerances.

The other challenge is to use a lightweight approach that does not stifle the flexibility and speed of implementation associated with delivery "as a service".

For too long security has been seen as a blocker and detached from the business, security architecture is the means to bridge this gap and help organisations to take advantage of new technologies within risk tolerances that the business understands and accepts.

**Limitations:**

The module which is implemented is for a particular enterprise where the employer can store the organization files in encrypted form so that Cloud Service Provider itself cannot access the confidential files and logs.

**FutureWork :**

We can use random of Encryption Algorithms so that the intruder or hacker find it difficult to peep into clients personal Data.

**REFERENCES**

1. en.wikipedia.org/wiki/Intranet
2. en.wikipedia.org/wiki/Email
3. JSP The complete reference
4. JAVA The complete reference
5. 5. Top five cloud computing security issues by [David Binning](http://www.computerweekly.com/authors/articleauthor.aspx?liArticleID=235782), April 2009

[http://www.computerweekly.com/Articles/2010/01/12/235782/Top-five-cloud-computing-](http://www.computerweekly.com/Articles/2010/01/12/235782/Top-five-cloud-computing-            security-issues.htm)

[security-issues.htm](http://www.computerweekly.com/Articles/2010/01/12/235782/Top-five-cloud-computing-            security-issues.htm)

6. Security Zone: Cloud computing puts the spotlight on security architecture by [Lee Newcombe](http://www.computerweekly.com/authors/articleauthor.aspx?liArticleID=244113),November2010,<http://www.computerweekly.com/Articles/2010/11/25/244113/Security-Zone-Cloud-computing-puts-the-spotlight-on-security.htm>

7. Cloud computing is facing trouble in seeping within the US federal lines because the US Federation doubts the security of their confidential data logs.[http://cloudtechsite.com/blogposts/us-federation-expresses-doubt-over-use-of %E2%80%98cloud-technology%E2%80%99.html](http://cloudtechsite.com/blogposts/us-federation-expresses-doubt-over-use-of%20%E2%80%98cloud-technology%E2%80%99.html)

8. White Paper on Ensuring Security The Last Barrier To Cloud Computing,March 2011,Cable and Wireless Worldwide

9. Research Paper from IEEE on Overview and analysis of cloud computing research and its application

10. Cloud security issues,2009 IEEE international conference on Service ComputingIEEE Cloud computing threats and responses,from faculty of computer engineering Azad University,Iran

11. Cloud security issues”-BalachandraReddyKandukuri,Ramakrishna Paturi V,Advanced Software Technologies,International Institute of Information Technology, Pune